Studies in Big Data 1

Wesley W. Chu *Editor*

Data Mining and Knowledge Discovery for Big Data

Methodologies, Challenge and Opportunities

Springer

Studies in Big Data

Volume 1

Series Editor

Janusz Kacprzyk, Warsaw, Poland

For further volumes:
http://www.springer.com/series/11970

Wesley W. Chu
Editor

Data Mining and Knowledge Discovery for Big Data

Methodologies, Challenge and Opportunities

Springer

Editor
Wesley W. Chu
Department of Computer Science
University of California
Los Angeles
USA

ISSN 2197-6503 ISSN 2197-6511 (electronic)
ISBN 978-3-642-40836-6 ISBN 978-3-642-40837-3 (eBook)
DOI 10.1007/978-3-642-40837-3
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013947706

© Springer-Verlag Berlin Heidelberg 2014
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The field of data mining has made significant and far-reaching advances over the past three decades. Because of its potential power for solving complex problems, data mining has been successfully applied to diverse areas such as business, engineering, social media, and biological science. Many of these applications search for patterns in complex structural information. This transdisciplinary aspect of data mining addresses the rapidly expanding areas of science and engineering which demand new methods for connecting results across fields. In biomedicine for example, modeling complex biological systems requires linking knowledge across many levels of science, from genes to disease. Further, the data characteristics of the problems have also grown from static to dynamic and spatiotemporal, complete to incomplete, and centralized to distributed, and grow in their scope and size (this is known as *big data*). The effective integration of big data for decision-making also requires privacy preservation. Because of the board-based applications and often interdisciplinary, their published research results are scattered among journals and conference proceedings in different fields and not limited to such journals and conferences in knowledge discovery and data mining (KDD). It is therefore difficult for researchers to locate results that are outside of their own field. This motivated us to invite experts to contribute papers that summarize the advances of data mining in their respective fields.Therefore, to a large degree, the following chapters describe problem solving for specific applications and developing innovative mining tools for knowledge discovery.

This volume consists of nine chapters that address subjects ranging from mining data from opinion, spatiotemporal databases, discriminative subgraph patterns, path knowledge discovery, social media, and privacy issues to the subject of computation reduction via binary matrix factorization. The following provides a brief description of these chapters.

Aspect extraction and entity extraction are two core tasks of aspect-based opinion mining. In Chapter 1, Zhang and Liu present their studies on people's opinions, appraisals, attitudes, and emotions toward such things as entities, products, services, and events.

Chapters 2 and 3 deal with spatiotemporal data mining(STDM) which covers many important topics such as moving objects and climate data. To understanding the activities of moving objects, and to predict future movements and detect anomalies in trajectories, in Chapter 2, Li and Han propose Periodica, a new mining technique, which uses reference spots to observe movement and detect periodicity from the in-and-out binary sequence. They also discuss the issue of working with sparse and incomplete observation in spatiotemporal data. Further, experimental results are provided on real movement data to verify the effectiveness of their techniques.

Climate data brings unique challenges that are different from those experienced by traditional data mining. In Chapter 3, Faghmous and Kumar refer to spatiotemporal data mining as a collection of methods that mine the data's spatiotemporal context to increase an algorithm's accuracy, scalability, or interpretability. They highlight some of the singular characteristics and challenges that STDM faces with climate data and their applications, and offer an overview of the advances in STDM and other related climate applications. Their case studies provide examples of challenges faced when mining climate data and show how effectively analyzing the spatiotemporal data context may improve the accuracy, interpretability, and scalability of existing methods.

Many scientific applications search for patterns in complex structural information. When this structural information is represented as a graph, discriminative subgraph mining can be used to discover the desired pattern. For example, the structures of chemical compounds can be stored as graphs, and with the help of discriminative subgraphs, chemists can predict which compounds are potentially toxic. In Chapter 4, Jin and Wang present their research on mining discriminative subgraph patterns from structural data. Many research studies have been devoted to developing efficient discriminative subgraph pattern-mining algorithms. Higher efficiency allows users to process larger graph datasets, and higher effectiveness enables users to achieve better results in applications. In this chapter, several existing discriminative subgraph pattern- mining algorithms are introduced, as well as an evaluation of the algorithms using real protein and chemical structure data.

The development of path knowledge discovery was motivated by problems in neuropsychiatry, where researchers needed to discover interrelationships extending across brain biology that link genotype (such as dopamine gene mutations) to phenotype (observable characteristics of organisms such as cognitive performance measures). Liu, Chu, Sabb, Parker, and Bilder present path knowledge discovery in Chapter 5. Path knowledge discovery consists of two integral tasks: 1) association path mining among concepts in multipart phenotypes that cross disciplines, and 2) fine-granularity knowledge-based content retrieval along the path(s) to permit deeper analysis. The methodology is validated using a published heritability study from cognition research and obtaining comparable results. The authors show how pheno-mining tools can greatly reduce a domain expert's time by several orders of magnitude

when searching and gathering knowledge from published literature, and can facilitate derivation of interpretable results.

Chapters 6, 7 and 8 present data mining in social media. In Chapter 6, Bhattacharyya and Wu, present "InfoSearch : A Social Search Engine" which was developed using the Facebook platform. InfoSearch leverages the data found in Facebook, where users share valuable information with friends. The user-to–content link structure in the social network provides a wealth of data in which to search for relevant information. Ranking factors are used to encourage users to search queries through InfoSearch.

As social media became more integrated into the daily lives of people, users began turning to it in times of distress. People use Twitter, Facebook, YouTube, and other social media platforms to broadcast their needs, propagate rumors and news, and stay abreast of evolving crisis situations. In Chapter 7, Landwehr and Carley discuss social media mining and its novel application to humanitarian assistance and disaster relief. An increasing number of organizations can now take advantage of the dynamic and rich information conveyed in social media for humanitarian assistance and disaster relief.

Social network analysis is very useful for discovering the embedded knowledge in social network structures. This is applicable to many practical domains such as homeland security, epidemiology, public health, electronic commerce, marketing, and social science. However, privacy issues prevent different users from effectively sharing information of common interest. In Chapter 8, Yang and Thuraisingham propose to construct a generalized social network in which only insensitive and generalized information is shared. Further, their proposed privacy-preserving method can satisfy a prescribed level of privacy leakage tolerance thatis measured independent of the privacy-preserving techniques.

Binary matrix factorization (BMF) is an important tool in dimension reduction for high-dimensional data sets with binary attributes, and it has been successfully employed in numerous applications. In Chapter 9, Jiang, Peng, Heath and Yang propose a clustering approach to updating procedures for constrained BMF where the matrix product is required to be binary. Numerical experiments show that the proposed algorithm yields better results than that of other algorithms reported in research literature.

Finally, we want to thank our authors for contributing their work to this volume, and also our reviewers for commenting on the readability and accuracy of the work. We hope that the new data mining methodologies and challenges will stimulate further research and gain new opportunities for knowledge discovery.

Los Angeles, California Wesley W. Chu
June 2013

Contents

Aspect and Entity Extraction for Opinion Mining

Lei Zhang and Bing Liu

Abstract. Opinion mining or sentiment analysis is the computational study of people's opinions, appraisals, attitudes, and emotions toward entities such as products, services, organizations, individuals, events, and their different aspects. It has been an active research area in natural language processing and Web mining in recent years. Researchers have studied opinion mining at the document, sentence and aspect levels. Aspect-level (called *aspect-based opinion mining*) is often desired in practical applications as it provides the detailed opinions or sentiments about different aspects of entities and entities themselves, which are usually required for action. Aspect extraction and entity extraction are thus two core tasks of aspect-based opinion mining. In this chapter, we provide a broad overview of the tasks and the current state-of-the-art extraction techniques.

1 Introduction

Opinion mining or sentiment analysis is the computational study of people's opinions, appraisals, attitudes, and emotions toward entities and their aspects. The entities usually refer to products, services, organizations, individuals, events, etc and the aspects are attributes or components of the entities (Liu, 2006). With the growth of *social media* (i.e., reviews, forum discussions, and blogs) on the Web, individuals and organizations are increasingly using the opinions in these media for decision making. However, people have difficulty, owing to their mental and physical limitations, producing consistent results when the amount of such information to be processed is large. Automated opinion mining is thus needed, as subjective biases and mental limitations can be overcome with an objective opinion mining system.

Lei Zhang · Bing Liu
Department of Computer Science, University of Illinois at Chicago,
Chicago, United States
e-mail: lzhang32@gmail.com, liub@cs.uic.edu

W.W. Chu (ed.), *Data Mining and Knowledge Discovery for Big Data*,
Studies in Big Data 1,
DOI: 10.1007/978-3-642-40837-3_1, © Springer-Verlag Berlin Heidelberg 2014

In the past decade, opinion mining has become a popular research topic due to its wide range of applications and many challenging research problems. The topic has been studied in many fields, including natural language processing, data mining, Web mining, and information retrieval. The survey books of Pang and Lee (2008) and Liu (2012) provide a comprehensive coverage of the research in the area. Basically, researchers have studied opinion mining at three levels of granularity, namely, document level, sentence level, and aspect level. Document level *sentiment classification* is perhaps the most widely studied problem (Pang, Lee and Vaithyanathan, 2002; Turney, 2002). It classifies an opinionated document (e.g., a product review) as expressing an overall positive or negative opinion. It considers the whole document as a basic information unit and it assumes that the document is known to be opinionated. At the sentence level, sentiment classification is applied to individual sentences in a document (Wiebe and Riloff, 2005; Wiebe et al., 2004; Wilson et al., 2005). However, each sentence cannot be assumed to be opinionated. Therefore, one often first classifies a sentence as opinionated or not opinioned, which is called *subjectivity classification*. The resulting opinionated sentences are then classified as expressing positive or negative opinions.

Although opinion mining at the document level and the sentence level is useful in many cases, it still leaves much to be desired. A positive evaluative text on a particular entity does not mean that the author has positive opinions on every aspect of the entity. Likewise, a negative evaluative text for an entity does not mean that the author dislikes everything about the entity. For example, in a product review, the reviewer usually writes both positive and negative aspects of the product, although the general sentiment on the product could be positive or negative. To obtain more fine-grained opinion analysis, we need to delve into the aspect level. This idea leads to *aspect-based opinion mining,* which was first called the *feature-based opinion mining* in Hu and Liu (2004b). Its basic task is to extract and summarize people's opinions expressed on entities and aspects of entities. It consists of three core sub-tasks.

(1) identifying and extracting entities in evaluative texts
(2) identifying and extracting aspects of the entities
(3) determining sentiment polarities on entities and aspects of entities

For example, in the sentence "*I brought a Sony camera yesterday, and its picture quality is great*," the aspect-based opinion mining system should identify the author expressed a positive opinion about the picture quality of the Sony camera. Here *picture quality* is an aspect and *Sony camera* is the entity. We focus on studying the first two tasks here. For the third task, please see (Liu, 2012). Note that some researchers use the term *feature* to mean aspect and the term *object* to mean entity (Hu and Liu, 2004a). Some others do not distinguish aspects and entities and call both of them *opinion targets* (Qiu et al., 2011; Jakob and Gurevych, 2010; Liu et al., 2012), *topics* (Li et al., 2012a) or simply *attributes* (Putthividhya and Hu, 2011) that opinions have been expressed on.

2 Aspect-Based Opinion Mining Model

In this section, we give an introduction to the aspect-based opinion mining model, and discuss the aspect-based opinion summary commonly used in opinion mining (or sentiment analysis) applications.

2.1 Model Concepts

Opinions can be expressed about anything such as a product, a service, or a person by any person or organization. We use the term *entity* to denote the target object that has been evaluated. An entity can have a set of components (or parts) and a set of attributes. Each component may have its own sub-components and its set of attributes, and so on. Thus, an entity can be hierarchically decomposed based on the *part-of* relation (Liu, 2006).

Definition (**entity**)**:** An *entity e* is a product, service, person, event, organization, or topic. It is associated with a pair, *e*: (*T*, *W*), where *T* is a hierarchy of *components* (or *parts*), *sub-components*, and so on, and *W* is a set of *attributes* of *e*. Each component or sub-component also has its own set of attributes.

Example: A particular brand of cellular phone is an entity, e.g., *iPhone*. It has a set of components, e.g., *battery* and *screen*, and also a set of attributes, e.g., *voice quality*, *size*, and *weight*. The battery component also has its own set of attributes, e.g., *battery life*, and *battery size*.

Based on this definition, an entity can be represented as a tree or hierarchy. The root of the tree is the name of the entity. Each non-root node is a component or sub-component of the entity. Each link is a *part-of* relation. Each node is associated with a set of attributes. An opinion can be expressed on any node and any attribute of the node.

Example: One can express an opinion about the iPhone itself (the root node), e.g., "*I do not like iPhone*", or on any one of its attributes, e.g., "*The voice quality of iPhone is lousy*". Likewise, one can also express an opinion on any one of the iPhone's components or any attribute of the component.

In practice, it is often useful to simplify this definition due to two reasons: First, natural language processing is difficult. To effectively study the text at an arbitrary level of detail as described in the definition is very hard. Second, for an ordinary user, it is too complex to use a hierarchical representation. Thus, we simplify and flatten the tree to two levels and use the term *aspects* to denote both components and attributes. In the simplified tree, the root level node is still the entity itself, while the second level nodes are the different aspects of the entity.

Definition (**aspect and aspect expression**)**:** The *aspects* of an entity *e* are the components and attributes of *e*. An *aspect expression* is an actual word or phrase that has appeared in text indicating an aspect.

Example: In the cellular phone domain, an aspect could be named *voice quality*. There are many expressions that can indicate the aspect, e.g., "*sound*," "*voice*," and "*voice quality*."

Aspect expressions are usually nouns and noun phrases, but can also be verbs, verb phrases, adjectives, and adverbs. We call aspect expressions in a sentence that are nouns and noun phrases *explicit aspect expressions*. For example, "*sound*" in "*The sound of this phone is clear*" is an explicit aspect expression. We call aspect expressions of the other types, *implicit aspect expressions***,** as they often imply some aspects. For example, "*large*" is an implicit aspect expression in "*This phone is too large*". It implies the aspect *size*. Many implicit aspect expressions are adjectives and adverbs, which imply some specific aspects, e.g., *expensive* (price), and *reliably* (reliability). Implicit aspect expressions are not just adjectives and adverbs. They can be quite complex, for example, "*This phone will not easily fit in pockets*". Here, "*fit in pockets*" indicates the aspect *size* (and/or *shape*).

Like aspects, an entity also has a name and many expressions that indicate the entity. For example, the brand *Motorola* (entity name) can be expressed in several ways, e.g., "*Moto*", "*Mot*" and "*Motorola*" itself.

Definition (**entity expression**)**:** An *entity expression* is an actual word or phrase that has appeared in text indicating a particular entity.

Definition (**opinion holder**)**:** The *holder* of an opinion is the person or organization that expresses the opinion.

For product reviews and blogs, opinion holders are usually the authors of the postings. Opinion holders are more important in news articles as they often explicitly state the person or organization that holds an opinion. Opinion holders are also called *opinion sources*. Some research has been done on identifying and extracting opinion holders from opinion documents (Bethard et al., 2004; Choi et al., 2005; Kim and Hovy, 2006; Stoyanov and Cardie, 2008).

We now turn to opinions. There are two main types of opinions: *regular opinions* and *comparative opinions* (Liu, 2010; Liu, 2012). Regular opinions are often referred to simply as opinions in the research literature. A comparative opinion is a relation of similarity or difference between two or more entities, which is often expressed using the comparative or superlative form of an adjective or adverb (Jindal and Liu, 2006a and 2006b).

An *opinion* (or regular opinion) is simply a positive or negative view, attitude, emotion or appraisal about an entity or an aspect of the entity from an opinion holder. Positive, negative and neutral are called *opinion orientations.* Other names for opinion orientation are *sentiment orientation*, *semantic orientation*, or *polarity*. In practice, neutral is often interpreted as no opinion. We are now ready to formally define an opinion.

Definition (opinion): An *opinion* (or *regular opinion*) is a quintuple,

$$(e_i, a_{ij}, oo_{ijkl}, h_k, t_l),$$

where e_i is the name of an entity, a_{ij} is an aspect of e_i, oo_{ijkl} is the orientation of the opinion about aspect a_{ij} of entity e_i, h_k is the opinion holder, and t_l is the time when the opinion is expressed by h_k. The opinion orientation oo_{ijkl} can be positive, negative or neutral, or be expressed with different strength/intensity levels. When an opinion is on the entity itself as a whole, we use the special aspect GENERAL to denote it.

We now put everything together to define a model of entity, a model of opinionated document, and the mining objective, which are collectively called the *aspect-based opinion mining*.

Model of Entity: An entity e_i is represented by itself as a whole and a finite set of aspects, $A_i = \{a_{i1}, a_{i2}, \ldots, a_{in}\}$. The entity itself can be expressed with any one of a final set of entity expressions $OE_i = \{oe_{i1}, oe_{i2}, \ldots, oe_{is}\}$. Each aspect $a_{ij} \in A_i$ of the entity can be expressed by any one of a finite set of aspect expressions $AE_{ij} = \{ae_{ij1}, ae_{ij2}, \ldots, ae_{ijm}\}$.

Model of Opinionated Document: An opinionated document d contains opinions on a set of entities $\{e_1, e_2, \ldots, e_r\}$ from a set of opinion holders $\{h_1, h_2, \ldots, h_p\}$. The opinions on each entity e_i are expressed on the entity itself and a subset A_{id} of its aspects.

Objective of Opinion Mining: Given a collection of opinionated documents D, discover all opinion quintuples $(e_i, a_{ij}, oo_{ijkl}, h_k, t_l)$ in D.

2.2 Aspect-Based Opinion Summary

Most opinion mining applications need to study opinions from a large number of opinion holders. One opinion from a single person is usually not sufficient for action. This indicates that some form of summary of opinions is desired. *Aspect-Based opinion summary* is a common form of opinion summary based on aspects, which is widely used in industry (see Figure 1). In fact, the discovered opinion quintuples can be stored in database tables. Then a whole suite of database and visualization tools can be applied to visualize the results in all kinds of ways for the user to gain insights of the opinions in structured forms as bar charts and/or pie charts. Researchers have also studied opinion summarization in the tradition fashion, e.g., producing a short *text summary* (Carenini et al, 2006). Such a summary gives the reader a quick overview of what people think about a product or service. A weakness of such a text-based summary is that it is not quantitative but only qualitative, which is usually not suitable for analytical purposes. For example, a traditional text summary may say "*Most people do not like this product*". However, a quantitative summary may say that 60% of the people do not like this product and 40% of them like it. In most applications, the quantitative side is crucial just like in the traditional survey research. Instead of generating a text summary directly from input reviews, we can also generate a text summary based on the mining results from bar charts and/or pie charts (see (Liu, 2012)).

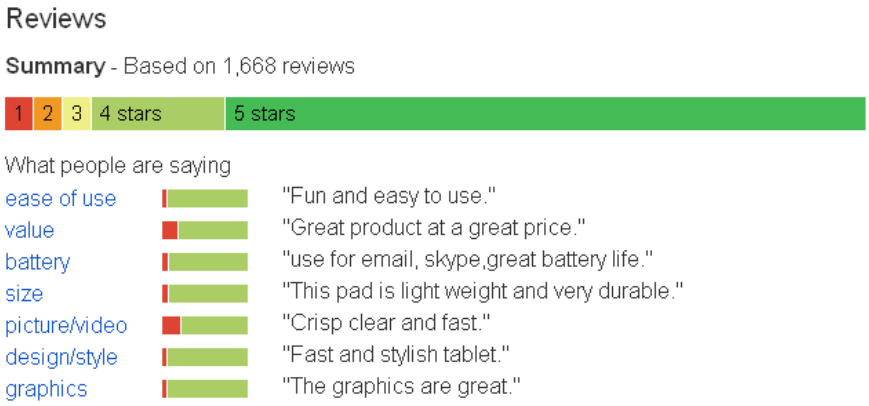


Fig. 1 Opinion summary based on product aspects of iPad (from Google Product[1])

3 Aspect Extraction

Both aspect extraction and entity extraction fall into the broad class of information extraction (Sarawagi, 2008), whose goal is to automatically extract structured information (e.g., names of persons, organizations and locations) from unstructured sources. However, traditional information extraction techniques are often developed for formal genre (e.g., news, scientific papers), which have some difficulties to be applied effectively to opinion mining applications. We aim to extract fine-grained information from opinion documents (e.g., reviews, blogs and forum discussions), which are often very noisy and also have some distinct characteristics that can be exploited for extraction. Therefore, it is beneficial to design extraction methods that are specific to opinion documents. In this section, we focus on the task of aspect extraction. Since aspect extraction and entity extraction are closely related, some ideas or methods proposed for aspect extraction can be applied to the task of entity extraction as well. In Section 4, we will discuss a special problem of entity extraction for opinion mining and some approaches for solving the problem.

Existing research on aspect extraction is mainly carried out on online reviews. We thus focus on reviews here. There are two common review formats on the Web.

Format 1 – Pros, Cons and the Detailed Review: The reviewer is asked to describe some brief Pros and Cons separately and also write a detailed/full review.

Format 2 – Free Format: The reviewer can write freely, i.e., no separation of pros and cons.

[1] `http://www.google.com/shopping`

To extract aspects from Pros and Cons in reviews of Format 1 (not the detailed review, which is the same as Format 2), many information extraction techniques can be applied. An important observation about Pros and Cons is that they are usually very brief, consisting of short phrases or sentence segments. Each sentence segment typically contains only one aspect, and sentence segments are separated by commas, periods, semi-colons, hyphens, *&*, *and*, *but*, etc. This observation helps the extraction algorithm to perform more accurately (Liu, Hu and Cheng, 2005). Since aspect extraction from Pros and Cons is relatively simple, we will not discuss it further.

We now focus on the more general case, i.e., extracting aspects from reviews of Format 2, which usually consist of full sentences.

3.1 Extraction Approaches

We introduce only the main extraction approaches for aspects (or aspect expressions) proposed in recent years. As discussed in Section 2.1, there are two types of aspect expressions in opinion documents: *explicit aspect expression* and *implicit aspect expression*. We will discuss implicit aspects in Section 3.4. In this section, we focus on explicit aspect extraction. We categorize the existing extraction approaches into three main categories: language rules, sequence models and topic models.

3.1.1 Exploiting Language Rules

Language rule-based systems have a long history of usage in information extraction. The rules are based on contextual patterns, which capture various properties of one or more terms and their relations in the text. In reviews, we can utilize the grammatical relations between aspects and opinion words or other terms to induce extraction rules.

Hu and Liu (2004a) first proposed a method to extract product aspects based on association rules. The idea can be summarized briefly by two points: (1) finding frequent nouns and noun phrases as frequent aspects. (2) using relations between aspects and opinion words to identify infrequent aspects. The basic steps of the approach are as follows.

Step 1: Find frequent nouns and noun phrases. Nouns and noun phrases are identified by a part-of-speech (POS) tagger. Their occurrence frequencies are counted, and only the frequent ones are kept. A frequency threshold is decided experimentally. The reason for using this approach is that when people comment on different aspects of a product, the vocabulary that they use usually converges. Thus, those nouns and noun phrases that are frequently talked about are usually genuine and important aspects. Irrelevant contents in reviews are often diverse, i.e., they are quite different in different reviews. Hence, those infrequent nouns are likely to be non-aspects or less important aspects.

Step 2: Find infrequent aspects by exploiting the relationships between aspects and opinion words (words that expressing positive or negative opinion, e.g., "*great*" and "*bad*"). The step 1 may miss many aspect expressions which are infrequent. This step tries to find some of them. The idea is as follows: The same opinion word can be used to describe or modify different aspects. Opinion words that modify frequent aspects can also modify infrequent aspects, and thus can be used to extract infrequent aspects. For example, "*picture*" has been found to be a frequent aspect, and we have the sentence,

"*The pictures are absolutely amazing*."

If we know that "*amazing*" is an opinion word, then "*software*" can also be extracted as an aspect from the following sentence,

"*The software is amazing*."

because the two sentences follow the same dependency pattern and "*software*" in the sentence is also a noun.

The idea of extracting frequent nouns and noun phrases as aspects is simple but effective. Blair-Goldensohn et al. (2008) refined the approach by considering mainly those noun phrases that are in sentiment-bearing sentences or in some syntactic patterns which indicate sentiments. Several filters were applied to remove unlikely aspects, for example, dropping aspects which do not have sufficient mentions along-side known sentiment words. The frequency-based idea was also utilized in (Popescu and Etzioni, 2005; Ku et al., 2006; Moghaddam and Ester, 2010; Zhu et al., 2009; Long et al., 2010).

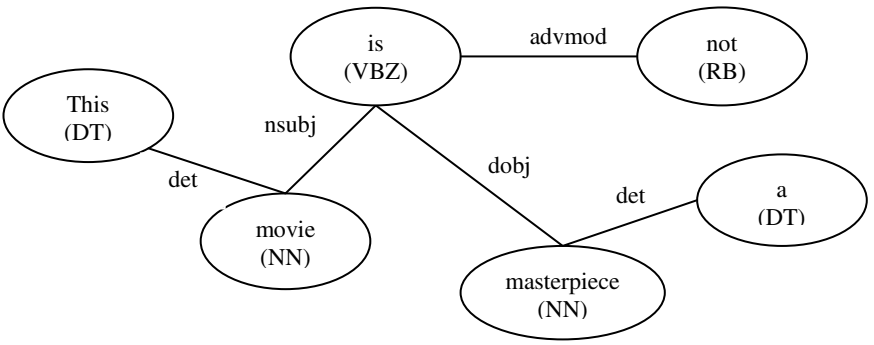


Fig. 2 Dependency grammar graph (Zhuang et al., 2006)

The idea of using the modifying relationship of opinion words and aspects to extract aspects can be generalized to using dependency relation. Zhuang et al. (2006) employed the dependency relation to extract aspect-opinion pairs from movie reviews. After parsed by a dependency parser (e.g., MINIPAR[2]

[2] http://webdocs.cs.ualberta.ca/~lindek/minipar.htm

(Lin, 1998)), words in a sentence are linked to each other by a certain dependency relation. Figure 2 shows the dependency grammar graph of an example sentence, "*This movie is not a masterpiece*", where "*movie*" and "*masterpiece*" have been labeled as aspect and opinion word respectively. A dependency relation template can be found as the sequence *"NN - nsubj - VB - dobj - NN"*. NN and VB are POS tags. *nsubj* and *dobj* are dependency tags. Zhuang et al. (2006) first identified reliable dependency relation templates from training data, and then used them to identify valid aspect-opinion pairs in test data.

In Wu et al. (2009), a phrase dependency parser was used for extracting noun phrases and verb phrases as aspect candidates. Unlike a normal dependency parser that identifies dependency of individual words only, a phrase dependency parser identifies dependency of phrases. Dependency relations have also been exploited by Kessler and Nicolov (2009).

Wang and Wang (2008) proposed a method to identify product aspects and opinion words simultaneously. Given a list of seed opinion words, a bootstrapping method is employed to identify product aspects and opinion words in an alternation fashion. Mutual information is utilized to measure association between potential aspects and opinion words and vice versa. In addition, linguistic rules are extracted to identify infrequent aspects and opinion words. The similar bootstrapping idea is also utilized in (Hai et al., 2012).

Double propagation (Qiu et al., 2011) further developed aforementioned ideas. Similar to Wang and Wang (2008), the method needs only an initial set of opinion word seeds as the input. It observed that opinions almost always have targets, and there are natural relations connecting opinion words and targets in a sentence due to the fact that opinion words are used to modify targets. Furthermore, it found that opinion words have relations among themselves and so do targets among themselves too. The opinion targets are usually aspects. Thus, opinion words can be recognized by identified aspects, and aspects can be identified by known opinion words. The extracted opinion words and aspects are utilized to identify new opinion words and new aspects, which are used again to extract more opinion words and aspects. This propagation process ends when no more opinion words or aspects can be found. As the process involves propagation through both opinion words and aspects, the method is called *double propagation*. Extraction rules are designed based on different relations between opinion words and aspects, and also opinion words and aspects themselves. Dependency grammar was adopted to describe these relations.

The method only uses a simple type of dependencies called *direct dependencies* to model useful relations. A direct dependency indicates that one word depends on the other word without any additional words in their dependency path or they both depend on a third word directly. Some constraints are also imposed. Opinion words are considered to be adjectives and aspects are nouns or noun phrases. Table 1 shows the rules for aspect and opinion word extraction. It uses *OA-Rel* to denote the relations between opinion words and aspects, *OO-Rel* between opinion words themselves and *AA-Rel* between aspects. Each relation in OA-Rel, OO-Rel

or AA-Rel can be formulated as a triple ⟨POS(w_i), R, POS(w_j)⟩, where POS(w_i) is the POS tag of word w_i, and R is the relation. For example, in an opinion sentence "*Canon G3 produces great pictures*", the adjective "*great*" is parsed as directly depending on the noun "*pictures*" through *mod*, formulated as an OA-Rel ⟨*JJ, mod, NNS*⟩. If we know "*great*" is an opinion word and are given the rule 'a noun on which an opinion word directly depends through *mod* is taken as an aspect', we can extract "*pictures*" as an aspect. Similarly, if we know "*pictures*" is an aspect, we can extract "*great*" as an opinion word using a similar rule. In a nut shell, the propagation performs four subtasks: (1) extracting aspects using opinion words, (2) extracting aspects using extracted aspects, (3) extracting opinion words using the extracted aspects, and (4) extracting opinion words using both the given and the extracted opinion words.

Table 1 Rules for aspect and opinion word extraction

	Observations	**Output**	**Examples**
$R1_1$ (OA-Rel)	$O \rightarrow O\text{-}Dep \rightarrow A$ ***s.t.*** $O \in \{O\}$, $O\text{-}Dep \in \{MR\}$, $POS(A) \in \{NN\}$	$a = A$	*The phone has a good "screen".* *good* **→** ***mod*** **→** *screen*
$R1_2$ (OA-Rel)	$O \rightarrow O\text{-}Dep \rightarrow H \leftarrow A\text{-}Dep \leftarrow A$ ***s.t.*** $O \in \{O\}$, $O/A\text{-}Dep \in \{MR\}$, $POS(A) \in \{NN\}$	$a = A$	*"iPod" is the best mp3 player.* *best* **→** ***mod*** **→** *player* **←** ***subj*** **←** *iPod*
$R2_1$ (OA-Rel)	$O \rightarrow O\text{-}Dep \rightarrow A$ ***s.t.*** $A \in \{A\}$, $O\text{-}Dep \in \{MR\}$, $POS(O) \in \{JJ\}$	$o = O$	same as $R1_1$ with *screen* as the known word and *good* as the extracted word
$R2_2$ (OA-Rel)	$O \rightarrow O\text{-}Dep \rightarrow H \leftarrow A\text{-}Dep \leftarrow A$ ***s.t.*** $A \in \{A\}$, $O/A\text{-}Dep \in \{MR\}$, $POS(O) \in \{JJ\}$	$o = O$	same as $R1_2$ with *iPod* is the known word and *best* as the extract word.
$R3_1$ (AA-Rel)	$A_{i(j)} \rightarrow A_{i(j)}\text{-}Dep \rightarrow A_{j(i)}$ ***s.t.*** $A_{j(i)} \in \{A\}$, $A_{i(j)}\text{-}Dep \in \{CONJ\}$, $POS(A_{i(j)}) \in \{NN\}$	$a = A_{i(j)}$	*Does the player play dvd with audio and "video"?* *video* **→** ***conj*** **→** *audio*
$R3_2$ (AA-Rel)	$A_i \rightarrow A_i\text{-}Dep \rightarrow H \leftarrow A_j\text{-}Dep \leftarrow A_j$ ***s.t.*** $A_i \in \{A\}$, $A_i\text{-}Dep = A_j\text{-}Dep$ OR ($A_i\text{-}Dep = subj$ AND $A_j\text{-}Dep = obj$), $POS(A_j) \in \{NN\}$	$a = A_j$	*Canon "G3" has a great len.* *len* **→** ***obj*** **→** *has* **←** ***subj*** **←** *G3*
$R4_1$ (OO-Rel)	$O_{i(j)} \rightarrow O_{i(j)}\text{-}Dep \rightarrow O_{j(i)}$ ***s.t.*** $O_{j(i)} \in \{O\}$, $O_{i(j)}\text{-}Dep \in \{CONJ\}$, $POS(O_{i(j)}) \in \{JJ\}$	$o = O_{i(j)}$	*The camera is amazing and "easy" to use.* *easy* **→** ***conj*** **→** *amazing*
$R4_2$ (OO-Rel)	$O_i \rightarrow O_i\text{-}Dep \rightarrow H \leftarrow O_j\text{-}Dep \leftarrow O_j$ ***s.t.*** $O_i \in \{O\}$, $O_i\text{-}Dep = O_j\text{-}Dep$ OR ($O_i / O_j\text{-}Dep \in \{pnmod, mod\}$), $POS(O_j) \in \{JJ\}$	$o = O_j$	*If you want to buy a sexy, "cool", accessory-available mp3 player, you can choose iPod.* *sexy* **→** ***mod*** **→** *player* **←** ***mod*** **←** *cool*

Column 1 is the rule ID, column 2 is the observed relation and the constraints that it must satisfy, column 3 is the output, and column 4 is an example. In each example, the underlined word is the known word and the word with double quotes is the extracted word. The corresponding instantiated relation is given right below the example.

OA-Rels are used for tasks (1) and (3), *AA-Rels* are used for task (2) and *OO-Rels* are used for task (4). Four types of rules are defined respectively for these four subtasks and the details are given in Table 1. In the table, *o* (or *a*) stands for the output (or extracted) opinion word (or aspect). {*O*} (or {*A*}) is the set of known opinion words (or the set of aspects) either given or extracted. *H* means any word. *POS*(*O*(or *A*)) and *O*(or *A*)-*Dep* stand for the POS tag and dependency relation of the word *O* (or *A*) respectively.{*JJ*} and {*NN*}are sets of POS tags of potential opinion words and aspects respectively. {*JJ*} contains *JJ*, *JJR* and *JJS*; {*NN*} contains *NN* and *NNS*. {*MR*} consists of dependency relations describing relations between opinion words and aspects (*mod*, *pnmod*, *subj*, *s*, *obj*, *obj2* and *desc*). {*CONJ*} contains *conj* only. The arrows mean dependency. For example, *O* → *O-Dep* → *A* means *O* depends on *A* through a syntactic relation *O-Dep*. Specifically, it employs $R1_i$ to extract aspects (*a*) using opinion words (*O*), $R2_i$ to extract opinion words (*o*) using aspects (*A*), $R3_i$ to extract aspects (*a*) using extracted aspects (A_i) and $R4_i$ to extract opinion words (*o*) using known opinion words (O_i). Take $R1_1$ as an example. Given the opinion word *O*, the word with the POS tag *NN* and satisfying the relation *O-Dep* is extracted as an aspect.

The double propagation method works well for medium-sized corpuses, but for large and small corpora, it may result in low precision and low recall. The reason is that the patterns based on *direct dependencies* have a large chance of introducing noises for large corpora and such patterns are limited for small corpora. To overcome the weaknesses, Zhang et al. (2010) proposed an approach to extend double propagation. It consists of two steps: *aspect extraction* and *aspect ranking*. For aspect extraction, it still adopts double propagation to populate aspect candidates. However, some new linguistic patterns (e.g., *part-whole* relation patterns) are introduced to increase recall. After extraction, it ranks aspect candidates by aspect importance. That is, if an aspect candidate is genuine and important, it will be ranked high. For an unimportant aspect or noise, it will be ranked low. It observed that there are two major factors affecting the aspect importance: *aspect relevance* and *aspect frequency*. The former describes how likely an aspect candidate is a genuine aspect. There are three clues to indicate aspect relevance in reviews. The first clue is that an aspect is often modified by multiple opinion words. For example, in the mattress domain, "*delivery*" is modified by "*quick*" "*cumbersome*" and "*timely*". It shows that reviewers put emphasis on the word "*delivery*". Thus, "*delivery*" is a likely aspect. The second clue is that an aspect can be extracted by multiple part-whole patterns. For example, in car domain, if we find following two sentences, "*the engine of the car*" and "*the car has a big engine*", we can infer that "*engine*" is an aspect for car, because both sentences contain part-whole relations to indicate "*engine*" is a part of "*car*". The third clue is that an aspect can be extracted by a combination of opinion word modification relation, part-whole pattern or other linguistic patterns. If an aspect candidate is not only modified by opinion words but also extracted by part-whole pattern, we can infer that it is a genuine aspect with high confidence. For example, for sentence "*there is a bad hole in the mattress*", it strongly

indicates that "*hole*" is an aspect for a mattress because it is modified by opinion word "*bad*" and also in the part-whole pattern. What is more, there are mutual enforcement relations between opinion words, linguistic patterns, and aspects. If an adjective modifies many genuine aspects, it is highly possible to be a good opinion word. Likewise, if an aspect candidate can be extracted by many opinion words and linguistic patterns, it is also highly likely to be a genuine aspect. Thus, Zhang et al. utilized the HITS algorithm (Klernberg, 1999) to measure aspect relevance. Aspect frequency is another important factor affecting aspect ranking. It is desirable to rank those frequent aspects higher than infrequent aspects. The final ranking score for a candidate aspect is the score of aspect relevancy multiplied by the log of aspect frequency.

Liu et al. (2012) also utilized the relation between opinion word and aspect to perform extraction. However, they formulated the opinion relation identification between aspects and opinion words as a word alignment task. They employed the word-based translation model (Brown et al., 1993) to perform monolingual word alignment. Basically, the associations between aspects and opinion words are measured by translation probabilities, which can capture opinion relations between opinion words and aspects more precisely and effectively than linguistic rules or patterns.

Li et al., (2012a) proposed a domain adaption method to extract opinion words and aspects together across domains. In some cases, it has no labeled data in the target domain but a plenty of labeled data in the source domain. The basic idea is to leverage the knowledge extracted from the source domain to help identify aspects and opinion words in the target domain. The approach consists of two main steps: (1) identify some common opinion words as seeds in the target domain (e.g., "*good*", "*bad*"). Then, high-quality opinion aspect seeds for the target domain are generated by mining some general syntactic relation patterns between the opinion words and aspects from the source domain. (2) a bootstrapping method called *Relational Adaptive bootstrapping* is employed to expand the seeds. First, a cross-domain classifier is trained iteratively on labeled data from the source domain and newly labeled data from the target domain, and then used to predict the labels of the target unlabeled data. Second, top predicted aspects and opinion words are selected as candidates based on confidence. Third, with the extracted syntactic patterns in the previous iterations, it constructs a bipartite graph between opinion words and aspects extracted from the target domain. A graph-based score refinement algorithm is performed on the graph, and the top candidates are added into aspect list and opinion words list respectively.

Besides exploiting relations between aspect and opinion words discussed above, Popescu and Etzioni (2005) proposed a method to extract product aspects by utilizing a *discriminator* relation in context, i.e., the relation between aspects and product class. They first extract noun phrases with high frequency from reviews as candidate product aspects. Then they evaluate each candidate by computing a pointwise mutual information (PMI) score between the candidate and some *meronymy discriminators* associated with the product class. For example, for "*scanner*", the meronymy discriminators for the scanner class are patterns such as

"*of scanner*", "*scanner has*", "*scanner comes with*", etc. The PMI measure is calculated by searching the Web. The equation is as follows.

$$PMI(a,d) = \frac{hits(a \wedge d)}{hits(a)hits(d)} \quad (1)$$

where a is a candidate aspect and d is a discriminator. Web search is used to find the number of hits of individual terms and also their co-occurrences. The idea of this approach is clear. If the PMI value of a candidate aspect is too low, it may not be a component or aspect of the product because a and d do not co-occur frequently. The algorithm also distinguishes components/parts from attributes using WordNet[3]'s *is-a* hierarchy (which enumerates different kinds of properties) and morphological cues (e.g., "*-iness*", *"-ity*" suffixes).

Kobayashi et al. (2007) proposed an approach to extract aspect-evaluation (aspect-opinion expression) and aspect-of relations from blogs, which also makes use of association between aspect, opinion expression and product class. For example, in aspect-evaluation pair extraction, evaluation expression is first determined by a dictionary look-up. Then, syntactic patterns are employed to find its corresponding aspect to form the candidate pair. The candidate pairs are tested and validated by a classifier, which is trained by incorporating two kinds of information: contextual and statistical clues in corpus. The contextual clues are syntactic relations between words in a sentence, which can be determined by the dependency grammar, and the statistical clues are normal co-occurrences between aspects and evaluations.

3.1.2 Sequence Models

Sequence models have been widely used in information extraction tasks and can be applied to aspect extraction as well. We can deem aspect extraction as a sequence labeling task, because product aspects, entities and opinion expressions are often interdependent and occur at a sequence in a sentence. In this section, we will introduce two sequence models: Hidden Markov Model (Rabiner, 1989) and Conditional Random Fields (Lafferty et al., 2001).

Hidden Markov Model

Hidden Markov Model (HMM) is a directed sequence model for a wide range of state series data. It has been applied successfully to many sequence labeling problems such as named entity recognition (NER) in information extraction and POS tagging in natural language processing. A generic HMM model is illustrated in Figure 3.

[3] http://wordnet.princeton.edu

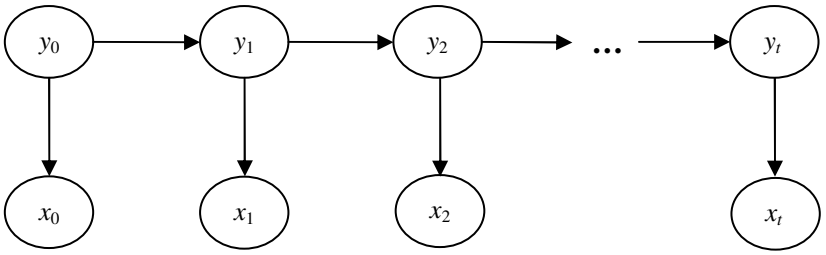


Fig. 3 Hidden Markov model

We have

$$Y = < y_0 , y_1 , \ldots y_t > = \text{hidden state sequence}$$

$$X = < x_0 , x_1 , \ldots x_t > = \text{observation sequence}$$

HMM models a sequence of observations X by assuming that there is a *hidden* sequence of states Y. Observations are dependent on states. Each state has a probability distribution over the possible observations. To model the joint distribution $p(y, x)$ tractably, two independence assumptions are made. First, it assumes that state y_t only depends on its immediate predecessor state y_{t-1}. y_t is independent of all its ancestor y_1, y_2, y_3, … , y_{t-2}. This is also called the *Markov* property. Second, the observation x_t only depends on the current state y_t. With these assumptions, we can specify HMM using three probability distributions: $p(y_0)$ over initial state, state transition distribution $p(y_t \mid y_{t-1})$ and observation distribution $p(x_t \mid y_t)$. That is, the joint probability of a state sequence Y and an observation sequence X factorizes as follows.

$$p(Y, X) = \prod_{t=1}^{t} p(y_t \mid y_{t-1}) p(x_t \mid y_t) \quad (2)$$

where we write the initial state distribution $p(y_1)$ as $p(y_1|y_0)$.

Given some observation sequences, we can learn the model parameter of HMM that maximizes the observation probability. That is, the learning of HMM can be done by building a model to best fit the training data. With the learned model, we can find an optimal state sequence for new observation sequences.

In aspect extraction, we can regard words or phrases in a review as observations and aspects or opinion expressions as underlying states. Jin et al. (2009a and 2009b) utilized lexicalized HMM to extract product aspects and opinion expressions from reviews. Different from traditional HMM, they integrate linguistic features such as part-of-speech and lexical patterns into HMM. For example, an observable state for the lexicalized HMM is represented by a pair ($word_i$, POS($word_i$)), where POS($word_i$) represents the part-of-speech of $word_i$.

Conditional Random Fields

One limitation of HMM is that its assumptions may not be adequate for real-life problems, which leads to reduced performance. To address the limitation, linear-chain Conditional Random fields (CRF) (Lafferty et al., 2001; Sutton and McCallum, 2006) is proposed as an undirected sequence model, which models a conditional probability $p(Y|X)$ over hidden sequence Y given observation sequence X. That is, the conditional model is trained to label an unknown observation sequence X by selecting the hidden sequence Y which maximizes $p(Y|X)$. Thereby, the model allows relaxation of the strong independence assumptions made by HMM. The linear-chain CRF model is illustrated in Figure 4.

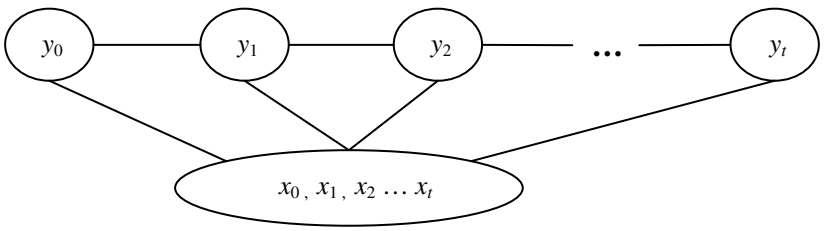


Fig. 4 Linear chain Conditional Random fields

We have

$$Y = < y_0, y_1, \ldots y_t > = \text{hidden state sequence}$$

$$X = < x_0, x_1, \ldots x_t > = \text{observation sequence}$$

The conditional distribution $p(Y|X)$ takes the form

$$p(Y \mid X) = \frac{1}{Z(X)} \exp\{\sum_{k=1}^{k} \lambda_k f_k(y_t, y_{t-1}, x_t)\} \tag{3}$$

where $Z(X)$ is a normalization function

$$Z(X) = \sum_{y} \exp\{\sum_{k=1}^{k} \lambda_k f_k(y_t, y_{t-1}, x_t)\} \tag{4}$$

CRF introduces the concept of *feature function*. Each feature function has the form $f_k(y_t, y_{t-1}, x_t)$ and λ_k is its corresponding weight. Figure 4 indicates that CRF makes independence assumption among Y, but not among X. Note that one argument for feature function f_k is the vector x_t which means each feature function can depend on observation X from any step. That is, all the components of the global observations X are needed in computing feature function f_k at step t. Thus, CRF can introduce more features than HMM at each step.

Jakob and Gurevych (2010) utilzied CRF to extract opinion targets (or aspects) from sentences which contain an opinion expression. They emplyed the following features as input for the CRF-based approach.

Token: This feature represents the string of the current token.

Part of Speech: This feature represents the POS tag of the current token. It can provide some means of lexical disambiguation.

Short Dependency Path: Direct dependency realtions show accurate connections between a target and an opinion expression. Thus, all tokens which have a direct dependency relation to an opinion expression in a sentence are labelled.

Word Distance: Noun phrases are good candidates for opinion targets in product reviews. Thus token(s) in the closest noun phrase regarding word distance to each opinion expression in a sentence are labelled.

Jakob and Gurevych represented the possible labels following the Inside-Outside-Begin (IOB) labelling schema: *B-Target*, identifying the beginning of an opinion target; *I-Target*, identifying the continuation of a target, and *O* for other (non-target) tokens.

Similar work has been done in (Li et al., 2010a). In order to model the long distance dependency with conjunctions (e.g., "*and*", "*or*", "*but*") at the sentence level and deep syntactic dependencies for aspects, positive opinions and negative opinions, they used the skip-tree CRF models to detect product aspects and opinions.

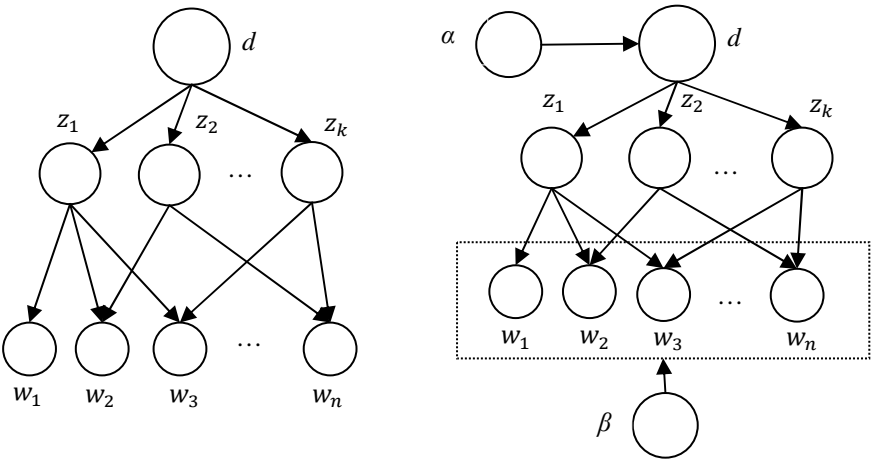
3.1.3 Topic Models

Topic models are widely applied in natural language processing and text mining. They are based on the idea that documents are mixtures of topics, and each topic is a probability distribution of words. A topic model is a *generative* model for documents. Generally, it specifies a probabilistic procedure by which documents can be generated. Assuming constructing a new document, one chooses a distribution D_i over topics. Then, for each word in that document, one chooses a topic randomly according to D_i and draws a word from the topic. Standard statistical techniques can be used to invert the procedure and infer the set of topics that were responsible for generating a collection of documents. Naturally, topic models can be applied to aspect extraction. We can deem that each aspect is a unigram language model, i.e., a multinomial distribution over words. Although such a representation is not as easy to interpret as aspects, its advantage is that different words expressing the same or related aspects (more precisely aspect expressions) can be automatically grouped together under the same aspect. Currently, a great deal of research has been done on aspect extraction using topic models. They basically adapted and extended the Probabilistic Latent Semantic Analysis (pLSA) model (Hofmann, 2001) and the Latent Dirichlet Allocation (LDA) model (Blei et al., 2003).

Probabilistic Latent Semantic Analysis

pLSA is also known as Probabilistic Latent Semantic Indexing (PLSI). It is proposed in (Hofmann, 2001), which uses a generative latent class model to perform a probabilistic mixture decomposition.

Figure 5(a) illustrate graphical model of pLSA. In the figure, d represents a document, z_i represents a latent topic (assuming K topics overall), and w_j represents a word, which are modeled by the parameters ρ, θ, φ respectively, where ρ is the probability of choosing document d, θ is the distribution $p(z_i|d)$ of topics in document d and φ is the distribution $p(w_j|z_i)$ of the word w_j in latent topic z_i. The ρ and φ are observable variables and the topic variable θ is a latent variable.



(a) pLSA (b) LDA

Fig. 5 PLSA and LDA topic models

The generation of a word by pLSA is defined as follows.

(1) choose document $d \sim \rho$

(2) choose topic $z_i \sim \theta$

(3) choose word $w_j \sim \varphi$

The probability of observed word w_j in a document d is then defined by the mixture of equation (5):

$$p(w_j \mid d) = \sum_{z=1}^{k} \theta(z)\varphi(w_j) \tag{5}$$

The joint probability of observing all words in document d is as follows:

$$p(d, w_j) = \rho(d) \prod_{j=1}^{n} p(w_j \mid d)^{c_j} \tag{6}$$

where c_j is the count of word w_j occur in document d.

And the joint probability of observing the document collection is given by the following equation (assuming m documents overall).

$$p(D) = \prod_{i=1}^{m} p(d_i) \tag{7}$$

Obviously, the main parameters of the model are θ and φ. They can be estimated by Expectation Maximization (EM) algorithm (Dempster et al., 1977), which is used to calculate maximum likelihood estimates of the parameters.

For aspect extraction task, we can regard product aspects as latent topics in opinion documents. Lu et al. (2009) proposed a method for aspect discovery and grouping in short comments. They assume that each review can be parsed into opinion phrases of the format < *head term*, *modifier* > and incorporate such structure of phrases into the pLSA model, using the co-occurrence information of head terms and their modifiers. Generally, the head term is an aspect, and the modifier is opinion word, which expresses some opinion towards the aspect. The proposed approach defines k unigram language models: $\Theta = \{\theta_1, \theta_2, \ldots, \theta_k\}$ as k topic models, each is a multinomial distribution of head terms, capturing one aspect. Note that each modifier could be represented by a set of head terms that it modifies as the following equations:

$$d(w_m) = \{w_h \mid (w_m, w_h) \in T\} \tag{8}$$

where w_h is the head term and w_m is the modifier.

Actually, a modifier can be regarded as a sample of the following mixture model.

$$p_{d(w_m)}(w_h) = \sum_{j=1}^{k} [\pi_{d(w_m),j} p(w_h \mid \theta_j)] \tag{9}$$

where $\pi_{d(w_m),j}$ is a modifier-specific mixing weight for the j-th aspect, which sums to one. The log-likelihood of the collection of modifiers V_m is

$$\log p(V_m \mid \Delta) =$$

$$\sum_{w_m \in v_m} \sum_{w_h \in v_h} \{c(w_h, d(w_m)) \times \log \sum_{j=1}^{k} [\pi_{d(w_m),j} p(w_h \mid \theta_j)]\} \tag{10}$$

where $c(w_h, d(w_m))$ is the number of co-occurrences of head term w_h with modifiers w_m, and Δ is the set of all model parameters.

Using the EM algorithm, k topic models can be estimated and aspect expressions can be grouped. In addition, Lu et al. use conjugate prior to incorporate human knowledge to guide the clustering of aspects. Since the proposed method models the co-occurrence of head terms at the level of the modifiers they use, it can use more meaningful syntactic relations.

Moghaddam and Ester (2011) extended the above pLSA model by incorporating latent rating information for reviews into the model to extract aspects and their corresponding ratings.

However, the main drawback of the pLSA method is that it is inherently transductive, i.e., there is no direct way to apply the learned model to new documents. In pLSA, each document d in the collection is represented as a mixture coefficients θ, but it does not define such representation for documents outside the collection.

Latent Dirichlet Allocation (LDA)

To address the limitation of pLSA, the Bayesian LDA model is proposed in (Blei et al., 2003). It extends pLSA by adding priors to the parameters θ and φ. In LDA, a prior Dirichlet distribution Dir (α) is added for θ and a prior Dirichlet distribution Dir (β) is added for φ. The generation of a document collection is started by sampling a word distribution φ from Dir (β) for each latent topic. Then each document d in LDA is assumed to be generated as follows.

(1) choose distribution of topics $\theta \sim Dir(\alpha)$

(2) choose distribution of words $\varphi \sim Dir(\beta)$

(3) for each word w_j in document d
- choose topic $z_i \sim \theta$
- choose word $w_j \sim \varphi$

The model is represented in Figure 5 (b). LDA has only two parameters: α and β, which prevent it from overfitting. Exact inference in such a model is intractable and various approximations have been considered, such as the variational EM method and the Markov Chain Monte Carlo (MCMC) algorithm (Gilks et al.,1996). Note that, compared with pLSA, LDA has a stronger generative power, as it describes how to generate topic distribution θ for an unseen document d.

LDA based topic models have been used for aspect extraction by several researchers. Titov and McDonald (2008a) pointed that global topic models such as pLSA and LDA might not be suitable for detecting aspects. Both pLSA and LDA use the bag-of-words representation of documents, which depends on topic distribution differences and word co-occurrence among documents to identify topics and word probability distribution in each topic. However, for opinion documents such as reviews about a particular type of products, they are quite homogenous. That is, every document talks about the same aspects, which makes global topic models ineffective and are only effective for discovering entities (e.g., brands or product names). In order to tackle this problem, they proposed

Multi-grain LDA (MG-LDA) to discover aspects, which models two distinct types of topics: global topics and local topics. As in pLSA and LDA, the distribution of global topics is fixed for a document (review). However, the distribution of local topics is allowed to vary across documents. A word in a document is sampled either from the mixture of global topics or from the mixture of local topics specific for the local context of the word. It is assumed that aspects will be captured by local topics and global topics will capture properties of reviewed items. For example, a review of a London hotel: "… *public transport in London is straightforward, the tube station is about an 8 minute walk … or you can get a bus for £1.50"*. The review can be regarded as a mixture of global topic *London* (words: "*London*", "*tube*", "*£*") and the local topic (aspect) *location* (words: "*transport*", "*walk*", "*bus*").

MG-LDA can distinguish local topics. But due to the many-to-one mapping between local topics and ratable aspects, the correspondence is not explicit. It lacks direct assignment from topics to aspects. To resolve the issue, Titov and McDonald (2008b) extended the MG-LDA model and constructed a joint model of text and aspect ratings, which is called the Multi-Aspect Sentiment model (MAS). It consists of two parts. The first part is based on MG-LDA to build topics what are representative of ratable aspects. The second part is a set of classifiers (sentiment predictors) for each aspect, which attempt to infer the mapping between local topics and aspects with the help of aspect-specific ratings provided along with the review text. Their goal is to use the rating information to identity more coherent aspects.

The idea of LDA has also been applied and extended in (Branavan et al., 2008; Lin and He, 2009; Brody and Elhadad, 2010; Zhao et al., 2010; Wang et al., 2010; Jo and Oh, 2011; Sauper et al., 2011; Moghaddam and Ester, 2011; Mukajeee and Liu, 2012). Branavan used the aspect descriptions as keyphrases in Pros and Cons of review *Format 1* to help finding aspects in the detailed review text. Keyphrases are clustered based on their distributional and orthographic properties, and a hidden topic model is applied to the review text. Then, a final graphical model integrates both of them. Lin and He (2009) proposed a joint topic-sentiment model (JST), which extends LDA by adding a sentiment layer. It can detect aspect and sentiment simultaneously from text. Brody and Elhadad (2010) proposed to identify aspects using a local version of LDA, which operates on sentences, rather than documents and employs a small number of topics that correspond directly to aspects. Zhao et al. (2010) proposed a MaxEnt-LDA hybrid model to jointly discover both aspect words and aspect-specific opinion words, which can leverage syntactic features to help separate aspects and opinion words. Wang et al. (2010) proposed a regression model to infer both aspect ratings and aspect weights at the level of individual reviews based on learned latent aspects. Jo and Oh (2011) proposed an *Aspect and Sentiment Unification* Model (ASUM) to model sentiments toward different aspects. Sauper et al. (2011) proposed a joint model, which worked only on short snippets already extracted from reviews. It combined topic modeling with a HMM, where the HMM models the sequence of words with

types (aspect, opinion word, or background word). Moghaddam and Ester (2011) proposed a model called ILDA, which is based on LDA and jointly models latent aspects and rating. ILDA can be viewed as a generative process that first generates an aspect and subsequently generates its rating. In particular, for generating each opinion phrase, ILDA first generates an aspect a_m from an LDA model. Then it generates a rating r_m conditioned on the sampled aspect a_m. Finally, a head term t_m and a sentiment s_m are drawn conditioned on a_m and r_m, respectively. Mukajeee and Liu (2012) proposed two models (SAS and ME-SAS) to jointly model both aspects and aspect specific sentiments by using seeds to discover aspects in an opinion corpus. The seeds reflect the user needs to discover specific aspects.

Other closely related work with topic model is the topic-sentiment model (TSM). Mei et al. (2007) proposed it to perform joint topic and sentiment modeling for blogs, which uses a positive sentiment model and a negative sentiment model in additional to aspect models. They do sentiment analysis on documents level and not on aspect level. In (Su et al., 2008), the authors also proposed a clustering based method with mutual reinforcement to identify aspects. Similar work has been done in (Scaffidi et al., 2007), they proposed a language model approach for product aspect extraction with the assumption that product aspects are mentioned more often in a product review than they are mentioned in general English text. However, statistics may not be reliable when the corpus is small.

In summary, topic modeling is a powerful and flexible modeling tool. It is also very nice conceptually and mathematically. However, it is only able to find some general/rough aspects, and has difficulty in finding fine-grained or precise aspects. We think it is too statistics centric and come with its limitations. It could be fruitful if we can shift more toward natural language and knowledge centric for a more balanced approach.

3.1.4 Miscellaneous Methods

Yi et al. (2003) proposed a method for aspect extraction based on the likelihood-ratio test. Bloom et al. (2007) manually built a taxonomy for aspects, which indicates aspect type. They also constructed an aspect list by starting with a sample of reviews that the list would apply to. They examined the seed list manually and used WordNet to suggest additional terms to add to the list. Lu et al. (2010) exploited the online ontology Freebase[4] to obtain aspects to a topic and used them to organize scattered opinions to generate structured opinion summaries. Ma and Wan (2010) exploited Centering theory (Grosz et al., 1995) to extract opinion targets from news comments. The approach uses global information in news articles as well as contextual information in adjacent sentences of comments. Ghani et al. (2006) formulated aspect extraction as a classification problem and used both traditional supervised learning and semi-supervised learning methods to extract product aspects. Yu et al. (2011) used a

[4] http://www.freebase.com

partially supervised learning method called one-class SVM to extract aspects. Using one-class SVM, one only needs to label some positive examples, which are aspects. In their case, they only extracted aspects from Pros and Cons of the reviews. Li et al. (2012b) formulated aspect extraction as a shallow semantic parsing problem. A parse tree is built for each sentence and structured syntactic information within the tree is used to identify aspects.

3.2 Aspect Grouping and Hierarchy

It is common that people use different words and expressions to describe the same aspect. For example, *photo* and *picture* refer to the same aspect in digital camera reviews. Although topic models (discussed in Section 3.1.3) can identify and group aspects to some extent, the results are not fine-grained because such models are based on word co-occurrences rather than word semantic meanings. As a result, a topic is often a list of related words about a general topic rather than a set of words referring to the same aspect. For example, a topic about *battery* may contain words like *life*, *battery*, *charger*, *long*, and *short*. We can clearly see that these words do not mean the same thing, although they may co-occur frequently. Alternatively, we can extract aspect expressions first and then group them into different aspect categories.

Grouping aspect expressions indicating the same aspect are essential for opinion applications. Although WordNet and other thesaurus dictionaries can help, they are far from sufficient due to the fact that many synonyms are domain dependent. For example, *picture* and *movie* are synonyms in movie reviews, but they are not synonyms in digital camera reviews as *picture* is more related to *photo* while *movie* refers to *video*. It is also important to note that although most aspect expressions of an aspect are domain synonyms, they are not always synonyms. For example, "*expensive*" and "*cheap*" can both indicate the aspect *price* but they are not synonyms of *price*.

Liu, Hu and Cheng (2005) attempted to solve the problem by using the WordNet synonym sets, but the results were not satisfactory because WordNet is not sufficient for dealing with domain dependent synonyms. Carenini et al. (2005) also proposed a method to solve this problem in the context of opinion mining. Their method is based on several similarity metrics defined using string similarity, synonyms and distances measured using WordNet. However, it requires a taxonomy of aspects to be given beforehand for a particular domain. The algorithm merges each discovered aspect expression to an aspect node in the taxonomy.

Guo et al. (2009) proposed a multilevel latent semantic association technique (called mLSA) to group product aspect expressions. At the first level, all the words in product aspect expressions are grouped into a set of concepts/topics using LDA. The results are used to build some latent topic structures for product aspect expressions. At the second level, aspect expressions are grouped by LDA

again according to their latent topic structures produced from level 1 and context snippets in reviews.

Zhai et al. (2010) proposed a semi-supervised learning method to group aspect expressions into the user specified aspect groups or categories. Each group represents a specific aspect. To reflect the user needs, they first manually label a small number of seeds for each group. The system then assigns the rest of the discovered aspect expressions to suitable groups using semi-supervised learning based on labeled seeds and unlabeled examples. The method used the Expectation-Maximization (EM) algorithm. Two pieces of prior knowledge were used to provide a better initialization for EM, i.e., (1) aspect expressions sharing some common words are likely to belong to the same group, and (2) aspect expressions that are synonyms in a dictionary are likely to belong to the same group. Zhai et al. (2011) further proposed an unsupervised method, which does not need any pre-labeled examples. Besides, it is further enhanced by lexical (or WordNet) similarity. The algorithm also exploited a piece of natural language knowledge to extract more discriminative distributional context to help grouping.

Mauge et al. (2012) used a maximum entropy based clustering algorithm to group aspects in a product category. It first trains a maximum-entropy classifier to determine the probability p that two aspects are synonyms. Then, an undirected weighted graph is constructed. Each vertex represents an aspect. Each edge weight is proportional to the probability p between two vertices. Finally, approximate graph partitioning methods are employed to group product aspects.

Closely related to aspect grouping, aspect hierarchy is to present product aspects as a tree or hierarchy. The root of the tree is the name of the entity. Each non-root node is a component or sub-component of the entity. Each link is a *part-of* relation. Each node is associated with a set of product aspects. Yu et al. (2011b) proposed a method to create aspect hierarchy. The method starts from an initial hierarchy and inserts the aspects into it one-by-one until all the aspects are allocated. Each aspect is inserted to the optimal position by semantic distance learning. Wei and Gulla (2010) studied the sentiment analysis based on aspect hierarchy trees.

3.3 Aspect Ranking

A product may have hundreds of aspects. Sometimes, we need to identify important one from reviews, which are more influential for people's decision making. Zhang et al. (2010) proposed a method to rank product aspects. They rank candidate aspects based on aspect importance which consists of two factors: aspect relevancy and aspect frequency. Aspect relevance indicates the aspect's correctness and aspect frequency is the occurrence frequency of an aspect in reviews. As discussed in Section 3.1.1, Zhang et al. modeled mutual enforcement relation between aspects and aspect indictors (e.g., opinion words and relation patterns) in a bipartite graph utilizing Web page ranking algorithm HITS. Aspects only have authority scores and aspect indicators only have hub scores. If an aspect candidate has a high authority score, it is considered as a highly relevant aspect. Likewise, if an aspect indicator has a high hub score, it is considered as a good

aspect indicator. The final ranking score of a candidate aspect is the multiplication of the aspect relevancy score (authority score) and logarithm of aspect frequency.

Yu et al. (2011a) showed the important aspects are identified according to two observations: the important aspects of a product are usually commented by a large number of consumers and consumers' opinions on the important aspects greatly influence their overall ratings on the product. Given reviews of a product, they first identify product aspects by a shallow dependency parser and determine opinions on these aspects via a sentiment classifier. They then develop an aspect ranking algorithm to identify the important aspects by considering the aspect frequency and the influence of opinions given to each aspect on their overall opinions.

Liu et al. (2012) proposed a graph-based algorithm to compute the confidence of each opinion target and its ranking. They argued that the ranking of a candidate is determined by two factors: *opinion relevancy* and *candidate importance*. To model these two factors, a bipartite graph (similar to that in Zhang et al., 2010) is constructed. An iterative algorithm based on the graph is proposed to compute candidate confidences. Then the candidates with high confidence scores are extracted as opinion targets. Similar work has also been reported in (Li et al., 2012a).

3.4 Mapping Implicit Aspect Expressions

There are many types of implicit aspect expressions. Adjectives are perhaps the most common type. Many adjectives modify or describe some specific attributes or properties of entities. For example, the adjective "*heavy*" usually describes the aspect *weight* of an entity. "*Beautiful*" is normally used to describe (positively) the aspect *look* or *appearance* of an entity. By no means, however, does this say that these adjectives only describe such aspects. Their exact meanings can be domain dependent. For example, "*heavy*" in the sentence "*the traffic is heavy*" does not describe the *weight* of the traffic. Note that some implicit aspect expressions are very difficult to extract and to map, e.g., "*fit in pockets*" in the sentence "*This phone will not easily fit in pockets*".

Limited research has been done on mapping implicit aspects to their explicit aspects. In Su et al. (2008), a clustering method was proposed to map implicit aspect expressions, which were assumed to be sentiment words, to their corresponding explicit aspects. The method exploits the mutual reinforcement relationship between an explicit aspect and a sentiment word forming a co-occurring pair in a sentence. Such a pair may indicate that the sentiment word describes the aspect, or the aspect is associated with the sentiment word. The algorithm finds the mapping by iteratively clustering the set of explicit aspects and the set of sentiment words separately. In each iteration, before clustering one set, the clustering results of the other set is used to update the pairwise similarity of the set. The pairwise similarity in a set is determined by a linear combination of intra-set similarity and inter-set similarity. The intra-set similarity of two items is the traditional similarity. The inter-set similarity of two items is computed based

on the degree of association between aspects and sentiment words. The association (or mutual reinforcement relationship) is modeled using a bipartite graph. An aspect and an opinion word are linked if they have co-occurred in a sentence. The links are also weighted based on the co-occurrence frequency. After the iterative clustering, the strong links between aspects and sentiment word groups form the mapping.

In Hai et al. (2011), a two-phase co-occurrence association rule mining approach was proposed to match implicit aspects (which are also assumed to be sentiment words) with explicit aspects. In the first phase, the approach generates association rules involving each sentiment word as the condition and an explicit aspect as the consequence, which co-occur frequently in sentences of a corpus. In the second phase, it clusters the rule consequents (explicit aspects) to generate more robust rules for each sentiment word mentioned above. For application or testing, given a sentiment word with no explicit aspect, it finds the best rule cluster and then assigns the representative word of the cluster as the final identified aspect.

Fei et al. (2012) focused on finding implicit aspects (mainly nouns) indicated by opinion adjectives, e.g., to identify *price*, *cost*, etc., for adjective *expensive*. A dictionary-based method was proposed, which tries to identify attribute nouns from the dictionary gloss of the adjective. They formulated the problem as a collective classification problem, which can exploit lexical relations of words (e.g., synonyms, antonyms, hyponym and hypernym) for classification.

Some other related work for implicit aspect mapping includes those in (Wang and Wang, 2008; Yu et al., 2011b).

3.5 Identifying Aspects That Imply Opinions

Zhang and Liu (2011a) found that in some domains nouns and noun phrases that indicate product aspects may also imply opinions. In many such cases, these nouns are not subjective but objective. Their involved sentences are also objective sentences but imply positive or negative opinions. For example, the sentence in a mattress review "*Within a month, a valley formed in the middle of the mattress*." Here "*valley*" indicates the quality of the mattress (a product aspect) and also implies a negative opinion. Identifying such aspects and their polarities is very challenging but critical for effective opinion mining in these domains.

Zhang and Liu observed that for a product aspect with an implied opinion, there is either no adjective opinion word that modifies it directly or the opinion words that modify it have the same opinion orientation.

Observation: No opinion adjective word modifies the opinionated product aspect ("*valley*"):

"*Within a month, a valley formed in the middle of the mattress.*"

Observation: An opinion adjective modifies the opinionated product aspect:

"*Within a month, a bad valley formed in the middle of the mattress*."

Here, the adjective "*bad*" modifies "*valley*". It is unlikely that a positive opinion word will also modify "*valley*" in another sentence, e.g., "*good valley*" in this context. Thus, if a product aspect is modified by both positive and negative opinion adjectives, it is unlikely to be an opinionated product aspect.

Based on these observations, they designed the following two steps to identify noun product aspects which imply positive or negative opinions:

Step 1: *Candidate Identification*: This step determines the surrounding sentiment context of each noun aspect. The intuition is that if an aspect occurs in negative (respectively positive) opinion contexts significantly more frequently than in positive (or negative) opinion contexts, we can infer that its polarity is negative (or positive). A statistical test (test for population proportion) is used to test the significance. This step thus produces a list of candidate aspects with positive opinions and a list of candidate aspects with negative opinions.

Step 2: *Pruning*: This step prunes the two lists. The idea is that when a noun product aspect is directly modified by both positive and negative opinion words, it is unlikely to be an opinionated product aspect.

3.6 Identifying Resource Noun

Liu (2010) point out that there are some types of words or phrases that do not bear sentiments on their own, but when they appear in some particular contexts, they imply positive or negative opinions. All these expressions have to be extracted and associated problems solved before sentiment analysis can achieve the next level of accuracy.

1. Positive ← consume no or little resource
2. | consume less resource
3. Negative ← consume a large quantity of resource
4. | consume more resource

Fig. 6 Sentiment polarity of statements involving resources

One such type of expressions involves resources, which occur frequently in many application domains. For example, *money* is a resource in probably every domain ("*this phone costs a lot of money*"), *gas* is a resource in the car domain, and *ink* is a resource in the printer domain. If a device consumes a large quantity of resource, it is undesirable (negative). If a device consumes little resource, it is desirable (positive). For example, the sentences, "*This laptop needs a lot of battery power*" and "*This car eats a lot of gas*" imply negative sentiments on the laptop and the car. Here, "*gas*" and "*battery power*" are resources, and we call

these words *resource terms* (which cover both *words* and *phrases*). They are a kind of special product aspects.

In terms of sentiments involving resources, the rules in Figure 6 are applicable (Liu, 2010). Rules 1 and 3 represent normal sentences that involve resources and imply sentiments, while rules 2 and 4 represent comparative sentences that involve resources and also imply sentiments, e.g., "*this washer uses much less water than my old GE washer*".

Zhang and Liu (2011a) formulated the problem based on a bipartite graph and proposed an iterative algorithm to solve the problem. The algorithm was based on the following observation:

Observation: The sentiment or opinion expressed in a sentence about resource usage is often determined by the flowing triple,

(verb, quantifier, noun_term),

where *noun_term* is a noun or a noun phrase representing a resource.

The proposed method used such triples to help identify resources in a domain corpus. The model used a circular definition to reflect a special reinforcement relationship between *resource usage verbs* (e.g., *consume*) and resource terms (e.g., *water*) based on the bipartite graph. The quantifier was not used in computation but was employed to identify candidate verbs and resource terms. The algorithm assumes that a list of quantifiers is given, which is not numerous and can be manually compiled. Based on the circular definition, the problem is solved using an iterative algorithm similar to the HITS algorithm in (Kleinberg, 1999). To start the iterative computation, some global *seed resources* are employed to find and to score some strong resource usage verbs. These scores are then applied as the initialization for the iterative computation for any application domain. When the algorithm converges, a ranked list of candidate resource terms is identified.

4 Entity Extraction

The task of entity extraction belongs to the traditional named entity recognition (NER) problem, which has been studied extensively. Many supervised information extraction approaches (e.g., HMM and CRF) can be adopted directly (Putthividhya and Hu, 2011). However, opinion mining also presents some special problems. One of them is the following: in a typical opinion mining application, the user wants to find opinions about some competing entities, e.g., competing products or brands (e.g., Canon, Sony, Samsung and many more). However, the user often can only provide a few names because there are so many different brands and models. Web users also write the names of the same product in various ways in forums and blogs. It is thus important for a system to automatically discover them from relevant corpora. The key requirement of this discovery is that the discovered entities must be relevant, i.e., they must be of the same class/type as the user provided entities, e.g., same brands or models.

Essentially, this is a *PU learning problem* (Positive and Unlabeled Learning), which is also called *learning from positive and unlabeled examples* (Liu et al., 2002). Formally, the problem is stated as follows: given a set of examples P of a particular class, called the *positive class*, and a set of unlabeled examples U, we wish to determine which of the unlabeled examples in U belong to the positive class represented by P. This gives us a two-class classification problem. Many algorithms are available in the literature for solving this problem (see the references in (Liu, 2006-2011).

A specialization of the PU learning problem for named entity extraction is called the *set expansion problem* (Ghahramani and Heller, 2005). The problem is stated similarly: Given a set Q of seed entities of a particular class C, and a set D of candidate entities, we wish to determine which of the entities in D belong to C. That is, we "*grow*" the class C based on the set of seed examples Q. As a specialization of PU learning, this is also a two-class classification problem which needs a binary decision for each entity in D (belonging to C or not belonging to C). However, in practice, the problem may be solved as a ranking problem, i.e., to rank the entities in D based on their likelihoods of belonging to C. In our scenario, the user-given entities are the set of initial seeds. The opinion mining system needs to expand the set using a text corpus.

4.1 Extraction Methods

The classic methods for solving set expansion problem are based on distributional similarity (Lee, 1999; Pantel et al., 2009). This approach works by comparing the similarity of the word distribution of the surrounding words of a candidate entity and the seed entities, and then ranking the candidate entities based on their similarity values. However, Li et al. (2010b) pointed out that this approach is inaccurate. In this section, we will discuss two machine learning approaches: *Positive and Unlabeled Learning* (PU Learning) and *Bayesian Sets*, which show better results than traditional methods.

4.1.1 PU Learning

In machine learning, there is a class of semi-supervised learning algorithms that learns from *positive* and *unlabeled* examples (PU learning). Its key characteristic (Liu et al., 2002) is that there is no negative training example available for learning. As stated above, PU learning is a two-class classification model. Its objective is to build a classifier using P and U to classifying the data in U or future test cases. The results can be either binary decisions (whether each test case belongs to the positive class or not), or a ranking based on how likely each test case belongs to the positive class represented by P. Clearly, the set expansion problem is a special case of PU learning, where the set Q is P here and the set D is U here.

There are several PU learning algorithms (Liu et al., 2002; Li and Liu, 2003; Li et al., 2007; Yu et al., 2002). Li et al. (2010b) used the S-EM algorithm proposed

in (Liu et al., 2002) for entity extraction in opinion documents. The main idea of S-EM is to use a *spy* technique to identify some *reliable negatives* (*RN*) from the unlabeled set *U*, and then use an EM algorithm to learn from *P*, *RN* and *U–RN*. To apply S-EM algorithm, Li et al. (2010b) takes following basic steps.

Generating Candidate Entities: It selects single words or phrases as candidate entities based on their part-of-speech (POS) tags. In particular, it chooses the following POS tags as entity indicators — NNP (proper noun), NNPS (plural proper noun), and CD (cardinal number).

Generating Positive and Unlabeled Sets: For each seed, each occurrence in the corpus forms a vector as a positive example in *P*. The vector is formed based on the surrounding word context of the seed mention. Similarly, for each candidate $d \in D$ (*D* denotes the set of all candidates), each occurrence also forms a vector as an unlabeled example in *U*. Thus, each unique seed or candidate entity may produce multiple feature vectors, depending on the number of times that the seed appears in the corpus. The components in the feature vectors are term frequencies.

Ranking Entity Candidates: With positive and unlabeled data, S-EM applied. At convergence, S-EM produces a Bayesian classifier *C*, which is used to classify each vector $u \in U$ and to assign a probability $p(+|u)$ to indicate the likelihood that *u* belongs to the positive class. Note that each unique candidate entity may generate multiple feature vectors, depending on the number of times that the candidate entity occurs in the corpus. As such, the rankings produced by S-EM are not the rankings of the entities, but rather the rankings of the entities' occurrences. Since different vectors representing the same candidate entity can have very different probabilities, Li et al. (2010b) compute a single score for each unique candidate entity for ranking based on Equation (11).

Let the probabilities (or scores) of a candidate entity $d \in D$ be $V_d = \{v_1, v_2 \ldots, v_n\}$ obtained from the feature vectors representing the entity. Let M_d be the median of V_d. The final score *f* for *d* is defined as following:

$$f(d) = M_d \times \log(1+n) \tag{11}$$

The use of the median of V_d can be justified based on the statistical *skewness* (Neter et al, 1993). Note that here *n* is the frequency count of candidate entity *d* in the corpus. The constant 1 is added to smooth the value. The idea is to push the frequent candidate entities up by multiplying the logarithm of frequency. *log* is taken in order to reduce the effect of big frequency counts.

The final score $f(d)$ indicates candidate *d*'s overall likelihood to be a relevant entity. A high $f(d)$ implies a high likelihood that *d* is in the expanded entity set. The top-ranked candidates are most likely to be relevant entities to the user-provided seeds.

4.1.2 Bayesian Sets

Bayesian Sets is also a semi-supervised learning method, more specifically, a PU learning method, which is based on Bayesian inference and only performs

ranking. Let D be a collection of items and Q be a user-given seed set of items, which is a (small) subset of D (i.e., $Q \subseteq D$). The task of Bayesian Sets is to use a model-based probabilistic criterion to give a score to each item e in D ($e \in D$) to gauge how well e fits into Q. In other words, it measures how likely e belongs to the *hidden* class represented/implied by Q. Each item e is represented with a binary feature vector.

The Bayesian criterion score for item e is expressed as follows:

$$score(e) = \frac{p(e \mid Q)}{p(e)} \tag{12}$$

$p(e \mid Q)$ represents how probable that e belongs to the same class as Q given the examples in Q. $p(e)$ is the prior probability of item e. Using Bayes rule, the equation can be re-written as:

$$score(e) = \frac{p(e,Q)}{p(e)p(Q)} \tag{13}$$

Equation (13) can be interpreted as the ratio of the joint probability of observing e and Q, to the probability of independently observing e and Q. The ratio basically compares the probability that e and Q are generated by the same model with parameters θ, and the probability that e and Q are generated by different models with different parameters θ and $\widetilde{\theta}$. Equation (13) says that if the probability that e and Q are generated from the same model with the parameters θ is high, the score of e will be high. On the other hand, if the probability that e and Q come from different models with different parameters θ and $\tilde{\theta}$ is high, the score will be low.

In pseudo code, the Bayesian Sets algorithm is given in Figure 7.

Algorithm: BayesianSets(Q, D)
Input: A small seed set Q of entities
A set of candidate entities D (= $\{e_1, e_2, e_3 \ldots e_n\}$)
Output: A ranked list of entities in D

1. **for** each entity e_i in D
2. compute: $score(e_i) = \frac{p(e_i,Q)}{P(e_i)p(Q)}$
3. **end for**
4. Rank the items in D based on their scores;

Fig. 7 The Bayesian Sets learning algorithm

If we assume that $q_k \in Q$ is independently and identically distributed (i.i.d.) and Q and e_i come from the same model with the same parameters θ, each of the three terms in Equation (13) are marginal likelihoods and can be written as integrals of the following forms:

$$p(Q) = \int [\prod_{q_k \in Q} p(q_k \mid \theta)] p(\theta) d\theta \tag{14}$$

$$p(e_i) = \int p(e_i \mid \theta) p(\theta) d\theta \tag{15}$$

$$p(e_i, Q) = \int [\prod_{q_k \in Q} p(q_k \mid \theta)] p(e_i \mid \theta) p(\theta) d\theta \tag{16}$$

Let us first compute the integrals of Equation (14). Each seed entity $q_k \in Q$ is represented as a binary feature vector (q_{k1}, q_{k2} , … q_{kj}). We assume each element of the feature vector has an independent Bernoulli distribution:

$$p(q_k \mid \theta) = \prod_{j=1}^{J} \theta_j^{q_{kj}} (1 - \theta_j)^{1 - q_{kj}} \tag{17}$$

The conjugate prior for the parameters of a Bernoulli distribution is the Beta distribution:

$$p(\theta \mid \alpha, \beta) = \prod_{j=1}^{J} \frac{\Gamma(\alpha_j + \beta_j)}{\Gamma(a_j)\Gamma(\beta_j)} \theta_j^{\alpha_j - 1} (1 - \theta_j)^{\beta_j - 1} \tag{18}$$

Where α and β are hyperparameters (which are also vectors). We set α and β empirically from the data, $\alpha_j = km_j$, $\beta_j = k(1- m_j)$, where m_j is the mean value of j-th components of all possible entities, and k is a scaling factor. The Gamma function is a generalization of the factorial function. For $Q = \{q_1, q_2, \ldots, q_n\}$, Equation (14) can be represented as follows:

$$p(Q \mid \alpha, \beta) = \prod_{j} \frac{\Gamma(a_j + \beta_j)}{\Gamma(\alpha_j)\Gamma(\beta_j)} \frac{\Gamma(\tilde{\alpha}_j)\Gamma(\tilde{\beta}_j)}{\Gamma(\tilde{\alpha}_j + \tilde{\beta}_j)} \tag{19}$$

where $\tilde{\alpha}_j = \alpha_j + \sum_{k=1}^{N} q_{kj}$ and $\tilde{\beta}_j = \beta_j + N - \sum_{k=1}^{N} q_{kj}$ With the same idea, we can compute Equation (15) and Equation (16).

Overall, the score of e_i, which is also represented a feature vector, (e_{i1}, e_{i2} , … e_{ij}) in the data, is computed with:

$$score(e_i) = \prod_{j} \frac{\alpha_j + \beta_j}{a_j + \beta_j + N} (\frac{\tilde{\alpha}_j}{\alpha_j}) e_{ij} (\frac{\tilde{\beta}_j}{\beta_j})^{1 - e_{ij}} \tag{20}$$

The log of the score is linear in e_i:

$$\log score(e_i) = c + \sum_j w_j e_{ij} \tag{21}$$

where

$$c = \sum_j \log(\alpha_j + \beta_j) - \log(\alpha_j + \beta_j + N) + \log \tilde{\beta}_j - \log \beta_j$$

$$\text{and} \quad w_j = \log \tilde{\alpha}_j - \log \alpha_j - \log \tilde{\beta}_j + \log \beta_j \tag{22}$$

All possible entities e_i will be assigned a similarity score by Equation (21). Then we can rank them accordingly. The top ranked entities should be highly related to the seed set Q according to the Bayesian Sets algorithm.

However, Zhang and Liu (2011c) found that this direct application of Bayesian Sets produces poor results. They believe there are two main reasons. First, since Bayesian Sets uses binary features, multiple occurrences of an entity in the corpus, which give rich contextual information, is not fully exploited. Second, since the number of seeds is very small, the learned results from Bayesian Sets can be quite unreliable.

They proposed a method to improve Bayesian Sets, which produces much better results. The main improvements are as follows.

Raising Feature Weights: From Equation (21), we can see that the score of an entity e_i is determined only by its corresponding feature vector and the weight vector $w = (w_1, w_2, \ldots, w_j)$. Equation (22) shows a value of the weight vector w. They rewrite Equation (22) as follows,

$$w_j = \log \frac{\tilde{\alpha}_j}{\alpha_j} - \log \frac{\tilde{\beta}_j}{\beta_j} = \log(1 + \frac{\sum_{i=1}^{N} q_{ij}}{k m_j}) - \log(1 + \frac{N - \sum_{i=1}^{N} q_{ij}}{k(1 - m_j)}) \tag{23}$$

In Equation (23), N is the number of items in the seed set. As mentioned before, m_j is the mean of feature j of all possible entities and k is a scaling factor. m_j can be regarded as the prior information empirically set from the data.

In order to make a positive contribution to the final score of entity e, w_j must be greater than zero. Under this circumstance, it can obtain the following inequality based on Equation (23).

$$\sum_{i=1}^{N} q_{ij} > N m_j \tag{24}$$

Equation (24) shows that if feature j is effective ($w_j > 0$), the seed data mean must be greater than the candidate data mean on feature j. Only such kind of features can be regarded as high-quality features in Bayesian Sets. Unfortunately, it is not

always the case due to the idiosyncrasy of the data. There are many high-quality features, whose seed data mean may be even less than the candidate data mean. For example, in drug data set, "*prescribe*" can be a left first verb for an entity. It is a very good entity feature. "*Prescribe EN/NNP*" (EN represents an entity, NNP is its POS tag) strongly suggests that EN is a drug. However, the problem is that the mean of this feature in the seed set is 0.024 which is less than its candidate set mean 0.025. So if we stick with Equation (24), the feature will have negative contribution, which means that it is worse than no feature at all. The fact that all pattern features are from sentences containing seeds, a candidate entity associated with a feature should be better than no feature.

Zhang and Liu tackled this problem by fully utilizing all features found in corpus. They changed original m_j to $\tilde{m}_j$ by multiplying a scaling factor t to force all feature weights $w_j > 0$:

$$\tilde{m}_j = tm_j \qquad (0 < t < 1) \tag{25}$$

The idea is that they lower the candidate data mean intentionally so that all the features found from the seed data can be utilized.

Identifying High-Quality Features: Equation (23) shows that besides m_j value, w_j value is also affected by the sum $\sum_{i=1}^{N} q_{ij}$. It means that if the feature occurs more times in the seed data, its corresponding w_j will also be high. However, Equation (23) may not be sufficient since it only considers the feature occurrence but does not take feature quality into consideration. For example, two different features A and B, which have the same feature occurrence in the seed data and thus the same mean, According to Equation (23), they should have the same feature weight w. However, for feature A, all feature counts may come from only one entity in the seed set, but for feature B, the feature counts are from four different entities in the seed set. Obviously, feature B is a better feature than feature A simply because the feature is shared by or associated with more entities. To detect such high-quality features to increase their weights, Zhang and Liu used the following formula to change the original w_j to $\tilde{w}_j$.

$$\tilde{w}_j = rw_j \tag{26}$$

$$r = (1 + \frac{\log h}{T}) \tag{27}$$

In Equation (26), r is used to represent feature quality for feature j. h is the number of unique entities that have j-th feature. T is the total number of entities in the seed set.

In Zhang and Liu (2011c), different vectors representing the same candidate entity are produced as in (Li et al., 2010b). Thus, the same ranking algorithm is adopted, which is the multiplication of the median of the score vector obtained from feature vectors representing the entity and the logarithm of entity frequency.

5 Summary

With the explosive growth of social media on the Web, organizations are increasingly relying on opinion mining methods to analyze the content of these media for their decision making. Aspect-based opinion mining, which aims to obtain detailed information about opinions, has attracted a great of deal of attention from both the research community and industry. Aspect extraction and entity extraction are two of its core tasks. In this chapter, we reviewed some representative works for aspect extraction and entity extraction from opinion documents.

For aspect extraction, existing solutions can be grouped into three main categories:

(1) using language dependency rules, e.g., *double propagation* (Qiu et al., 2011). These methods utilize the relationships between aspects and opinion words or other terms to perform aspect extraction. The approaches are unsupervised and domain-independent. Thus, they can be applied to any domain.
(2) using sequence learning algorithms such as HMM and CRF (Jin et al., 2009a; Jakob and Gurevych, 2010). These supervised methods are the dominating techniques for traditional information extraction. But they need a great deal of manual labeling effort.
(3) using topic models, e.g., MG-LDA (Titov and McDonald, 2008a). This is a popular research area for aspect extraction recently. The advantages of topic models are that they can group similar aspect expressions together and that they are unsupervised. However, their limitation is that the extracted aspects are not fine-grained.

For entity extraction, supervised learning has also been the dominating approach. However, semi-supervised methods have drawn attention recently. As in opinion mining, users often want to find competing entities for opinion analysis, they can provide some knowledge (e.g., entity instances) as seeds for semi-supervised learning. In this chapter, we introduced *PU learning* and *Bayesian Sets* based semi-supervised extraction methods.

For evaluation, the commonly used measures for information extraction such as precision, recall and F-1 scores are also often used in aspect and entity extraction. The current F-1 score results range from 0.60 to 0.85 depending on domains and datasets. Thus, the problems, especially aspect extraction, remain to be highly challenging. We expect that the future work will improve the accuracy significantly. We also believe that semi-supervised and unsupervised methods will play a larger role in these tasks.

References

Bethard, S., Yu, H., Thornton, A., Hatzivassiloglou, V., Jurafsky, D.: Automatic extraction of opinion propositions and their holders. In: Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text (2004)

Blair-Goldensohn, S., Hannan, K., McDonald, R., Neylon, T., Reis, G.A., Reyna, J.: Building a sentiment summarizer for local service reviews. In: Proceedings of International Conference on World Wide Web Workshop of NLPIX, WWW-NLPIX-2008 (2008)

Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. The Journal of Machine Learning Research (2003)

Bloom, K., Grag, N., Argamon, S.: Extracting apprasial expressions. In: Proceedings of the 2007 Annual Conference of the North American Chapter of the ACL (NAACL 2007) (2007)

Branavan, S.R.K., Chen, H., Eisenstein, J., Barzilay, R.: Learning document-level semantic properties from free-text annotations. In: Proceedings of Annual Meeting of the Association for Computational Linguistics, ACL 2008 (2008)

Brown, F.P., Della Pietra, S.A., Della Pietra, V.J., Mercer, R.L.: The mathematics of statitical machine translation: parameter estimation. Computational Linguistics (1993)

Brody, S., Elhadad, S.: An unsupervised aspect-sentiment model for online reviews. In: Proceedings of the 2010 Annual Conference of the North American Chapter of the ACL, NAACL 2010 (2010)

Carenini, G., Ng, R., Pauls, A.: Multi-Document summarization of evaluative text. In: Proceeding of Conference of the European Chapter of the ACL, EACL 2006 (2006)

Carenini, G., Ng, R., Zwart, E.: Extracting knowledge from evaluative text. In: Proceedings of Third International Conference on Knowledge Capture, K-CAP 2005 (2005)

Choi, Y., Cardie, C., Riloff, E., Patwardhan, S.: Identifying sources of opinions with conditional random fields and extraction patterns. In: Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing, HLT/EMNLP 2005 (2005)

Dempster, P., Laird, A.M.N., Rubin, B.D.: Maximum likelihood from incomplete data via the EM algorithms. Journal of the Royal Statistical Society, Series B (1977)

Fei, G., Liu, B., Hsu, M., Castellanos, M., Ghosh, R.: A dictionary-based approach to identifying aspects implied by adjectives for opinion mining. In: Proceedings of International Conference on Computational Linguistics, COLING 2012 (2012)

Ghahramani, Z., Heller, K.A.: Bayesian sets. In: Proceeding of Annual Neural Information Processing Systems, NIPS 2005 (2005)

Ghani, R., Probst, K., Liu, Y., Krema, M., Fano, A.: Text mining for product attribute extraction. ACM SIGKDD Explorations Newsletter 8(1) (2006)

Gilks, R.W., Richardson, S., Spiegelhalter, D.: Markov Chain Monte Carlo in practice. Chapman and Hall (1996)

Grosz, J.B., Winstein, S., Joshi, A.K.: Centering: a framework for modeling the local coherence of discourse. Computational Linguistics 21(2) (1995)

Guo, H., Zhu, H., Guo, Z., Zhang, X., Su, Z.: Product feature categorization with multilevel latent semantic association. In: Proceedings of ACM International Conference on Information and Knowledge Management, CIKM 2009 (2009)

Hai, Z., Chang, K., Kim, J.: Implicit feature identification via co-occurrence association rule mining. Computational Linguistic and Intelligent Text Processing (2011)

Hai, Z., Chang, K., Cong, G.: One seed to find them all: mining opinion features via association. In: Proceedings of ACM International Conference on Information and Knowledge Management, CIKM 2012 (2012)

Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Machine Learning (2001)

Hu, M., Liu, B.: Mining opinion features in customer reviews. In: Proceedings of National Conference on Artificial Intelligence, AAAI 2004 (2004a)

Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2004 (2004b)

Jakob, N., Gurevych, I.: Extracting opinion targets in a single and cross-domain setting with conditional random fields. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, EMNLP 2010 (2010)

Jin, W., Ho, H.: A novel lexicalized HMM-based learning framework for web opinion mining. In: Proceedings of International Conference on Machine Learning, ICML 2009 (2009a)

Jin, W., Ho, H., Srihari, R.K.: OpinionMiner: a novel machine learning system for web opinion mining and extraction. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2009 (2009b)

Jindal, N., Liu, B.: Mining comparative sentences and relations. In: Proceedings of National Conference on Artificial Intelligence, AAAI 2006 (2006a)

Jindal, N., Liu, B.: Identifying comparative sentences in text documents. In: Proceedings of ACM SIGIR International Conference on Information Retrieval, SIGIR 2006 (2006b)

Jo, Y., Oh, A.: Aspect and sentiment unification model for online review analysis. In: Proceedings of the Conference on Web Search and Web Data Mining, WSDM 2011 (2011)

Kessler, J., Nicolov, N.: Targeting sentiment expressions through supervised ranking of linguistic configurations. In: Proceedings of the International AAAI Conference on Weblogs and Social Media, ICWSM 2009 (2009)

Kim, S.M., Hovy, E.: Extracting opinions, opinion holders, and topics expressed in online news media text. In: Proceedings of the ACL Workshop on Sentiment and Subjectivity in Text (2006)

Kleinberg, J.: Authoritative sources in hyper-linked environment. Journal of the ACM 46(5), 604–632 (1999)

Kobayashi, N., Inui, K., Matsumoto, Y.: Extracting aspect-evaluation and aspect-of relations in opinion mining. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP 2007 (2007)

Ku, L., Liang, Y., Chen, H.: Opinion extraction, summarization and tracking in news and blog corpora. In: Proceedings of AAAI-CAAW 2006 (2006)

Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of International Conference on Machine Learning, ICML 2001 (2001)

Lee, L.: Measures of distributional similarity. In: Proceedings of Annual Meeting of the Association for Computational Linguistics, ACL 1999 (1999)

Li, F., Han, C., Huang, M., Zhu, X., Xia, Y., Zhang, S., Yu, H.: Structure-aware review mining and summarization. In: Proceedings of International Conference on Computational Linguistics, COLING 2010 (2010a)

Li, F., Pan, S.J., Jin, Q., Yang, Q., Zhu, X.: Cross-Domain co-extraction of sentiment and topic lexicons. In: Proceedings of Annual Meeting of the Association for Computational Linguistics, ACL 2012 (2012a)

Li, S., Wang, R., Zhou, G.: Opinion target extraction using a shallow semantic parsing framework. In: Proceedings of National Conference on Artificial Intelligence, AAAI 2012 (2012b)

Li, X., Liu, B.: Learning to classify texts using positive and unlabeled data. In: Proceedings of International Joint Conferences on Artificial Intelligence, IJCAI 2003 (2003)

Li, X., Liu, B., Ng, S.: Learning to identify unexpected instances in the test set. In: Proceedings of International Joint Conferences on Artificial Intelligence, IJCAI 2007 (2007)

Li, X., Zhang, L., Liu, B., Ng, S.: Distributional similarity vs. PU learning for entity set expansion. In: Proceedings of Annual Meeting of the Association for Computational Linguistics, ACL 2010 (2010b)

Lin, C., He, Y.: Joint sentiment/topic model for sentiment analysis. In: Proceedings of ACM International Conference on Information and Knowledge Management, CIKM 2009 (2009)

Lin, D.: Dependency-based evaluation of MINIPAR. In: Proceedings of the Workshop on Evaluation of Parsing System, ICLRE 1998 (1998)

Liu, B.: Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, 1st edn. Springer (2006), 2nd edn. (2011)

Liu, B.: Sentiment analysis and subjectivity, 2nd edn. Handbook of Natural Language Processing (2010)

Liu, B.: Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers (2012)

Liu, B., Hu, M., Cheng, J.: Opinion observer: analyzing and comparing opinions on the web. In: Proceedings of International Conference on World Wide Web, WWW 2005 (2005)

Liu, B., Lee, W.-S., Yu, P.S., Li, X.: Partially supervised text classification. In: Proceedings of International Conference on Machine Learning, ICML 2002 (2002)

Liu, K., Xu, L., Zhao, J.: Opinion target extraction using word-based translation model. In: Proceeding of Conference on Empirical Methods in Natural Language Processing, EMNLP 2012 (2012)

Long, C., Zhang, J., Zhu, X.: A review selection approach for accurate feature rating estimation. In: Proceedings of International Conference on Computational Linguistics, COLING 2010 (2010)

Lu, Y., Duan, H., Wang, H., Zhai, C.: Exploiting structured ontology to organize scattered online opinions. In: Proceedings of International Conference on Computational Linguistics, COLING 2010 (2010)

Lu, Y., Zhai, C., Sundaresan, N.: Rated aspect summarization of short comments. In: Proceedings of International Conference on World Wide Web, WWW 2009 (2009)

Ma, T., Wan, X.: Opinion target extraction in Chinese news comments. In: Proceedings of International Conference on Computational Linguistics (COLING 2010) (2010)

Mauge, K., Rohanimanesh, K., Ruvini, J.D.: Structuring e-commerce inventory. In: Proceedings of Annual Meeting of the Association for Computational Linguistics, ACL 2012 (2012)

Mukherjee, A., Liu, B.: Aspect extraction through semi-supervised modeling. In: Proceedings of Annual Meeting of the Association for Computational Linguistics, ACL 2012 (2012)

Mei, Q., Ling, X., Wondra, M., Su, H., Zhai, C.: Topic sentiment mixture: modeling facets and opinions in weblogs. In: Proceedings of International Conference on World Wide Web, WWW 2007 (2007)

Moghaddam, S., Ester, M.: Opinion digger: an unsupervised opinion miner from unstructured product reviews. In: Proceedings of ACM International Conference on Information and Knowledge Management, CIKM 2010 (2010)

Moghaddam, S., Ester, M.: ILDA: interdependent LDA model for learning latent aspects and their ratings from online product reviews. In: Proceedings of ACM SIGIR International Conference on Information Retrieval, SIGIR 2011 (2011)

Neter, J., Wasserman, W., Whitmore, G.A.: Applied Statistics. Allyn and Bacon (1993)

Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (2008)

Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, EMNLP 2002 (2002)

Pantel, P., Crestan, E., Borkovsky, A., Popescu, A.: Web-Scale distributional similarity and entity set expansion. In: Proceedings of the 2009 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP 2009 (2009)

Popescu, A., Etzioni, O.: Extracting product features and opinions from reviews. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, EMNLP 2005 (2005)

Putthividhya, D., Hu, J.: Bootstrapped name entity recognition for product attribute extraction. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, EMNLP 2011 (2011)

Qiu, G., Liu, B., Bu, J., Chen, C.: Opinion word expansion and target extraction through double propagation. Computational Linguistics (2011)

Rabiner, R.L.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of IEEE 77(2) (1989)

Sarawagi, S.: Information Extraction. Foundations and Trends in Databases (2008)

Sauper, C., Haghighi, A., Barzilay, R.: Content models with attribute. In: Proceedings of Annual Meeting of the Association for Computational Linguistics, ACL 2011 (2011)

Scaffidi, C., Bierhoff, K., Chang, E., Felker, M., Ng, H., Jin, C.: Red opal: product-feature scoring from reviews. In: Proceedings of the 9th International Conference on Electronic Commerce, EC 2007 (2007)

Stoyanov, V., Cardie, C.: Topic identification for fine-grained opinion analysis. In: Proceedings of International Conference on Computational Linguistics, COLING 2008 (2008)

Su, Q., Xu, X., Guo, H., Guo, Z., Wu, X., Zhang, X., Swen, B., Su, Z.: Hidden sentiment association in Chinese web opinion mining. In: Proceedings of International Conference on World Wide Web, WWW 2008 (2008)

Sutton, C., McCallum, A.: An introduction to conditional random fields for relational learning. Introduction to Statistical Relational Learning. MIT Press (2006)

Titov, I., McDonald, R.: Modeling online reviews with multi-grain topic models. In: Proceedings of International Conference on World Wide Web, WWW 2008 (2008a)

Titov, I., McDonald, R.: A joint model of text and aspect ratings for sentiment summarization. In: Proceedings of Annual Meeting of the Association for Computational Linguistics, ACL 2008 (2008b)

Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: Proceedings of Annual Meeting of the Association for Computational Linguistics, ACL 2002 (2002)

Wang, B., Wang, H.: Bootstrapping both product features and opinion words from Chinese customer reviews with cross-inducing. In: Proceedings of the International Joint Conference on Natural Language Processing, IJCNLP 2008 (2008)

Wang, H., Lu, Y., Zhai, C.: Latent aspect rating analysis on review text data: a rating regression approach. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2010 (2010)

Wei, W., Gulla, J.A.: Sentiment learning on product reviews via sentiment ontology tree. In: Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL 2010) (2010)

Wiebe, J., Riloff, E.: Creating subjective and objective sentence classifiers from unannotated texts. In: Proceedings of Computational Linguistics and Intelligent Text Processing, CICLing 2005 (2005)

Wiebe, J., Wilson, T., Bruce, R., Bell, M., Martin, M.: Learning subjective language. Computational Linguistics 30(3), 277–308 (2004)

Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing, HLT/EMNLP 2005 (2005)

Wu, Y., Zhang, Q., Huang, X., Wu, L.: Phrase dependency parsing for opinion mining. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, EMNLP 2009 (2009)

Yi, J., Nasukawa, T., Bunescu, R., Niblack, W.: Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques. In: Proceedings of International Conference on Data Mining, ICDM 2003 (2003)

Yu, H., Han, J., Chang, K.: PEBL: Positive example based learning for Web page classification using SVM. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2002 (2002)

Yu, J., Zha, Z., Wang, M., Chua, T.: Aspect ranking: identifying important product aspects from online consumer reviews. In: Proceedings of Annual Meeting of the Association for Computational Linguistics, ACL 2011 (2011a)

Yu, J., Zha, Z., Wang, M., Wang, K., Chua, T.: Domain-Assisted product aspect hierarchy generation: towards hierarchical organization of unstructured consumer reviews. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, EMNLP 2011 (2011b)

Zhai, Z., Liu, B., Xu, H., Jia, P.: Clustering product features for opinion mining. In: Proceedings of ACM International Conference on Web Search and Data Mining, WSDM 2011 (2011)

Zhai, Z., Liu, B., Xu, H., Jia, P.: Grouping product features using semi-supervised learning with soft-constraints. In: Proceedings of International Conference on Computational Linguistics, COLING 2010 (2010)

Zhang, L., Liu, B., Lim, S., O'Brien-Strain, E.: Extracting and ranking product features in opinion documents. In: Proceedings of International Conference on Computational Linguistics, COLING 2010 (2010)

Zhang, L., Liu, B.: Identifying noun product features that imply opinions. In: Proceedings of Annual Meeting of the Association for Computational Linguistics, ACL 2011 (2011a)

Zhang, L., Liu, B.: Extracting resource terms for sentiment analysis. In: Proceedings of the International Joint Conference on Natural Language Processing, IJCNLP 2011 (2011b)

Zhang, L., Liu, B.: Entity set expansion in opinion documents. In Proceedings of ACM Conference on Hypertext and Hypermedia (HT 2011) (2011c)

Zhao, W., Jiang, J., Yan, H., Li, X.: Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, EMNLP 2010 (2010)

Zhu, J., Wang, H., Tsou, B.K., Zhu, M.: Multi-aspect opinion polling from textual reviews. In: Proceedings of ACM International Conference on Information and Knowledge Management, CIKM 2009 (2009)

Zhuang, L., Jing, F., Zhu, X.: Movie review mining and summarization. In: Proceedings of ACM International Conference on Information and Knowledge Management, CIKM 2006 (2006)

Mining Periodicity from Dynamic and Incomplete Spatiotemporal Data

Zhenhui Li and Jiawei Han

Abstract. As spatiotemporal data becomes widely available, mining and understanding such data have gained a lot of attention recently. Among all important patterns, periodicity is arguably the most frequently happening one for moving objects. Finding periodic behaviors is essential to understanding the activities of objects, and to predict future movements and detect anomalies in trajectories. However, periodic behaviors in spatiotemporal data could be complicated, involving multiple interleaving periods, partial time span, and spatiotemporal noises and outliers. Even worse, due to the limitations of positioning technology or its various kinds of deployments, real movement data is often highly incomplete and sparse. In this chapter, we discuss existing techniques to mine periodic behaviors from spatiotemporal data, with a focus on tackling the aforementioned difficulties risen in real applications. In particular, we first review the traditional time-series method for periodicity detection. Then, a novel method specifically designed to mine periodic behaviors in spatiotemporal data, Periodica, is introduced. Periodica proposes to use reference spots to observe movement and detect periodicity from the in-and-out binary sequence. Then, we discuss the important issue of dealing with sparse and incomplete observations in spatiotemporal data, and propose a new general framework Periodo to detect periodicity for temporal events despite such nuisances. We provide experiment results on real movement data to verify the effectiveness of the proposed methods. While these techniques are developed in the context of spatiotemporal data mining, we believe that they are very general and could benefit researchers and practitioners from other related fields.

Zhenhui Li
Pennsylvania State University, University Park, PA
e-mail: jessieli@ist.psu.edu

Jiawei Han
University of Illinois at Urbana-Champaign, Champaign, IL
e-mail: hanj@cs.uiuc.edu

W.W. Chu (ed.), *Data Mining and Knowledge Discovery for Big Data,*
Studies in Big Data 1,
DOI: 10.1007/978-3-642-40837-3_2, © Springer-Verlag Berlin Heidelberg 2014

1 Introduction

With the rapid development of positioning technologies, sensor networks, and online social media, spatiotemporal data is now widely collected from smartphones carried by people, sensor tags attached to animals, GPS tracking systems on cars and airplanes, RFID tags on merchandise, and location-based services offered by social media. While such tracking systems act as real-time monitoring platforms, analyzing spatiotemporal data generated from these systems frames many research problems and high-impact applications. For example, understanding and modeling animal movement is important to addressing environmental challenges such as climate and land use change, bio-diversity loss, invasive species, and infectious diseases.

As spatiotemporal data becomes widely available, there are emergent needs in many applications to understand the increasingly large collections of data. Among all the patterns, one most common pattern is the *periodic behavior*. A periodic behavior can be loosely defined as the repeating activities at certain locations with regular time intervals. For example, bald eagles start migrating to South America in late October and go back to Alaska around mid-March. People may have weekly periodicity staying in the office.

Mining periodic behaviors can benefit us in many aspects. First, periodic behaviors provide an insightful and concise explanation over the long moving history. For example, animal movements can be summarized using mixture of multiple *daily* and *yearly* periodic behaviors. Second, periodic behaviors are also useful for compressing spatiotemporal data [17, 25, 4]. Spatiotemporal data usually have huge volume because data keeps growing as time passes. However, once we extract periodic patterns, it will save a lot of storage space by recording the periodic behaviors rather than original data, without losing much information. Finally, periodicity is extremely useful in future movement prediction [10], especially for a distant querying time. At the same time, if an object fails to follow regular periodic behaviors, it could be a signal of abnormal environment change or an accident.

More importantly, since spatiotemporal data is just a special class of temporal data, namely two-dimensional temporal data, many ideas and techniques we discuss in this chapter can actually be applied to other types of temporal data collected in a broad range of fields such as bioinformatics, social network, environmental science, and so on. For example, the notion of probabilistic periodic behavior can be very useful in understanding the social behaviors of people via analyzing the social network data such as tweets. Also, the techniques we developed for period detection from noisy and incomplete observations can be applied to any kind of temporal event data, regardless of the type of the collecting sensor.

1.1 Challenges in Mining Periodicity from Spatiotemporal Data

Mining periodic behaviors can bridge the gap between raw data and semantic understanding of the data, but it is a challenging problem. For example, Figure 1 shows the raw movement data of a student David along with the expected periodic behaviors.

Based on manual examination of the raw data (on the left), it is almost impossible to extract the periodic behaviors (on the right). In fact, the periodic behaviors are quite complicated. There are multiple periods and periodic behaviors that may interleave with each other. Below we summarize the major challenges in mining periodic behavior from movement data:

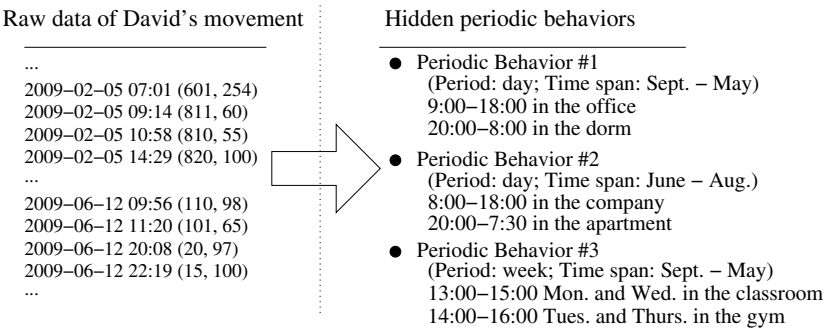


Fig. 1 Interleaving of multiple periodic behaviors

1. *A real life moving object does not ever strictly follow a given periodic pattern*. For example, birds never follow exactly the same migration paths every year. Their migration routes are strongly affected by weather conditions and thus could be substantially different from previous years. Meanwhile, even though birds generally stay in north in the summer, it is not the case that they stay at exactly the same locations, on exactly the same days of the year, as previous years. Therefore, "north" is a fairly vague geo-concept that is hard to be modeled from raw trajectory data. Moreover, birds could have multiple interleaved periodic behaviors at different spatiotemporal granularities, as a result of daily periodic hunting behaviors, combined with yearly migration behaviors.
2. We usually have *incomplete observations*, which *are unevenly sampled* and *have large portion of missing data*. For example, a bird can only carry small sensors with one or two reported locations in three to five days. And the locations of a person may only be recorded when he uses his cellphone. Moreover, if a sensor is not functioning or a tracking facility is turned off, it could result in a large portion of missing data.
3. With the periods detected, *the corresponding periodic behaviors should be mined* to provide a semantic understanding of movement data, such as the hidden periodic behaviors shown in Figure 1. The challenge in this step lies in the interleaving nature of multiple periodic behaviors. As we can see that, for a person's movement as shown in Figure 1, one periodic behavior can be associated with different locations, such as periodic behavior #1 is associated with both office and dorm. Also, the same period (*i.e.*, day) could be associated with two different periodic behaviors, one from September to May and the other from June to August.

1.2 Existing Periodicity Mining Techniques

In this section, we will describe the existing periodicity mining techniques on various types of data, such as signal processing, gene data, and symbolic sequences. The techniques for spatiotemporal mining will be discussed in more detail in Section 2. Here we focus on two problems: (1) period detection and (2) periodic behavior mining. Period detection is to *automatically* detect the periods in time series or sequences. Periodic behavior mining problem is to mine periodic patterns with *a given period*.

1.2.1 Period Detection in Signals

A signal is a function that conveys information about the behavior or attributes of some phenomenon. If the function is on the time domain, the signal is a temporal function (*i.e.*, time series). The most frequently used method to detect periods in signals are *Fourier transform* and *autocorrelation* [18].

Fourier Transform maps a function of time into a new function whose argument is frequency with units of cycles/sec (hertz). In the case of a periodic function, the Fourier transform can be simplified to the calculation of a discrete set of complex amplitudes, called Fourier series coefficients. Given a sequence $x(n)$, $n = 0, 1, \ldots, N-1$, the normalized Discrete Fourier Transform is a sequence of complex numbers $X(f)$:

$$X(f_{k/N}) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi kn}{N}}$$

where the subscript k/N denotes the frequency that each coefficient captures. In order to discover potential periodicities of a time series, one can use *periodogram* to estimate the spetral density of a signal. The periodogram P is provided by the squared length of each Fourier coefficient:

$$P(f_{k/N}) = \|X(f_{k/N})\|^2, k = 0, 1, \ldots, \lceil \frac{N-1}{2} \rceil$$

If $P(f_{k^*/N})$ is the maximum over all periodogram values of other frequencies, it means that frequency k^*/N has the strongest power in signal. Mapping frequency to time domain, a frequency k^*/N corresponds to time range $[\frac{N}{k^*}, \frac{N}{k^*-1})$.

Autocorrelation is the cross-correlation of a signal with itself. It is often used to find repeating patterns, such as the presence of a periodic signal. In statistics, autocorrelation of a time lag τ is defined as:

$$ACF(\tau) = \frac{1}{N} \sum_{n=0}^{N-1} x(\tau) \cdot x(n+\tau)$$

If $ACF(\tau^*)$ is the maximum over autocorrelation values of all time lags, it means that τ^* is most likely to be the period of the sequence. Different from Fourier transform that k^*/N is in frequency domain, time lag τ^* is in time domain.

Vlachos *et al.* [21] gives a comprehensive analysis and comparison between Fourier transform and autocorrelation. In general, Fourier transform is a great indicator for potential periods but the indicator is on the frequency domain. When mapping a frequency to time domain, it could correspond to a time range instead of one particular time. On the other hand, autocorrelation is not a good indicator for the true period because the true period and the multipliers of the true period will all have high autocorrelation values. For example, if τ^* is the true period, $ACF(k \cdot \tau^*)$ are all likely to have similar or even higher values than $ACF(\tau^*)$. Thus, it is hard to use a cut-off threshold to determine the true period. However, autocorrelation calculates the periodicity score on the time domain, so it does not have the mapping frequency problem in Fourier transform. In [21], Vlachos *et al.* proposes a method to combine autocorrelation and Fourier transform. It uses Fourier transform to find a good indicator of the potential period range and use autocorrelation to further validate the exact period.

1.2.2 Period Detection in Symbolic Sequences

Studies on period detection in data mining and database area usually assume the input to be a sequence of symbols instead of real value time series. A symbol could represent an event. An event could be a transaction record, for example, a person bought a bottle of milk. In transaction history, people could buy certain items periodically. Every timestamp is associated with one event or a set of events. The problem is to find whether there is an event or a set of events that have periodicity.

A common way to tackle the period detection in symbolic sequence is to get all the time indexes for each event and check whether these time indexes show periodicity. The time series that is being examined here can be considered as a binary sequence, $x = x_1 x_2 \ldots x_n$, where $x_t = 1$ means this event happens at time t and $x_t = 0$ means this event does not happen. The characteristics of such data is that the number of 1s could only be a very small portion in the sequence. And because of such sparsity, the period detection method is more sensitive to noise.

Ma *et al.* [16] proposes a chi-squared test for finding period by considering time differences in adjacent occurrences of an event. Let $s = \{t_1, t_2, \ldots, t_m\}$ denote all the timestamps that an event happens. It considers the time differences between every adjacent occurrences of the event: $\tau_i = t_{i+1} - t_i$. Looking at the histogram of all τ_i values, the true period p should have high frequency. In this method, authors use Chi-square measure to set the threshold for the frequency. If a time difference value p has frequency more than this threshold, it outputs p as the period.

Berberdis *et al.* [3] uses autocorrelation to detect periods in the binary sequence x. Elfeky *et al.* [5] further improves this method by considering *multiple* events at the same time. It assumes that there is only one event at each timestamp. Each event is mapped to a binary sequence. For example, event "a" maps to "001", event "b" maps to "010", event "c" maps to "100". Then the original symbolic sequence input is transformed into a binary sequence. It further applies autocorrelation on this binary sequence to detect periods. In a follow-up work [5], Elfeky *et al.* mention the previous methods [3, 5] are sensitive to noises. These noises include insertion,

deletion, replacement of an event at some timestamps. So [6] proposes a method based on Dynamic Time Warping to detect periods. The method is slower (*i.e.*, $O(n^2)$) compared with the previous method [5] (*i.e.*, $O(nlogn)$). But it is more accurate in terms of noises.

1.2.3 Period Detection in Gene Data

In bioinformatics, there are several studies in mining periods in gene data. A DNA sequence is a high-dimensional symbolic sequence. In [7], Glynn *et al.* mention that DNA sequence is often unevenly spaced and Fourier transform could fail when the data contains an excessive number of missing values. They propose to use Lomb-Scargle periodogram in such case. Lomb-Scargle periodogram [15, 19] is a variation of Fourier transform to handle unevenly spaced data using least-squares fitting of sinusoidal curves. In a follow-up work [1], Ahdesmäki *et al.* mention that Lomb-Scargle periodogram used in [7] is not robust since it is the basic Fisher's test. So they propose to use regression method for periodicity detection in non-uniformly sampled gene data. In [13], Liang *et al.* also mention that the performance of Lomb-Scargle periodogram [7] degrades in the presence of heavy-tailed non-Gaussian noise. In the presence of noises in gene data, Liang *et al.* [13] propose to use Laplace periodogram for more robust discovery of periodicity. They show Laplace periodogram is better than Lomb-Scarlge periodogram [7] and regression method [1]. An interesting previous study [11] has studied the problem of periodic pattern detection in sparse boolean sequences for gene data, where the ratio of the number of 1's to 0's is small. It proposes a scoring function for a potential period p by checking the alignment properties of periodic points in solenoidal coordinates w.r.t. p.

1.2.4 Periodic Behavior Mining

A number of *periodic pattern mining* techniques have been proposed in data mining literature. In this problem setting, each timestamp corresponds to a set of items. The goal is to, with a *given* period, find the period patterns that appear at least *min_sup* times. Han *et al.* [9, 8] propose algorithms for mining frequent partial periodic patterns. Yang *et al.* [27, 28, 23, 29] propose a series of work dealing with variations of periodic pattern mining, such as asynchronous patterns [27], surprising periodic patterns [28], patterns with gap penalties [29], and higher level patterns [23]. In [30], it further addresses the gap requirement problem in biologic sequences. Different from previous works which focus on the categorical data, Mamoulis *et al.* [17] detects the periodic patterns for moving objects. Frequent periodic pattern mining tend to output a large set of patterns, most of which are slightly different.

1.3 Organization of This Chapter

In Section 2, we first review in more details the existing work on applying time-series methods to detect periodicity in spatiotemporal data. Then, we introduce a

new approach, Periodica, which is able to discover complicated periodic behaviors from movement data. Section 3 is devoted to the important issue of detecting periodicity in real data: highly incomplete observations. We describe a novel method Periodo for robust periodicity detection for temporal events in these challenging cases, and verify its effectiveness by comparing it with existing methods on synthetic datasets. In Section 5, we show the results of applying the techniques introduced in this chapter to real spatiotemporal datasets, including the movement data of animals and humans. We conclude our discussion and point out future directions in Section 6.

2 Techniques for Periodicity Mining in Spatiotemporal Data

In this section, we describe techniques which are developed to detect periodic behaviors in spatiotemporal data. Let $D = \{(x_1, y_1, time_1), (x_2, y_2, time_2), \ldots\}$ be the original movement data for a moving object. Throughout this section, we assume that the raw data is linearly interpolated with constant time gap, such as hour or day. The interpolated sequence is denoted as $LOC = loc_1 loc_2 \cdots loc_n$, where loc_i is a spatial point represented as a pair $(loc_i.x, loc_i.y)$. Hence, our goal is to detect the periodicity in the movement sequence LOC.

While period detection in 1-D time series has been long studied, with standard techniques such as fast Fourier transform (FFT) and auto-correlation existing in the literature, solution to the problem of detecting periods in 2-D spatiotemporal data remains largely unknown until the recent work [2]. In this work, the authors first describe an intuitive approach to identify recursions in movement data, and then propose an extension of the 1-D Fourier Transform, named complex Fourier transform (CFT), to detect circular movements from the input sequence. Therefore, in this section we first review both methods, and point out their limitations in handling real-world movement data. Then, we show how such limitations can be overcome using a novel two-stage algorithm, Periodica, which is designed to mine complex periodic behaviors from real-world movement data.

2.1 Existing Time-Series Methods

There have been many period detection methods developed for time series analysis. A direct usage of time series techniques requires we transform the location sequence into time series. A simple transform is mapping a location (x, y) onto complex plane $x + iy$, where $i = \sqrt{-1}$. We denote the mapping of a location loc_k as a complex number z_k, where $z_k = loc_k.x + i loc_k.y$.

2.1.1 Recursion Analysis

Recursion analysis is used to identify *closed paths* in the movement patterns. In order to define a closed path, or a recursion, one needs to divide the landscape into a grid of patches (a 105×105 matrix is used in [2]). Then, a close path exists in the

movement sequence if an exact (to the resolution of landscape discretization) recursion to a previous location at a later time is found. To detect such recursions, one simply notices that the sum of vector displacements along a closed path is zero and thus requires the identification of zero-valued partial summations of the coordinates of sequential locations.

Specifically, given a sequence of locations vectors $z_k, k = 1,2,\ldots,n$, the method first compute the difference vectors $v_k = z_{k+1} - z_k$, for $k = 1,2,\ldots,n-1$. Then, for any time window $(s,t), t > s$, the segment of the path from z_s to z_t is denoted as $V(s,t)$:

$$V(s,t) = \sum_{k=s}^{t} v_k. \tag{1}$$

Thus, a recursion of duration D is a window for which $V(s,t) = 0$ and $t - s = D$. Notice that the recursion analysis identifies all closed paths, their length, and locations. These recursions are then sorted according to their durations to identify significant and semantic meaningful lengths of recursion (e.g., a day).

2.1.2 Circle Analysis

Fourier transform is one of the most widely used tools for time-series analysis. By extending it to complex numbers, one can identify circular paths, clockwise or counterclockwise, in the movement. Mathematically, given a sequence of location coordinates represented by a series of complex numbers $\{z_k\}_{k=1}^{n}$, the periodogram of the complex Fourier transform (CFT) of z_k is defined as:

$$Z(f) = \sum_{k=1}^{n} z_k \times e^{-i2\pi fk}, \quad f > 0 \tag{2}$$

Note that these spectra of Z are functions of the frequency f, which is the reciprocal of duration, D (*i.e.*, $D = 1/f$). It can be shown that $Z(f)$ provides an indication of the trend of circular motion, and can also be used to distinguish clockwise from counterclockwise patterns. Interested readers are referred to [2] for detailed illustrations and results of CFT.

Meanwhile, it is important to distinguish the circular analysis from the aforementioned recursion analysis. Note that a close path detected by recursion analysis is not necessarily circular, and similarly a clockwise or counterclockwise movement does not ensure a recursion. In this sense, these two methods are complementary to each other. Consequently, one can combine these two methods to answer more complex questions such as whether there is a circular path between recursions.

2.1.3 Limitations of Time-Series Methods

While tools from time-series analysis have demonstrated certain success when generalized to handle spatiotemporal data, it also has several major limitations as we elaborate below.

First, the performance of recursion analysis heavily rely on the resolution of landscape discretization, for which expert information about the moving objects' typical range of activity is crucial. For example, one will miss a lot of recursions when the resolution is set too coarse, whereas when the resolution is set too fine a large number of false positives will occur. Due to the same reason, the recursion analysis is also very sensitive to noise in the movement data.

Second, while circle analysis does not have the same dependency issue as recursion analysis, its usage is however strictly restricted to detecting circular paths in the movement data. Unfortunately, real-world spatiotemporal data often exhibit much more complex periodic patterns which are not necessarily circular (see Figure 2 for an example). Therefore, the development of a more flexible method is of great important in practice.

Finally, as we mentioned before, the objects of interest (e.g., humans, animals) often have multiple periodic behaviors with the same period, which is completely ignored by existing methods. In order to achieve semantic understanding of the data, it is important for our algorithm to be able to mine such multiple behaviors in movement data.

With all of these considerations in mind, we now proceed to describe a new algorithms for periodic behavior mining in spatiotemporal data, which handles all the aforementioned difficulties in a unified framework.

2.2 Periodica: Using Reference Spots to Detect Periodicity

As discussed above, periodic behaviors mined from spatiotemporal data can provide people with valuable semantic understanding of the movement. In order to mine periodic behaviors, one typically encounters the following two major issues.

First, the *periods* (*i.e.*, the regular time intervals in a periodic behavior) are usually unknown. Even though there are many period detection techniques that are proposed in signal processing area, such as Fourier transform and autocorrelation, we will see in Section 2.2.2 that these methods cannot be *directly* applied to the spatiotemporal data. Besides, there could be *multiple* periods existing at the same time, for example in Figure 1, David has one period as "day" and another as "week". If we consider the movement sequence as a whole, the longer period (*i.e.*, week) will have fewer repeating times than the shorter period (*i.e.*, day). So it is hard to select a threshold to find all periods. Surprisingly, there is no previous work that can handle the issue about how to detect multiple periods from the noisy moving object data.

Second, even if the periods are known, the *periodic behaviors* still need to be mined from the data because there could be *several* periodic behaviors with the same period. As we can see that, in David's movement, the same *period* (*i.e.*, day) is associated with two different *periodic behaviors*, one from September to May and the other from June to August. In previous work, Mamoulis *et al.* [17] studied the frequent periodic pattern mining problem for a moving object with a *given* period. However, the rigid definition of frequent periodic pattern does not encode the *statistical information*. It cannot describe the case such as "David has 0.8 probability

to be in the office at 9:00 everyday." One may argue that these frequent periodic patterns can be further summarized using probabilistic modeling approach [26, 22]. But such models built on frequent periodic patterns do not truly reflect the real underlying periodic behaviors from the original movement, because frequent patterns are already a lossy summarization over the original data. Furthermore, if we can directly mine periodic behaviors on the original movement using polynomial time complexity, it is unnecessary to mine frequent periodic patterns and then summarize over these patterns.

We formulate the periodic behavior mining problem and propose the assumption that the observed movement is generated from several *periodic behaviors* associated with some *reference locations*. We design a two-stage algorithm, Periodica, to detect the periods and further find the periodic behaviors.

At the first stage, we focus on detecting all the periods in the movement. Given the raw data as shown in Figure 1, we use the kernel method to discover those reference locations, namely *reference spots*. For each reference spot, the movement data is transformed from a spatial sequence to a binary sequence, which facilitates the detection of periods by filtering the spatial noise. Besides, based on our assumption, every period will be associated with at least one reference spot. *All* periods in the movement can be detected if we try to detect the periods in every reference spot. At the second stage, we statistically model the periodic behavior using a *generative model*. Based on this model, underlying periodic behaviors are generalized from the movement using a hierarchical clustering method and the number of periodic behaviors is automatically detected by measuring the *representation error*.

2.2.1 Problem Definition

Given a location sequence *LOC*, our problem aims at mining all periodic behaviors. Before defining periodic behavior, we first define some concepts. A *reference spot* is a dense area that is frequently visited in the movement. The set of all reference spots is denoted as $O = \{o_1, o_2, \ldots, o_d\}$, where d is the number of reference spots. A *period* T is a regular time interval in the (partial) movement. Let t_i $(1 \leq i \leq T)$ denote the i-th *relative timestamp* in T.

A *periodic behavior* can be represented as a pair $\langle T, \mathbf{P} \rangle$, where $\mathbf{P}$ is a probability distribution matrix. Each entry $\mathbf{P}_{ik} (1 \leq i \leq d, 1 \leq k \leq T)$ of $\mathbf{P}$ is the probability that the moving object is at the reference spot o_i at relative timestamp t_k.

As an example, for $T = 24$ (hours), David's daily periodic behavior (Figure 1 involved with 2 reference spots (*i.e.*, "office" and "dorm") could be represented as $(2+1) \times 24$ probability distribution matrix, as shown Table 1. This table is an intuitive explanation of formal output of periodic behaviors, which is not calculated according to specific data in Figure 1. The probability matrix encodes the noises and uncertainties in the movement. It statistically characterizes the periodic behavior such as "David arrives at office *around* 9:00."

Definition 1 (Periodic Behavior Mining). Given a length-n movement sequence *LOC*, our goal is to mine all the periodic behaviors $\{\langle T, \mathbf{P} \rangle\}$.

Table 1 A daily periodic behavior of David

	8:00	9:00	10:00	···	17:00	18:00	19:00
dorm	0.9	0.2	0.1	···	0.2	0.7	0.8
office	0.05	0.7	0.85	···	0.75	0.2	0.1
unknown	0.05	0.1	0.05	···	0.05	0.1	0.1

Since there are two subtasks in the periodic behavior mining problem, detecting the periods and mining the periodic behaviors. We propose a two-stage algorithm Periodica, where the overall procedure of the algorithm is developed in two stages and each stage targets one subtask.

Algorithm 1 shows the general framework of Periodica. At the first stage, we first find all the reference spots (Line 2) and for each reference spot, the periods are detected (Lines 3∼5). Then for every period T, we consider the reference spots with period T and further mine the corresponding periodic behaviors (Lines 7∼10).

Algorithm 1. Periodica

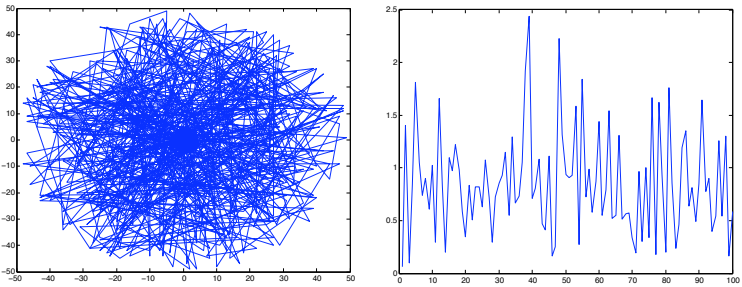
INPUT: A movement sequence $LOC = loc_1 loc_2 \cdots loc_n$.
OUTPUT: A set of periodic behaviors.
ALGORITHM:

```
1:  /* Stage 1: Detect periods */
2:  Find reference spots O = {o_1, o_2, ··· , o_d};
3:  for each o_i ∈ O do
4:      Detect periods in o_i and store the periods in P_i;
5:      P_set ← P_set ∪ P_i;
6:  end for
7:  /* Stage 2: Mine periodic behaviors */
8:  for each T ∈ P_set do
9:      O_T = {o_i | T ∈ P_i};
10:     Construct the symbolized sequence S using O_T;
11:     Mine periodic behaviors in S.
12: end for
```

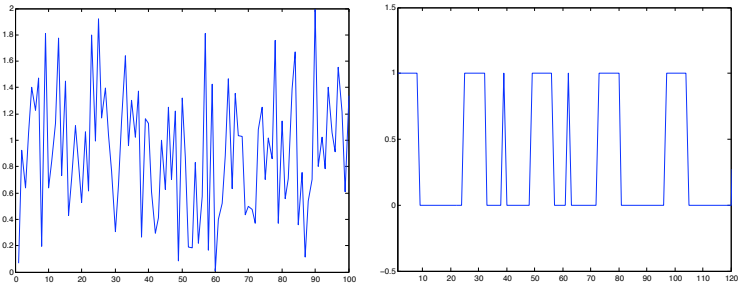
2.2.2 Detecting Period

In this section, we will discuss how to detect periods in the movement data. This includes two subproblems, namely, finding reference spots and detecting periods on binary sequence generated by these spots. First of all, we want to show why the idea of reference spots is essential for period detection. Consider the following example.

We generate a movement dataset simulating an animal's daily activities. Every day, this animal has 8 hours staying at the den and the rest time going to some random places hunting for food. Figure 2(a) shows its trajectories. We first try the method introduced in [2]. The method transforms locations (x, y) onto complex



(a) Raw trajectories (b) Fourier transform on $x + yi$



(c) Fourier transform on $y + xi$ (d) Binary sequence as viewed from the den

Fig. 2 Illustration of the importance to view movement from reference spots

plane and use Fourier transform to detect the periods. However, as shown in Figure 2(b) and Figure 2(c), there is no strong signal corresponding to the correct period because such method is sensitive to the spatial noise. If the object does not follow more or less the same hunting *route* every day, the period can hardly be detected. However, in real cases, few objects repeat the exactly same route in the periodic movement.

Our key observation is that, if we view the data from the den, the period is easier to be detected. In Figure 2(d), we transform the movement into a binary sequence, where 1 represents the animal is at den and 0 when it goes out. It is easy to see the regularity in this binary sequence. Our idea is to find some important reference locations, namely *reference spots* , to view the movement. In this example, the den serves as our reference spot.

The notion of reference spots has several merits. First, it *filters out the spatial noise* and turns the period detection problem from a 2-dimensional space (*i.e.*, spatial) to a 1-dimensional space (*i.e.*, binary). As shown in Figure 2(d), we do not care where the animal goes when it is out of the den. As long as it follows a regular pattern going out and coming back to the den, there is a period associated with the den. Second, we can detect *multiple* periods in the movement. Consider the scenario that there is a daily period with one reference spot and a weekly period with another reference spot, it is possible that only period “day” is discovered because the shorter

period will repeat more times. But if we view the movement from two reference spots separately, both periods can be individually detected. Third, based on the assumption that each periodic behavior is associated with some reference locations, all the periods can be found through reference spots.

The rest of this section will discuss in details how to find reference spots and detect the periods on the binary sequence for each reference spot.

Finding Reference Spots. Since an object with periodic movement will repeatedly visit some specific places, if we only consider the spatial information of the movement, reference spots are those dense regions containing more points than the other regions. Note that the reference spots are obtained for each individual object.

Many methods can be applied to detect the reference spots, such as density-based clustering. The methods could vary according to different applications. We adapt a popular kernel method [24], which is designed for the purpose of finding home ranges of animals. For human movement, we may use important location detection methods in [14, 31].

While computing the density for each location in a continuous space is computationally expensive, we discretize the space into a regular $w \times h$ grid and compute the density for each cell. The grid size is determined by the desired resolution to view the spatial data. If an animal has frequent activities at one place, this place will have higher probability to be its home. This actually aligns very well with our definition of reference spots.

For each grid cell c, the density is estimated using the bivariate normal density kernel,

$$f(c) = \frac{1}{n\gamma^2} \sum_{i=1}^{n} \frac{1}{2\pi} \exp(-\frac{|c - loc_i|^2}{2\gamma^2}),$$

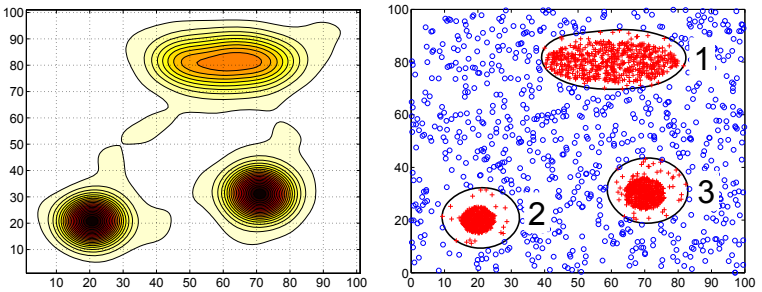
where $|c - loc_i|$ is the distance between cell c and location loc_i. In addition, γ is a smoothing parameter which is determined by the following heuristic method [2],

$$\gamma = \frac{1}{2}(\sigma_x^2 + \sigma_y^2)^{\frac{1}{2}} n^{-\frac{1}{6}},$$

where σ_x and σ_y are the standard deviations of the whole sequence *LOC* in its x and y-coordinates, respectively. The time complexity for this method is $O(w \cdot h \cdot n)$.

After obtaining the density values, a reference spot can be defined by a contour line on the map, which joins the cells of the equal density value, with some density threshold. The threshold can be determined as the top-$p\%$ density value among all the density values of all cells. The larger the value p is, the bigger the size of reference spot is. In practice, p can be chosen based on prior knowledge about the size of the reference spots. In many real applications, we can assume that the reference spots are usually very small on a large map (e.g., within 10% of whole area). So, by setting $p\% = 15\%$, most parts of reference spots should be detected with high probability.

To illustrate this idea, assume that a bird stays in a nest for half a year and moves to another nest staying for another half year. At each nest, it has a daily periodic



(a) Density map calculated by kernel method.

(b) Reference spots defined by contours.

Fig. 3 Finding reference spots

behavior of going out for food during the daytime and coming back to the nest at night, as shown in Figure 3. Note that the two small areas (spot #2 and spot #3) are the two nests and the bigger region is the food resource (spot #1). Figure 3(a) shows the density calculated using the kernel method. The grid size is 100×100. The darker the color is, the higher the density is. Figure 3(b) is the reference spots identified by contour using top-15% density value threshold.

Periods Detection on Binary Sequence. Given a set of reference spots, we further propose a method to obtain the potential periods within *each* spot *separately*. Viewed from a single reference spot, the movement sequence now can be transformed into a binary sequence $B = b_1 b_2 \dots b_n$, where $b_i = 1$ when this object is within the reference spot at timestamp i and 0 otherwise. In discrete signal processing area, to detect periods in a sequence, the most popular methods are Fourier transform and autocorrelation, which essentially complement each other in the following sense, as discussed in [21]. On one hand, Fourier transform often suffers from the low resolution problem in the low frequency region, hence provides poor estimation of large periods. Also, the well-known spectral leakage problem of Fourier transform tends to generate a lot of false positives in the periodogram. On the other hand, autocorrelation offers accurate estimation for both short and large periods, but is more difficult to set the significance threshold for important periods. Consequently, [21] proposed to combine Fourier transform and autocorrelation to find periods. Here, we adapt this approach to find periods in the binary sequence B.

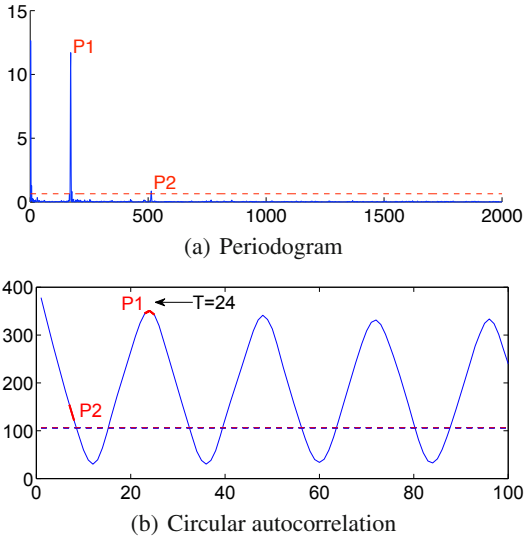
In Discrete Fourier Transform (DFT), the sequence $B = b_1 b_2 \dots b_n$ is transformed into the sequence of n complex numbers $X_1, X_2, \dots, X_n$. Given coefficients X, the periodogram is defined as the squared length of each Fourier coefficient: $F_k = \|X_k\|^2$. Here, F_k is the power of frequency k. In order to specify which frequencies are important, we need to set a threshold and identify those higher frequencies than this threshold.

The threshold is determined using the following method. Let B' be a randomly permutated sequence from B. Since B' should not exhibit any periodicities, even the

maximum power does not indicate the period in the sequence. Therefore, we record its maximum power as p_{max}, and only the frequencies in B that have higher power than p_{max} may correspond to real periods. To provide a 99% confidence level on what frequencies are important, we repeat the above random permutation experiment 100 times and record the maximum power of each permutated sequence. The 99-th largest value of these 100 experiments will serve as a good estimator of the power threshold.

Given that F_k is larger than the power threshold, we still need to determine the exact period in the time domain, because a single value k in *frequency domain* corresponds to a range of periods $[\frac{n}{k}, \frac{n}{k-1})$ in *time domain*. In order to do this, we use circular autocorrelation, which examines how similar a sequence is to its previous values for different τ lags: $R(\tau) = \sum_{i=1}^{n} b_\tau b_{i+\tau}$.

Thus, for each period range $[l,r)$ given by the periodogram, we test whether there is a peak in $\{R(l), R(l+1), \ldots, R(r-1)\}$ by fitting the data with a quadratic function. If the resulting function is concave in the period range, which indicates the existence of a peak, we return $t^* = \arg\max_{l \le t < r} R(t)$ as a detected period. Similarly, we employ a 99% confidence level to eliminate false positives caused by noise.



(a) Periodogram

(b) Circular autocorrelation

Fig. 4 Finding periods

In Figure 4(a), we show the periodogram of reference spot #2 in Figure 3. The red dashed line denotes the threshold of 99% confidence. There are two points P_1 and P_2 that are above the threshold. In Figure 4(b), P_1 and P_2 are mapped to a range of periods. We can see that there is only one peak, P_1, corresponding to $T = 24$ on the autocorrelation curve.

2.2.3 Modeling Periodic Behaviors

After obtaining the periods for each reference spot, now we study the task how to mine periodic behaviors. We will consider the reference spots with the same period together in order to obtain more concise and informative periodic behaviors. But, since a behavior may only exist in a *partial* movement, there could be several periodic behaviors with the same period. For example, there are two daily behaviors in David's movement: One corresponds to the school days and the other occurs during the summer. However, given a long history of movement and a period as a "day", we actually do not know how many periodic behaviors exist in this movement and which days belong to which periodic behavior. This motivates us to use a clustering method. Because the "days" that belong to the same periodic behavior should have the similar temporal location pattern. We propose a generative model to measure the distance between two "days". Armed with such distance measure, we can further group the "days" into several clusters and each cluster represents one periodic behavior. As in David's example, "school days" should be grouped into one cluster and "summer days" should be grouped into another one. Note that, we assume that for each period, such as "day", one "day" will *only* belong to one behavior.

Since every period in the movement will be considered separately, *the rest of this section will focus on one specific period T*. First, we retrieve all the reference spots with period T. By combining the reference spots with the same period together, we will get a more informative periodic behaviors associated with different reference spots. For example, we can summarize David's daily behavior as "9:00∼18:00 at office *and* 20:00∼8:00 in the dorm". We do not consider combining two different periods in current work.

Let $O_T = \{o_1, o_2, \ldots, o_d\}$ denote reference spots with period T. For simplicity, we denote o_0 as any other locations outside the reference spots $o_1, o_2, \ldots, o_d$. Given $LOC = loc_1 loc_2 \cdots loc_n$, we generate the corresponding *symbolized movement sequence* $S = s_1 s_2 \ldots s_n$, where $s_i = j$ if loc_i is within o_j. S is further segmented into $m = \lfloor \frac{n}{T} \rfloor$ *segments*[1]. We use I^j to denote the j-th segment and t_k $(1 \leq k \leq T)$ to denote the k-th relative timestamp in a period. $I^j_k = i$ means that the object is within o_i at t_k in the j-th segment. For example, for $T = 24$ (hours), a segment represents a "day", t_9 denotes 9:00 in a day, and $I^5_9 = 2$ means that the object is within o_2 at 9:00 in the 5-th day. Naturally, we may use the categorical distribution to model the probability of such events.

Definition 2 (Categorical Distribution Matrix). Let $\mathscr{T} = \{t_1, t_2, \ldots, t_T\}$ be a set of relative timestamps, x_k be the categorical random variable indicating the selection of reference spot at timestamp t_k. $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_T]$ is a categorical distribution matrix with each column $\mathbf{p}_k = [p(x_k = 0), p(x_k = 1), \ldots, p(x_k = d)]^T$ being an independent categorical distribution vector satisfying $\sum_{i=0}^{d} p(x_k = i) = 1$.

Now, suppose I^1, I^2, …, I^l follow the same periodic behavior. The probability that the segment set $\mathscr{I} = \bigcup_{j=1}^{l} I^j$ is generated by some distribution matrix $\mathbf{P}$ is

[1] If n is not a multiple of T, then the last ($n \mod T$) positions are truncated.

$$P(\mathscr{I}|\mathbf{P}) = \prod_{I^j \in \mathscr{I}} \prod_{k=1}^{T} p(x_k = I_k^j).$$

Now, we formally define the concept of periodic behavior.

Definition 3 (Periodic Behavior). Let $\mathscr{I}$ be a set of segments. A periodic behavior over all the segments in $\mathscr{I}$, denoted as $\mathbf{H}(\mathscr{I})$, is a pair $\langle T, \mathbf{P} \rangle$. T is the period and $\mathbf{P}$ is a probability distribution matrix. We further let $|\mathscr{I}|$ denote the number of segments covered by this periodic behavior.

2.2.4 Discovery of Periodic Behaviors

With the definition of periodic behaviors, we are able to estimate periodic behaviors over a set of segments. Now given a set of segments $\{I^1, I^2, \ldots, I^m\}$, we need to discover which segments are generated by the same periodic behavior. Suppose there are K underlying periodic behaviors, each of which exists in a partial movement, the segments should be partitioned into K groups so that each group represents one periodic behavior.

A potential solution to this problem is to apply some clustering methods. In order to do this, a distance measure between two periodic behaviors needs to be defined. Since a behavior is represented as a pair $\langle T, \mathbf{P} \rangle$ and T is fixed, the distance should be determined by their probability distribution matrices. Further, a small distance between two periodic behaviors should indicate that the segments contained in each behavior are likely to be generated from the same periodic behavior.

Several measures between the two probability distribution matrices $\mathbf{P}$ and $\mathbf{Q}$ can be used to fulfill these requirements. Here, since we assume the independence of variables across different timestamps, we propose to use the well-known Kullback-Leibler divergence as our distance measure:

$$KL(\mathbf{P}\|\mathbf{Q}) = \sum_{k=1}^{T} \sum_{i=0}^{d} p(x_k = i) \log \frac{p(x_k = i)}{q(x_k = i)}.$$

When $KL(\mathbf{P}\|\mathbf{Q})$ is small, it means that the two distribution matrices $\mathbf{P}$ and $\mathbf{Q}$ are similar, and vice versa.

Note that $KL(\mathbf{P}\|\mathbf{Q})$ becomes infinite when $p(x_k = i)$ or $q(x_k = i)$ has zero probability. To avoid this situation, we add to $p(x_k = i)$ (and $q(x_k = i)$) a background variable u which is uniformly distributed among all reference spots,

$$p(x_k = i) = (1 - \lambda) p(x_k = i) + \lambda u, \tag{3}$$

where λ is a small smoothing parameter $0 < \lambda < 1$.

Now, suppose we have two periodic behaviors, $\mathbf{H}_1 = \langle T, \mathbf{P} \rangle$ and $\mathbf{H}_2 = \langle T, \mathbf{Q} \rangle$. We define the distance between these two behaviors as

$$dist(\mathbf{H}_1, \mathbf{H}_2) = KL(\mathbf{P}\|\mathbf{Q}).$$

Suppose there exist K underlying periodic behaviors. There are many ways to group the segments into K clusters with the distance measure defined. However, the number of underlying periodic behaviors (*i.e.*, K) is usually unknown. So we propose a hierarchical agglomerative clustering method to group the segments while at the same time determine the optimal number of periodic behaviors. At each iteration of the hierarchical clustering, two clusters with the minimum distance are merged. In Algorithm 2, we first describe the clustering method assuming K is given. We will return to the problem of selecting optimal K later.

Algorithm 2. Mining periodic behaviors.

INPUT: symbolized sequence S, period T, number of clusters K.
OUTPUT: K periodic behaviors.
ALGORITHM:

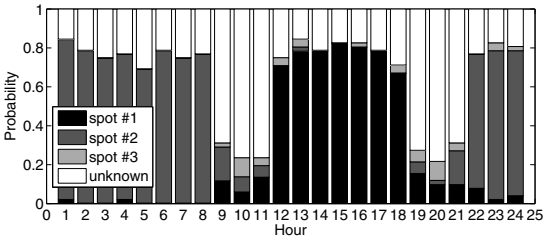
1: segment S into m segments;
2: initialize $k = m$ clusters, each of which has one segment;
3: compute the pairwise distances among $C_1, \ldots, C_k$, $d_{ij} = dist(\mathbf{H}(C_i), \mathbf{H}(C_j))$;
4: **while** $(k > K)$ **do**
5: select d_{st} such that $s,t = \arg\min_{i,j} d_{ij}$;
6: merge clusters C_s and C_t to a new cluster C;
7: calculate the distances between C and the remaining clusters;
8: $kv = k - 1$;
9: **end while**
10: return $\{\mathbf{H}(C_i), 1 \leq i \leq K\}$.

Algorithm 2 illustrates the hierarchical clustering method. It starts with m clusters (Line 1). A cluster C is defined as a collection of segments. At each iteration, two clusters with the minimum distance are merged (Lines 4∼8). When two clusters are merged, the new cluster inherits the segments that owned by the original clusters C_s and C_t. It has a newly built behavior $\mathbf{H}(C) = \langle T, \mathbf{P} \rangle$ over the merged segments, where $\mathbf{P}$ is computed by the following updating rule:

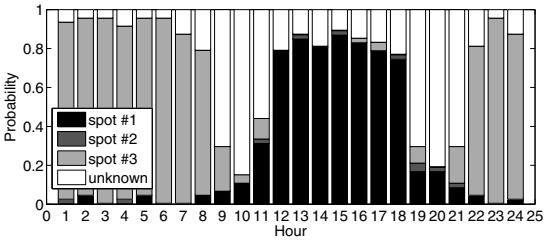
$$\mathbf{P} = \frac{|C_s|}{|C_s| + |C_t|}\mathbf{P}_s + \frac{|C_t|}{|C_s| + |C_t|}\mathbf{P}_t. \tag{4}$$

Finally, K periodic behaviors are returned (Line 9).

To illustrate the method, we again use the example shown in Figure 3. There are two periodic behaviors with period $T = 24$ (hours) in the bird's movement. Figure 5 shows the probability distribution matrix for each discovered periodic behavior. A close look at Figure 5(a) shows that at time 0:00∼8:00 and 22:00∼24:00, the bird has a high probability being at reference spot #2, which is a nest shown in Figure 3(b). At time 12:00∼18:00, it is very likely to be at reference spot #1, which is the food resources shown in Figure 3(b). And at the time 9:00∼11:00, there are also some probability that the bird is at reference spot #1 or reference spot #2. This indicates the bird goes out of the nest around 8:00 and arrives at the food resources



(a) **P** of periodic behavior #1



(b) **P** of periodic behavior #2

Fig. 5 Periodic behaviors

place around 12:00. Such periodic behaviors well represent the bird's movement and truly reveal the mechanism we employed to generate this synthetic data.

Now, we discuss how to pick the appropriate parameter K. Ideally, during the hierarchical agglomerative clustering, the segments generated from the same behavior should be merged first because they have smaller KL-divergence distance. Thus, we judge a cluster is good if all the segments in the cluster are concentrated in one single reference spot at a particular timestamp. Hence, a natural representation error measure to evaluate the representation quality of a cluster is as follows. Note that here we exclude the reference spot o_0 which essentially means the location is unknown.

Definition 4 (Representation Error). Given a set of segments $C = \{I^1, I^2, \ldots, I^l\}$ and its periodic behavior $\mathbf{H}(C) = \langle T, \mathbf{P} \rangle$, the representation error is,

$$E(C) = \frac{\sum_{I^j \in C} \sum_{i=1}^{T} \mathbf{1}_{I_i^j \neq 0} \cdot (1 - p(x_i = I_i^j))}{\sum_{I^j \in C} \sum_{i=1}^{T} \mathbf{1}_{I_i^j \neq 0}}.$$

At each iteration, all the segments are partitioned into k clusters $\{C_1, C_2, \ldots, C_k\}$. The overall representation error at current iteration is calculated as the mean over all clusters,

$$\mathscr{E}_k = \frac{1}{k} \sum_{i=1}^{k} E(C_i).$$

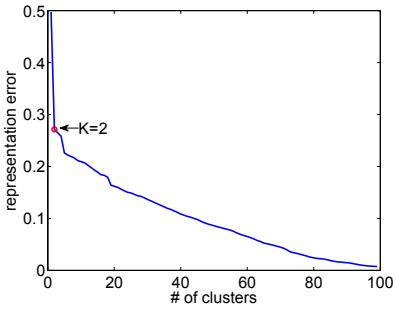


Fig. 6 Representation error

During the clustering process, we monitor the change of $\mathscr{E}_k$. If $\mathscr{E}_k$ exhibits dramatical increases comparing with $\mathscr{E}_{k-1}$, it is a sign the newly merged cluster may contain two different behaviors and $k-1$ is likely to be a good choice of K. The degree of such change can be observed from the derivative of $\mathscr{E}$ over k, $\frac{\partial \mathscr{E}}{\partial k}$. Since a sudden increase of $\mathscr{E}$ will result in a peak in its derivative, we can find the optimal K as $K = \arg\max_k \frac{\partial \mathscr{E}}{\partial k}$.

As we can see in Figure 6, the representation error suddenly increases at $k = 2$ for the bird's movement. This indicates that there are actually two periodic behaviors in the movement. This is true because the bird has one daily periodic behavior at the first nest and later has another one at the second nest.

3 Mining Periodicity from Incomplete Observations

So far, we have presented a complete framework, Periodica, for mining periodic behaviors from spatio-temporal data. Using the notion of reference spots, Periodica is able to discover complex periodic behaviors from real-world movement data. Nevertheless, we note that Periodica still relies on traditional periodicity analysis methods, namely Fourier transform and auto-correlation [18, 21, 5, 12], to detect periods after the movement data is converted to binary sequences. A fundamental assumption of all the traditional periodicity analysis methods is that they require the data to be *evenly sampled*, that is, there is an observation at every timestamp.

Unfortunately, due to the *limitations of data collection devices and methods*, this seemingly weak assumption is often seriously violated in practice. For example, a bird can only carry small sensors with one or two reported locations in three to five days. And the locations of a person may only be recorded when he uses his cellphone. Moreover, if a sensor is not functioning or a tracking facility is turned off, it could result in a large portion of missing data. Therefore, we usually have *incomplete observations*, which *are unevenly sampled* and *have large portion of missing data*. In fact, the issue with incomplete observations is a common problem on data collected from GPS and sensors, making period detection an even more challenging problem.

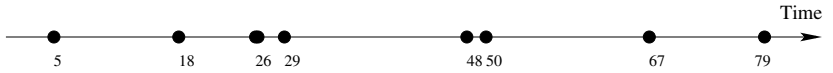


Fig. 7 Incomplete observations

To illustrate the difficulties, let us first take a look at Figure 7. Suppose we have observed the occurrences of an event at timestamps 5, 18, 26, 29, 48, 50, 67, and 79. The observations of the event at other timestamps are not available. It is certainly not an easy task to infer the period directly from these *incomplete* observations. Even though some extensions of Fourier transform have been proposed to handle uneven data samples [15, 19], they are still not applicable to the case with very low sampling rate.

Besides, the periodic behaviors could be inherently *complicated and noisy*. A periodic event does not necessarily happen at *exactly* the same timestamp in each periodic cycle. For example, the time that a person goes to work in the morning might *oscillate* between 8:00 to 10:00. *Noises* could also occur when the "in office" event is expected to be observed on a weekday but fails to happen.

In this section, we take a completely different approach to the period detection problem and handle all the aforementioned difficulties occurring in data collection process and periodic behavior complexity in a unified framework. The basic idea of our method is illustrated in Example 1.

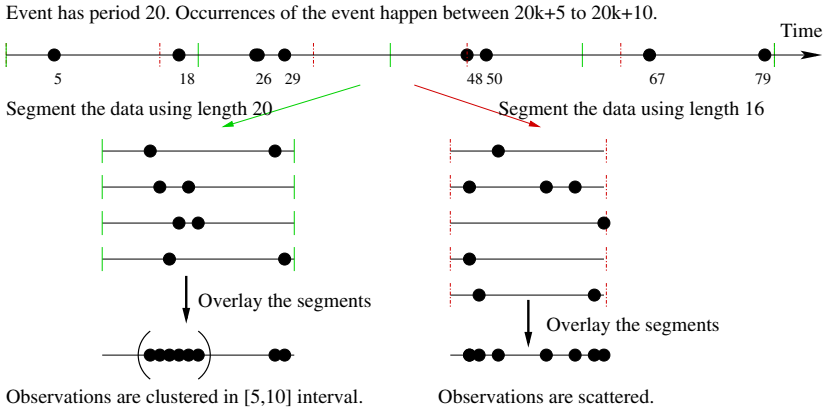


Fig. 8 Illustration example of our method

Example 1. Suppose an event has a period $T = 20$ and we have eight observations of the event, as shown in Figure 8. If we overlay the observations with the correct period $T = 20$, we can see that most of the observations concentrate in time interval [5,10]. On the contrary, if we overlay the points with a wrong period, say $T = 16$, we cannot observe such clusters.

As suggested by Example 1, we could segment the timeline using a potential period T and summarize the observations over all the segments. If most of the observations fall into some time intervals, such as interval $[5,10]$ in Example 1, T is *likely* to be the true period. In this section, we formally characterize such likelihood by introducing a probabilistic model for periodic behaviors. The model naturally handles the oscillation and noise issues because the occurrence of an event at any timestamp is now modeled with a probability. Next, we propose a new measure for periodicity based on this model. The measure essentially examines whether the distribution of observations is highly skewed w.r.t. a potential period T. As we will see later, even when the observations are incomplete, the overall distribution of observations, after overlaid with the correct T, remains skewed and is similar to the true periodic behavior model.

In summary, our major contributions are as follows. First, we introduce a probabilistic model for periodic behaviors and a random observation model for incomplete observations. This enables us to model all the variations we encounter in practice in a unified framework. Second, we propose a novel probabilistic measure for periodicity and design a practical algorithm to detect periods directly from the raw data. We further give rigorous proof of its validity under both the probabilistic periodic behavior model and the random observation model. Finally, we point out that our method can be used to detect periodicity for any temporal events, not necessarily restricting to movement data.

3.1 Problem Definition

Now we formally define the problem of period detection for events. We first assume that there is an observation at every timestamp. The case with incomplete observations will be discussed in Section 3.2.2. We use a binary sequence $\mathscr{X} = \{x(t)\}_{t=0}^{n-1}$ to denote observations. For example, if the event is "in the office", $x(t) = 1$ means this person is in the office at time t and $x(t) = 0$ means this person is *not* in the office at time t. Later we will refer $x(t) = 1$ as a *positive observation* and $x(t) = 0$ as a *negative observation*.

Definition 5 (Periodic Sequence). A sequence $\mathscr{X} = \{x(t)\}_{t=0}^{n-1}$ is said to be periodic if there exists some $T \in \mathbb{Z}$ such that $x(t+T) = x(t)$ for all values of t. We call T a period of $\mathscr{X}$.

A fundamental ambiguity with the above definition is that if T is a period of $\mathscr{X}$, then mT is also a period of $\mathscr{X}$ for any $m \in \mathbb{Z}$. A natural way to resolve this problem is to use the so called *prime period*.

Definition 6 (Prime Period). The prime period of a periodic sequence is the smallest $T \in \mathbb{Z}$ such that $x(t+T) = x(t)$ for all values of t.

For the rest of the section, unless otherwise stated, we always refer the word "period" to "prime period".

As we mentioned before, in real applications the observed sequences always deviate from the perfect periodicity due to the oscillating behavior and noises. To

model such deviations, we introduce a new probabilistic framework, which is based on the *periodic distribution vectors* as defined below.

Definition 7 (Periodic Distribution Vector). For any vector $\mathbf{p}^T = [p_0^T, \ldots, p_{T-1}^T] \in [0,1]^T$ other than $\mathbf{0}^T$ and $\mathbf{1}^T$, we call it a periodic distribution vector of length T. A binary sequence $\mathscr{X}$ is said to be generated according to $\mathbf{p}^T$ if $x(t)$ is independently distributed according to Bernoulli$(p^T{}_{\mathrm{mod}\,(t,T)})$.

Here we need to exclude the trivial cases where $\mathbf{p}^T = \mathbf{0}^T$ or $\mathbf{1}^T$. Also note that if we restrict the value of each p_i^T to $\{0,1\}$ only, then the resulting $\mathscr{X}$ is *strictly* periodic according to Definition 5. We are now able to formulate our period detection problem as follows.

Problem 1 (Event Period Detection). Given a binary sequence $\mathscr{X}$ generated according to any periodic distribution vector $\mathbf{p}^{T_0}$, find T_0.

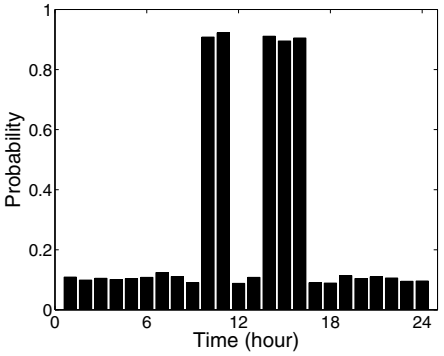


Fig. 9 (Running Example) Periodic distribution vector of a event with daily periodicity $T_0 = 24$

Example 2 (Running Example). We will use a running example throughout the section to illustrate our method. Assume that a person has a daily periodicity visiting his office during 10am-11am and 2pm-4pm. His observation sequence is generated from the periodic distribution vector with high probabilities at time interval [10:11] and [14:16] and low but nonzero probabilities at other timestamps, as shown in Figure 9.

3.2 A Probabilistic Model for Period Detection

As we see in Example 8, when we overlay the binary sequence with its true period T_0, the resulting sequence correctly reveals its underlying periodic behavior. Now we make this observation formal using the concept of periodic distribution vector. Then, we propose a novel probabilistic measure of periodicity based on this observation and prove its validity even when observations are incomplete.

3.2.1 A Probabilistic Measure of Periodicity

Given a binary sequence $\mathscr{X}$, we define $S^+ = \{t : x(t) = 1\}$ and $S^- = \{t : x(t) = 0\}$ as the collections of timestamps with 1's and 0's, respectively. For a candidate period T, let $\mathscr{I}_T$ denote the power set of $[0 : T-1]$. Then, for any set of timestamps (*possibly non-consecutive*) $I \in \mathscr{I}_T$, we can define the collections of original timestamps that fall into this set after overlay as follows:

$$S_I^+ = \{t \in S^+ : \mathscr{F}_T(t) \in I\}, \quad S_I^- = \{t \in S^- : \mathscr{F}_T(t) \in I\},$$

where $\mathscr{F}_T(t) = \mod(t,T)$, and further compute the ratios of 1's and 0's whose corresponding timestamps fall into I after overlay:

$$\mu_{\mathscr{X}}^+(I,T) = \frac{|S_I^+|}{|S^+|}, \quad \mu_{\mathscr{X}}^-(I,T) = \frac{|S_I^-|}{|S^-|}. \tag{5}$$

The following lemma says that these ratios indeed reveal the true underlying probabilistic model parameters, given that the observation sequence is sufficiently long.

Lemma 1. *Suppose a binary sequence $\mathscr{X} = \{x(t)\}_{t=0}^{n-1}$ is generated according to some periodic distribution vector $\mathbf{p}^T$ of length T, write $q_i^T = 1 - p_i^T$. Then $\forall I \in \mathscr{I}_T$,*

$$\lim_{n\to\infty} \mu_{\mathscr{X}}^+(I,T) = \frac{\sum_{i\in I} p_i^T}{\sum_{i=0}^{T-1} p_i^T}, \quad \lim_{n\to\infty} \mu_{\mathscr{X}}^-(I,T) = \frac{\sum_{i\in I} q_i^T}{\sum_{i=0}^{T-1} q_i^T}.$$

Proof. The proof is a straightforward application of the Law of Large Numbers (LLN), and we only prove the first equation. With a slight abuse of notation we write $S_i = \{t : \mathscr{F}_T(t) = i\}$ and $S_i^+ = \{t \in S^+ : \mathscr{F}_T(t) = i\}$. Since $\{x(t) : t \in S_i\}$ are i.i.d. Bernoulli(p_i^T) random variables, by LLN we have

$$\lim_{n\to\infty} \frac{|S_i^+|}{n} = \lim_{n\to\infty} \frac{\sum_{t\in S_i} x(t)}{n} = \frac{p_i^T}{T},$$

where we use $\lim_{n\to\infty} \frac{|S_i|}{n} = \frac{1}{T}$ for the last equality. So,

$$\lim_{n\to\infty} \mu_{\mathscr{X}}^+(I,T) = \lim_{n\to\infty} \frac{|S_I^+|/n}{|S^+|/n} = \lim_{n\to\infty} \frac{\sum_{i\in I} |S_i^+|/n}{\sum_{i=0}^{T-1} |S_i^+|/n} = \frac{\sum_{i\in I} p_i^T/T}{\sum_{i=0}^{T-1} p_i^T/T} = \frac{\sum_{i\in I} p_i^T}{\sum_{i=0}^{T-1} p_i^T}.$$

Now we introduce our measure of periodicity based on Lemma 1. For any $I \in \mathscr{I}_T$, its discrepancy score is defined as:

$$\Delta_{\mathscr{X}}(I,T) = \mu_{\mathscr{X}}^+(I,T) - \mu_{\mathscr{X}}^-(I,T). \tag{6}$$

Then, the periodicity measure of $\mathscr{X}$ w.r.t. period T is:

$$\gamma_{\mathscr{X}}(T) = \max_{I\in\mathscr{I}_T} \Delta(I,T). \tag{7}$$

It is obvious that $\gamma_{\mathscr{X}}(T)$ is bounded: $0 \le \gamma_{\mathscr{X}}(T) \le 1$. Moreover, $\gamma_{\mathscr{X}}(T) = 1$ if and only if $\mathscr{X}$ is strictly periodic with period T. But more importantly, we have the following lemma, which states that under our probabilistic periodic behavior model, $\gamma_{\mathscr{X}}(T)$ is indeed a desired measure of periodicity.

Lemma 2. *If a binary sequence $\mathscr{X}$ is generated according to any periodic distribution vector $\mathbf{p}^{T_0}$ for some T_0, then*

$$\lim_{n\to\infty} \gamma_{\mathscr{X}}(T) \le \lim_{n\to\infty} \gamma_{\mathscr{X}}(T_0), \quad \forall T \in \mathbb{Z}.$$

Proof. Define

$$c_i = \frac{p_i^{T_0}}{\sum_{k=0}^{T_0-1} p_k^{T_0}} - \frac{q_i^{T_0}}{\sum_{k=0}^{T_0-1} q_k^{T_0}},$$

it is easy to see that the value $\lim_{n\to\infty} \gamma_{\mathscr{X}}(T_0)$ is achieved by $I^* = \{i \in [0, T_0-1] : c_i > 0\}$. So it suffices to show that for any $T \in \mathbb{Z}$ and $I \in \mathscr{I}_T$,

$$\lim_{n\to\infty} \Delta_{\mathscr{X}}(I,T) \le \lim_{n\to\infty} \Delta_{\mathscr{X}}(I^*,T_0) = \sum_{i\in I^*} c_i.$$

Observe now that for any (I,T),

$$\lim_{n\to\infty} \mu^+_{\mathscr{X}}(I,T) = \sum_{i\in I} \left(\frac{1}{T} \sum_{j=0}^{T_0-1} \frac{p^{T_0}_{\mathscr{F}_{T_0}(i+j\times T)}}{\sum_{k=0}^{T_0-1} p_k^{T_0}} \right),$$

$$\lim_{n\to\infty} \mu^-_{\mathscr{X}}(I,T) = \sum_{i\in I} \left(\frac{1}{T} \sum_{j=0}^{T_0-1} \frac{q^{T_0}_{\mathscr{F}_{T_0}(i+j\times T)}}{\sum_{k=0}^{T_0-1} q_k^{T_0}} \right).$$

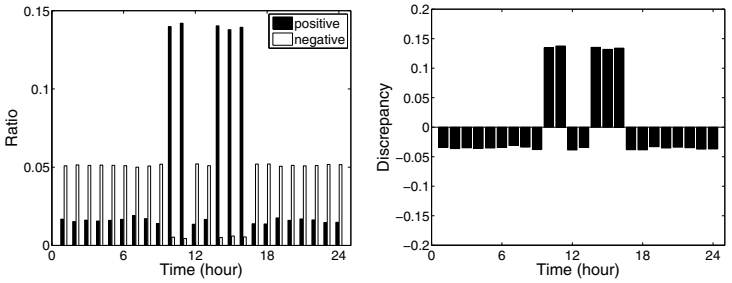
Therefore we have

$$\begin{aligned}
\lim_{n\to\infty} \Delta_{\mathscr{X}}(I,T) &= \frac{1}{T} \sum_{i\in I} \sum_{j=0}^{T_0-1} \left(\frac{p^{T_0}_{\mathscr{F}_{T_0}(i+j\times T)}}{\sum_{k=0}^{T_0-1} p_k^{T_0}} - \frac{q^{T_0}_{\mathscr{F}_{T_0}(i+j\times T)}}{\sum_{k=0}^{T_0-1} q_k^{T_0}} \right) \\
&= \frac{1}{T} \sum_{i\in I} \sum_{j=0}^{T_0-1} c_{\mathscr{F}_{T_0}(i+j\times T)} \le \frac{1}{T} \sum_{i\in I} \sum_{j=0}^{T_0-1} \max(c_{\mathscr{F}_{T_0}(i+j\times T)}, 0) \\
&\le \frac{1}{T} \sum_{j=0}^{T_0 T-1} \max(c_{\mathscr{F}_{T_0}(i+j\times T)}, 0) = \frac{1}{T} \times T \sum_{i\in I^*} c_i = \sum_{i\in I^*} c_i,
\end{aligned}$$

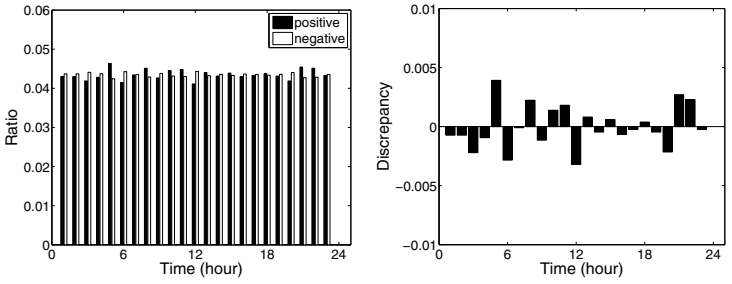
where the third equality uses the definition of I^*.

Note that, similar to the deterministic case, the ambiguity of multiple periods still exists as we can easily see that $\lim_{n\to\infty} \gamma_{\mathscr{X}}(mT_0) = \lim_{n\to\infty} \gamma_{\mathscr{X}}(T_0)$ for all m$\in \mathbb{Z}$. But we are only interested in finding the smallest one.

Example 3 (Running Example (cont.)). When we overlay the sequence using potential period $T = 24$, Figure 10(a) shows that positive observations have high



(a) Ratios ($T = 24$) (b) Discrepancy ($T = 24$)



(c) Ratios ($T = 23$) (d) Discrepancy ($T = 23$)

Fig. 10 (a) and (c): Ratios of 1's and 0's at a single timestamp (*i.e.*, $\mu^{+}_{\mathscr{X}}(\cdot, T)$ and $\mu^{-}_{\mathscr{X}}(\cdot, T)$) when $T = 24$ and $T = 23$, respectively. (b) and (d): Discrepancy scores at a single timestamp (*i.e.* $\Delta_{\mathscr{X}}(\cdot, T)$) when $T = 24$ and $T = 23$.

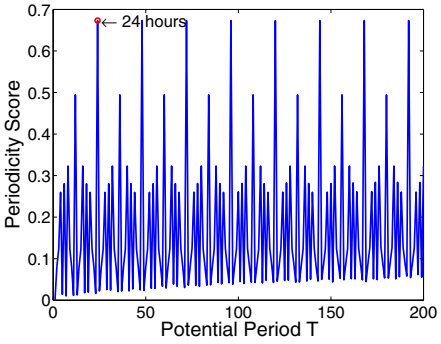


Fig. 11 Periodicity scores of potential periods

probability to fall into the set of timestamps: $\{10, 11, 14, 15, 16\}$. However, when using the wrong period $T = 23$, the distribution is almost uniform over time, as shown in Figure 10(c). Similarly, we see large discrepancy scores for T = 24 (Figure 10(b)) whereas the discrepancy scores are very small for T = 23 (Figure 10(d)). Therefore, we will have $\gamma_{\mathscr{X}}(24) > \gamma_{\mathscr{X}}(23)$. Figure 11 shows the periodicity scores for all

potential periods in $[1:200]$. We can see that the score is maximized at $T=24$, which is the true period of the sequence.

3.2.2 Random Observation Model

Next, we extend our analysis on the proposed periodicity measure to the case of incomplete observations with a random observation model. To this end, we introduce a new label "-1" to the binary sequence $\mathscr{X}$ which indicates that the observation is unavailable at a specific timestamp. In the random observation model, each observation $x(t)$ is associated with a probability $d_t \in [0,1]$ and we write $\mathbf{d} = \{d_t\}_{t=0}^{n-1}$.

Definition 8. A sequence $\mathscr{X}$ is said to be generated according to $(\mathbf{p}^T, \mathbf{d})$ if

$$x(t) = \begin{cases} \text{Bernoulli}(p^T_{\mathscr{F}_T(t)}) & \text{w.p. } d_t \\ -1 & \text{w.p. } 1-d_t \end{cases} \tag{8}$$

In general, we may assume that each d_t is independently drawn from some fixed but unknown distribution f over the interval $[0,1]$. To avoid the trivial case where $d_t \equiv 0$ for all t, we further assume that it has nonzero mean: $\rho_f > 0$. Although this model seems to be very flexible, in the section we prove that our periodicity measure is still valid. In order to do so, we need the following lemma, which states that $\mu^+_{\mathscr{X}}(I,T)$ and $\mu^-_{\mathscr{X}}(I,T)$ remain the same as before, assuming infinite length observation sequence.

Lemma 3. *Suppose* $\mathbf{d} = \{d_t\}_{t=0}^{n-1}$ *are i.i.d. random variables in* $[0,1]$ *with nonzero mean, and a sequence* $\mathscr{X}$ *is generated according to* $(\mathbf{p}^T, \mathbf{d})$*, write* $q_i^T = 1 - p_i^T$*. Then* $\forall I \in \mathscr{I}_T$*,*

$$\lim_{n\to\infty} \mu^+_{\mathscr{X}}(I,T) = \frac{\sum_{i\in I} p_i^T}{\sum_{i=0}^{T-1} p_i^T}, \quad \lim_{n\to\infty} \mu^-_{\mathscr{X}}(I,T) = \frac{\sum_{i\in I} q_i^T}{\sum_{i=0}^{T-1} q_i^T}.$$

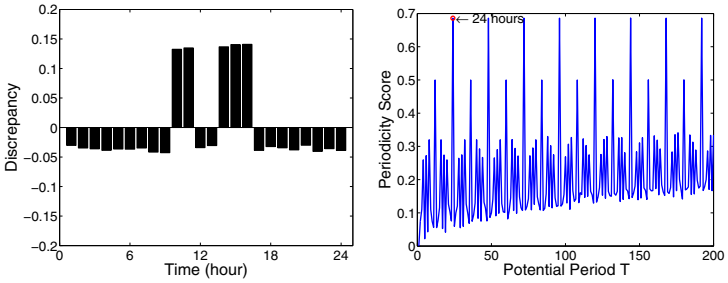
Proof. We only prove the first equation. Let $y(t)$ be a random variable distributed according to Bernoulli(d_t) and $z(t) = x(t)y(t)$. Then $\{z(t)\}_{t=0}^{n-1}$ are independent random variables which take value in $\{0,1\}$, with mean $\mathbb{E}[z(t)]$ computed as follows:

$$\begin{aligned} \mathbb{E}[z(t)] &= \mathbf{P}(z(t)=1) = \mathbf{P}(x(t)=1, y(t)=1) \\ &= \mathbf{P}(x(t)=1 | y(t)=1)\mathbf{P}(y(t)=1) \\ &= p^T_{\mathscr{F}_T(t)}\mathbf{P}(y(t)=1) = p^T_{\mathscr{F}_T(t)}\mathbb{E}[d_t] = p^T_{\mathscr{F}_T(t)}\rho_f. \end{aligned}$$

Define $S_i = \{t : \mathscr{F}_T(t) = i\}$ and $S_i^+ = \{t \in S^+ : \mathscr{F}_T(t) = i\}$, it is easy to see that $|S_i^+| = \sum_{t\in S_i} z(t)$. Using LLN we get

$$\lim_{n\to\infty} \frac{|S_i^+|}{n} = \lim_{n\to\infty} \frac{\sum_{t\in S_i} z(t)}{n} = \frac{p_i^T \rho_f}{T},$$

where we use $\lim_{n\to\infty} \frac{|S_i|}{n} = 1/T$ for the last equality. Therefore,



(a) Discrepancy ($T = 24$) (b) Periodicity scores

Fig. 12 Period detection with unknown observations

$$\lim_{n\to\infty} \mu^+_{\mathscr{X}}(I,T) = \lim_{n\to\infty} \frac{|S^+_I|/n}{|S^+|/n} = \lim_{n\to\infty} \frac{\sum_{i\in I} |S^+_i|/n}{\sum_{i=0}^{T-1} |S^+_i|/n} = \frac{\sum_{i\in I} \frac{p_i^T \rho_f}{T}}{\sum_{i=0}^{T-1} \frac{p_i^T \rho_f}{T}} = \frac{\sum_{i\in I} p_i^T}{\sum_{i=0}^{T-1} p_i^T}.$$

Since our periodicity measure only depends on $\mu^+_{\mathscr{X}}(I,T)$ and $\mu^-_{\mathscr{X}}(I,T)$, it is now straightforward to prove its validity under the random observation model. We summarize our main result as the following theorem.

Theorem 1. *Suppose* $\mathbf{d} = \{d_t\}_{t=0}^{n-1}$ *are i.i.d. random variables in* $[0,1]$ *with nonzero mean, and a sequence* $\mathscr{X}$ *is generated according to any* $(\mathbf{p}^{T_0}, \mathbf{d})$ *for some* T_0*, then*

$$\lim_{n\to\infty} \gamma_{\mathscr{X}}(T) \le \lim_{n\to\infty} \gamma_{\mathscr{X}}(T_0), \quad \forall T \in \mathbb{Z}.$$

The proof is exactly the same as that of Lemma 2 given the result of Lemma 3, hence is omitted here.

Here we make two useful comments on this result. First, the assumption that d_t's are independent of each other plays an important role in the proof. In fact, if this does not hold, the observation sequence could exhibit very different periodic behavior from its underlying periodic distribution vector. But a thorough discussion on this issue is beyond the scope of this book. Second, this result only holds exactly with infinite length sequences. However, it provides a good estimate on the situation with finite length sequences, assuming that the sequences are long enough. Note that this length requirement is particularly important when a majority of samples are missing (*i.e.*, ρ_f is close to 0).

Example 4 (Running Example (cont.)). To introduce random observations, we sample the original sequence with sampling rate 0.2. The generated sequence will have 80% of its entries marked as unknown. Comparing Figure 12(a) with Figure 10(b), we can see very similar discrepancy scores over time. Random sampling has little effect on our period detection method. As shown in Figure 12(b), we can still detect the correct period at 24.

3.2.3 Handling Sequences without Negative Samples

In many real world applications, negative samples may be completely unavailable to us. For example, if we have collected data from a local cellphone tower, we will know that a person is in town when he makes phone call through the local tower. However, we are not sure whether this person is in town or not for the rest of time because he could either be out of town or simply not making any call. In this case, the observation sequence $\mathscr{X}$ takes value in $\{1,-1\}$ only, with -1 indicating the missing entries. In this section, we modify our measure of periodicity to handle this case.

Note that due to the lack of negative samples, $\mu^-_{\mathscr{X}}(I,T)$ can no longer be computed from $\mathscr{X}$. Thus, we need find another quantity to compare $\mu^+_{\mathscr{X}}(I,T)$ with. To this end, consider a binary sequence $\mathscr{U} = \{u(t)\}_{t=0}^{n-1}$ where each $u(t)$ is an i.i.d. Bermoulli(p) random variable for some fixed $p > 0$. It is easy to see that for any T and $I \in \mathscr{I}_T$, we have

$$\lim_{n\to\infty} \mu^+_{\mathscr{U}}(I,T) = \frac{|I|}{T}. \tag{9}$$

This corresponds to the case where the positive samples are evenly distributed over all entries after overlay. So we propose the new discrepancy score of I as follows:

$$\Delta^+_{\mathscr{X}}(I,T) = \mu^+_{\mathscr{X}}(I,T) - \frac{|I|}{T}, \tag{10}$$

and define the periodicity measure as:

$$\gamma^+_{\mathscr{X}}(T) = \max_{I\in\mathscr{I}_T} \Delta^+_{\mathscr{X}}(I,T). \tag{11}$$

In fact, with some slight modification to the proof of Lemma 2, we can show that it is a desired measure under our probabilistic model, resulting in the following theorem.

Theorem 2. *Suppose* $\mathbf{d} = \{d_t\}_{t=0}^{n-1}$ *are i.i.d. random variables in* $[0,1]$ *with nonzero mean, and a sequence* $\mathscr{X}$ *is generated according to any* $(\mathbf{p}^{T_0}, \mathbf{d})$ *for some* T_0*, then*

$$\lim_{n\to\infty} \gamma^+_{\mathscr{X}}(T) \le \lim_{n\to\infty} \gamma^+_{\mathscr{X}}(T_0), \quad \forall T \in \mathbb{Z}.$$

Proof. Define $c_i^+ = \frac{p_i^{T_0}}{\sum_{k=0}^{T_0-1} p_k^{T_0}} - \frac{1}{T_0}$, it is easy to see that the value $\lim_{n\to\infty} \gamma^+_{\mathscr{X}}(T_0)$ is achieved by $I^* = \{i \in [0, T_0-1] : c_i^+ > 0\}$. So it suffices to show that for any $T \in \mathbb{Z}$ and $I \in \mathscr{I}_T$,

$$\lim_{n\to\infty} \Delta^+_{\mathscr{X}}(I,T) \le \lim_{n\to\infty} \Delta^+_{\mathscr{X}}(I^*,T_0) = \sum_{i\in I^*} c_i^+.$$

Observe now that for any (I,T),

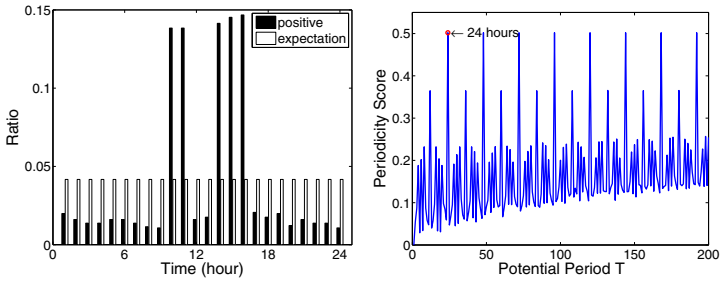
$$\lim_{n\to\infty} \mu^{+}_{\mathscr{X}}(I,T) = \sum_{i\in I}\left(\frac{1}{T}\sum_{j=0}^{T_0-1}\frac{p^{T_0}_{\mathscr{F}_{T_0}(i+j\times T)}}{\sum_{k=0}^{T_0-1}p^{T_0}_{k}}\right).$$

Therefore we have

$$\begin{aligned}
\lim_{n\to\infty}\Delta^{+}_{\mathscr{X}}(I,T) &= \frac{1}{T}\sum_{i\in I}\left\{\sum_{j=0}^{T_0-1}\left(\frac{p^{T_0}_{\mathscr{F}_{T_0}(i+j\times T)}}{\sum_{k=0}^{T_0-1}p^{T_0}_{k}}\right)-1\right\}\\
&= \frac{1}{T}\sum_{i\in I}\sum_{j=0}^{T_0-1}\left(\frac{p^{T_0}_{\mathscr{F}_{T_0}(i+j\times T)}}{\sum_{k=0}^{T_0-1}p^{T_0}_{k}}-\frac{1}{T_0}\right) = \frac{1}{T}\sum_{i\in I}\sum_{j=0}^{T_0-1}c^{+}_{\mathscr{F}_{T_0}(i+j\times T)}\\
&\le \frac{1}{T}\sum_{i\in I}\sum_{j=0}^{T_0-1}\max(c^{+}_{\mathscr{F}_{T_0}(i+j\times T)},0) \le \frac{1}{T}\sum_{j=0}^{T_0T-1}\max(c^{+}_{\mathscr{F}_{T_0}(i+j\times T)},0)\\
&= \frac{1}{T}\times T\sum_{i\in I^*}c^{+}_{i} = \sum_{i\in I^*}c^{+}_{i},
\end{aligned}$$

where the fourth equality uses the definition of I^*.

Note that this new measure $\gamma^{+}_{\mathscr{X}}(T)$ can also be applied to the cases where negative samples are available. Given the same validity result, readers may wonder if it can replace $\gamma_{\mathscr{X}}(T)$. This is certainly not the case in practice, as our results only hold exactly when the sequence has infinite length. As we will see in experiment results, negative samples indeed provide additional information for period detection in finite length observation sequences.



(a) Distribution ($T = 24$) (b) Periodicity scores

Fig. 13 (Running Example) Period detection on sequences without negative samples

Example 5 (Running Example (cont.)). In this example we further marked all the negative samples in the sequence we used in Example 4 as unknown. When there is no negative samples, the portion of positive samples at a single timestamp i is expected to be $\frac{1}{T}$, as shown in Figure 13(a). The discrepancy scores when $T = 24$ still have large values at $\{10, 11, 14, 15, 16\}$. Thus the correct period can be successfully detected as shown in Figure 13(b).

4 Algorithm: Periodo

In Section 3.2, we have introduced our periodicity measure for any potential period $T \in \mathbb{Z}$. Our period detection method simply computes the periodicity scores for every T and report the one with the highest score.

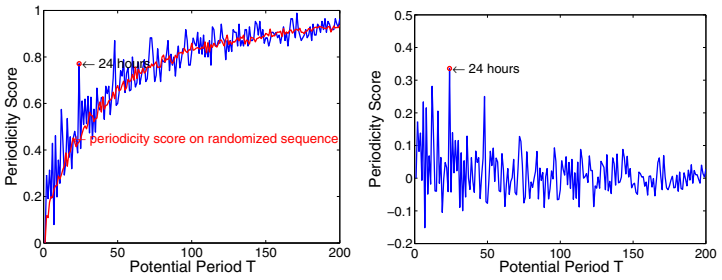
In this section, we first describe how to compute the periodicity score for a potential period and then discuss a practical issue when applying our method to finite length sequence. We will focus on the case with both positive and negative observations. The case without negative observations can be solved in the same way.

As we have seen in Section 3.2.1, the set of timestamps I^* that maximizes $\gamma_{\mathscr{X}}(T)$ can be expressed as

$$I^* = \{i \in [0, T_0 - 1] : c_i > 0\}, \tag{12}$$

where $c_i = \frac{p_i^{T_0}}{\sum_{k=0}^{T_0-1} p_k^{T_0}} - \frac{q_i^{T_0}}{\sum_{k=0}^{T_0-1} q_k^{T_0}}$. Therefore, to find I^*, it suffices to compute c_i for each $i \in [0, T_0 - 1]$ and select those ones with $c_i > 0$.

Time Complexity Analysis. For every potential period T, it takes $O(n)$ time to compute discrepancy score for a single timestamp (*i.e.*, c_i) and then $O(T)$ time to compute periodicity $\gamma_{\mathscr{X}}(T)$. Since potential period should be in range $[1, n]$, the time complexity of our method is $O(n^2)$. In practice, it is usually unnecessary to try all the potential periods. For example, we may have common sense that the periods will be no larger than certain values. So we only need to try potential periods up to n_0, where $n_0 \ll n$. This will make our method efficient in practice with time complexity as $O(n \times n_0)$.



(a) Periodicity scores (b) Normalized scores

Fig. 14 Normalization of periodicity scores

Now we want to point out a practical issue when applying our method on finite length sequence. As one may already notice in our running example, we usually see a general increasing trend of periodicity scores $\gamma_{\mathscr{X}}(T)$ and $\gamma^+_{\mathscr{X}}(T)$ for a larger potential period T. This trend becomes more dominating as the number of observations decreases. For example, the original running example has observations for 1000 days. If the observations are only for 20 days, our method may result in incorrect period detection result, as the case shown in Figure 14(a). In fact, this phenomenon is expected and can be understood in the following way. Let us take $\gamma^+_{\mathscr{X}}(T)$ as an

example. Given a sequence $\mathscr{X}$ with *finite number* of positive observations, it is easy to see that the size of I that maximizes $\gamma^{+}_{\mathscr{X}}(T)$ for any T is bounded above by the number of positive observations. Therefore the value $\frac{|I^*|}{T}$ always decreases as T increases, no matter whether or not T is a true period of $\mathscr{X}$.

To remedy this issue for finite length sequence, we use periodicity scores on *randomized* sequence to normalize the original periodicity scores. Specifically, we randomly permute the positions of observations along the timeline and compute the periodicity score for each potential period T. This procedure is repeated N times and the average periodicity scores over N trials are output as the base scores. The redline in Figure 14(a) shows the base scores generated from randomized sequences by setting $N = 10$, which agree well with the trend.

For every potential period T, we subtract the base score from the original periodicity score, resulting in the normalized periodicity score. Note that the normalized score also slightly favors shorter period, which helps us to avoid detecting duplicated periods (*i.e.*, multiples of the prime period).

4.1 Experiment Results on Synthetic Datasets

In order to test the effectiveness of our method under various scenarios, we first use synthetic datasets generated according to a set of parameter. We take the following steps to generate a synthetic test sequence SEQ.

Step 1. We first fix a period T, for example, $T = 24$. The periodic segment SEG is a boolean sequence of length T, with values -1 and 1 indicating negative and positive observations, respectively. For simplicity of presentation, we write $SEG = [s_1 : t_1, s_2 : t_2, \ldots]$ where $[s_i, t_i]$ denote the i-th interval of SEG whose entries are all set to 1.
Step 2. Periodic segment SEG is repeated for TN times to generate the complete observation sequence, denoted as standard sequence SEQ_{std}. SEQ_{std} has length $T \times TN$.
Step 3 (Random sampling η). We sample the standard sequence with sampling rate η. For any element in SEQ_{std}, we set its value to 0 (*i.e.*, unknown) with probability $(1 - \eta)$.
Step 4 (Missing segments α). For any segment in standard segment SEQ_{std}, we set all the elements in that segment as 0 (*i.e.*, unknown) with probability $(1 - \alpha)$.
Step 5 (Random noise β). For any remaining observation in SEQ_{std}, we reverse its original values (making -1 as 1 and 1 as -1) with probability β.

The input sequence SEQ has values -1, 0, and 1 indicating negative, unknown, and positive observations. In the case when negative samples are unavailable, all the -1 values will be set to 0. Note that here we set negative observations as -1 and unknown ones as 0, which is different from the description in Section 3.1. The reason is that the unknown entries are set as -1, in the presence of many missing entries, traditional methods such as Fourier transform will be dominated by missing entries instead of actual observations. The purpose of such adjustment is to facilitate traditional methods and it has no effect on our method.

4.1.1 Methods for Comparison

We compare our method with the following methods, which are frequently used to detect periods in boolean sequence [11].

1. Fourier Transform (FFT): The frequency with the highest spectral power from Fourier transform via FFT is converted into time domain and output as the result.
2. Auto-correlation and Fourier Transform (Auto): We first compute the auto-correlation of the input sequence. Since the output of auto-correlation will have peaks at all the multiples of the true period, we further apply Fourier transform to it and report the period with the highest power.
3. Histogram and Fourier Transform (Histogram): We calculate the distances between any two positive observations and build a histogram of the distances over all the pairs. Then we apply Fourier transform to the histogram and report the period with the highest power.

We will FFT(pos) and Auto(pos) to denote the methods FFT and Auto-correlation for cases without any negative observations. For Histogram, since it only considers the distances between positive observations, the results for cases with or without negative observations are exactly the same.

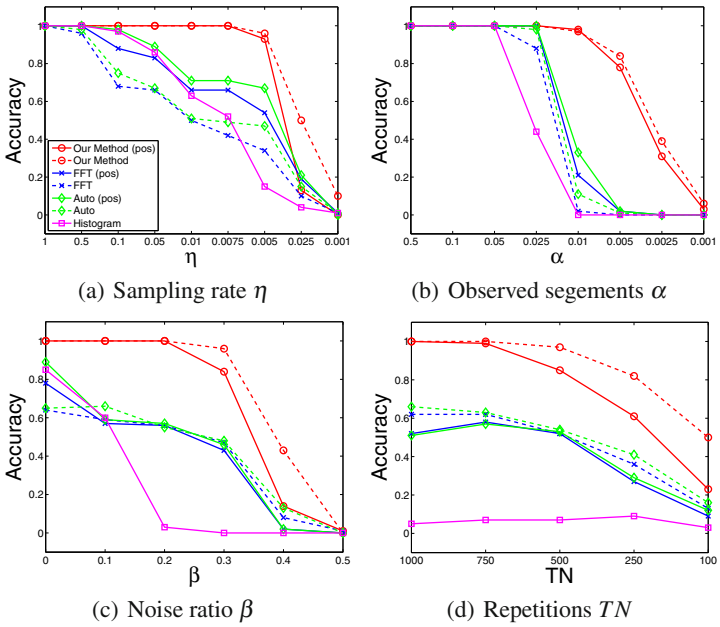
4.1.2 Performance Studies

In this section, we test all the methods on synthetic data under various settings. The default parameter setting is the following: $T = 24$, $SEG = [9 : 10, 14 : 16]$. $TN = 1000$, $\eta = 0.1$, $\alpha = 0.5$, and $\beta = 0.2$. For each experiment, we report the performance of all the methods with one of these parameters varying while the others are fixed. For each parameter setting, we repeat the experiment for 100 times and report the accuracy, which is the number of correct period detections over 100 trials. Results are shown in Figure 15.

Performance w.r.t. Sampling Rate η. To better study the effect of sampling rate, we set $\alpha = 1$ in this experiment. Figure 15(a) shows that our method is significantly better than other methods in terms of handling data with low sampling rate. The accuracy of our method remains 100% even when the sampling rate is as low as 0.0075. The accuracies of other methods start to decrease when sampling rate is lower than 0.5. Also note that Auto is slightly better than FFT because auto-correlation essentially generates a smoothed version of the categorical data for Fourier transform. In addition, it is interesting to see that FFT and Auto performs better in the case without negative observations.

Performance w.r.t. Ratio of Observed Segments α. In this set of experiments, sampling rate η is set as 1 to better study the effect of α. Figure 15(b) depicts the performance of the methods. Our method again performs much better than other methods. Our method is almost perfect even when $\alpha = 0.025$. And when all other methods fail at $\alpha = 0.005$, our method still achieves 80% accuracy.

Performance w.r.t. Noise Ratio β. In Figure 15(c), we show the performance of the methods w.r.t. different noise ratios. Histogram is very sensitive to random noises



(a) Sampling rate η (b) Observed segements α

(c) Noise ratio β (d) Repetitions TN

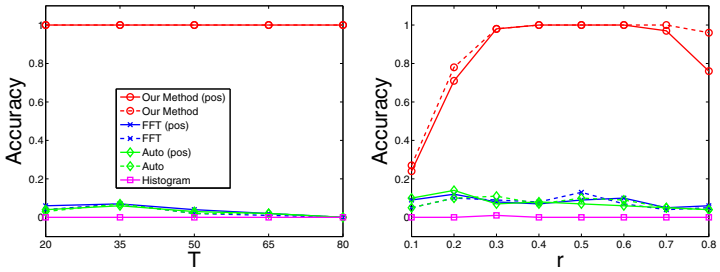
Fig. 15 Comparison results on synthetic data with various parameter settings

since it considers the distances between any two positive observations. Our method is still the most robust one among all. For example, with $\beta = 0.3$, our method achieves accuracy as high as 80%.

Performance w.r.t. Number of Repetitions TN**.** Figure 15(d) shows the accuracies as a function of TN. As expected, the accuracies decrease as TN becomes smaller for all the methods, but our method again significantly outperforms the other ones.

Performance w.r.t. Periodic Behavior. We also study the performance of all the methods on randomly generated periodic behaviors. Given a period T and fix the ratio of 1's in a SEG as r, we generate SEG by setting each element to 1 with probability r. Sequences generated in this way will have positive observations scattered within a period, which will cause big problems for all the methods using Fourier transform, as evidenced in Figure 16. *This is because Fourier transform is very likely to have high spectral power at short periods if the input values alternate between 1 and 0 frequently.* In Figure 16(a) we set $r = 0.4$ and show the results w.r.t. period length T. In Figure 16(b), we fix $T = 24$ and show the results with varying r. As we can see, all the other methods fail miserably when the periodic behavior is randomly generated. In addition, when the ratio of positive observations is low, *i.e.* fewer observations, it is more difficult to detect the correct period in general.

Comparison with Lomb-Scargle Method. Lomb-Scargle periodogram (Lomb) [15, 19] was introduced as a variation of Fourier transform to detect periods in *unevenly* sampled data. The method takes the timestamps with observations and their



(a) True period T (b) Periodic segment SEG

Fig. 16 Comparison results on randomly generated periodic behaviors

Table 2 Comparison with Lomb-Scargle method

Parameter	Accuracy		
	Our Method	FFT	Lomb
$\eta = 0.5$	1	0.7	0.09
$\eta = 0.1$	1	0.52	0.10
$\alpha = 0.5$	1	1	0.01
$\alpha = 0.1$	0.99	0.35	0

corresponding values as input. It does not work for the positive-sample-only case, because all the input values will be the same hence no period can be detected. The reason we do not compare with this method systematically is that the method performs poorly on the binary data and it is very slow. Here, we run it on a smaller dataset by setting $TN = 100$. We can see from Table 2 that, when $\eta = 0.5$ or $\alpha = 0.5$, our method and FFT perform well whereas the accuracy of Lomb is already approaching 0. As pointed out in [20], Lomb does not work well in bi-modal periodic signals and sinusoidal signals with non-Gaussian noises, hence not suitable for our purpose.

5 Experiments Results on Real Datasets

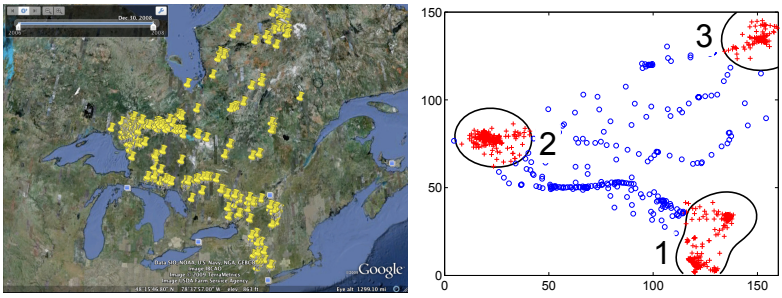
In this section, we demonstrate the effectiveness of the methods developed in this book on real-world spatio-temporal datasets. We first show the results of applying our periodic behavior mining algorithm described in Section 2 to a real dataset of bald eagle movements[2]. This experiment verifies that the proposed method is able to discover semantic meaning periodic behaviors of real animals, as long as there are enough samples within each period. Then, we use real human movement data to test the new period detection method introduced in Section 3 when the observations are highly incomplete and unevenly sampled. The experiment results suggest that

[2] The data set is obtained from `www.movebank.org`

our method is extremely robust to uncertainties, noises and missing entries of the input data obtained in real-world applications.

5.1 Mining Periodic Behaviors: A Bald Eagle Real Case

The data used in this experiment contains a 3-year tracking (2006.1∼2008.12) of a bald eagle in the North America. The data is first linearly interpolated using the sampling rate as a day.



(a) Raw data of bald eagle plotted on Google Earth.

(b) Reference spots.

Fig. 17 Real bald eagle data

Figure 17(a) shows the original data of bald eagle using Google Earth. It is an enlarged area of Northeast in America and Quebec area in Canada. As shown in Figure 17(b), three reference spots are detected in areas of New York, Great Lakes and Quebec. By applying period detection to each reference spot, we obtain the periods for each reference spot, which are 363, 363 and 364 days, respectively. The periods can be roughly explained as a year. It is a sign of yearly migration in the movement.

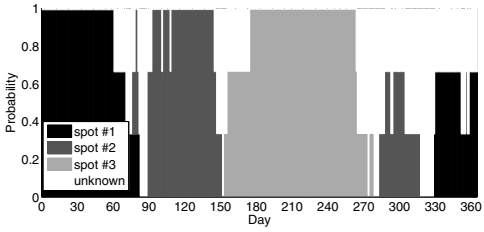


Fig. 18 Periodic behaviors of bald eagle

Now we check the periodic behaviors mined from the movement. Ideally, we want to consider three reference spots together because they all show yearly period.

However, we may discover that the periods are not exactly the same for all the reference spots. This is a very practical issue. In real cases, we can hardly get perfectly the same period for some reference spots. So, we should relax our constraint and consider the reference spots with *similar* periods together. If the difference of periods is within some tolerance threshold, we take the average of these periods and set it as the common period. Here, we take period T as 363 days, and the probability matrix is summarized in Figure 18. Using such probability matrix, we can well explain the yearly migration behavior as follows.

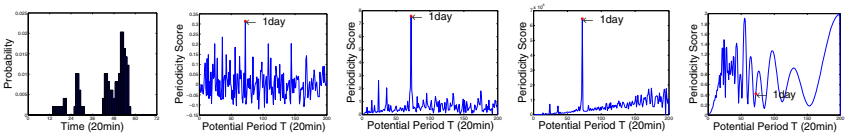
"This bald eagle stays in New York area (i.e., reference spot # 1) from December to March. In March, it flies to Great Lakes area (i.e., reference spot #2) and stays there until the end of May. It flies to Quebec area (i.e., reference spot #3) in the summer and stays there until late September. Then it flies back to Great Lake again staying there from mid October to mid November and goes back to New York in December."

This real example shows the periodic behaviors mined from the movement provides an insightful explanation for the movement data.

5.2 Mining Periodicity from Incomplete Observations: Real Human Movements

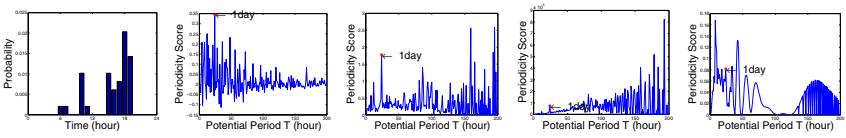
In this experiment, we use the real GPS locations of a person who has tracking record for 492 days. We first pick one of his frequently visited locations and generate a boolean observation sequence by treating all the visits to this location as positive observations and visits to other locations as negative observations. We study the performance of the methods on this symbolized movement data at different sampling rates. In Figure 19 and Figure 20, we compare the methods at two sampling rates, 20 minutes and 1 hour. As one can see in the figures (a) in Figure 19 and Figure 20, when overlaying this person's activity onto an period of one day, most of the visits occur in time interval [40, 60] for sampling rate of 20 minutes, or equivalently, in interval [15, 20] when the time unit is 1 hour. On one hand, when sampling rate is 20 minutes, all the methods except FFT(pos) and Histogram successfully detect the period of 24 hours, as they all have the strongest peaks at 24 hours (so we take 24 hours as the true period). On the other hand, when the data is sampled at each hour only, all the other methods fail to report 24 hours as the strongest peak whereas our method still succeeds. In fact, the success of our method can be easily inferred from the left-most figures in Figure 19 and Figure 20, as one can see that lowering the sampling rate has little effect on the distribution graph of the overlaid sequence. We further show the periods reported by all the methods at various sampling rates in Table 3. Our method obviously outperforms the others in terms of tolerating low sampling rates.

Next, in Figure 21, we use the symbolized sequence of the same person at a different location and demonstrate the ability of our method in detecting multiple potential periods, especially those long ones. As we can see in Figure 21(a), this person clearly has weekly periodicity w.r.t. this location. It is very likely that this



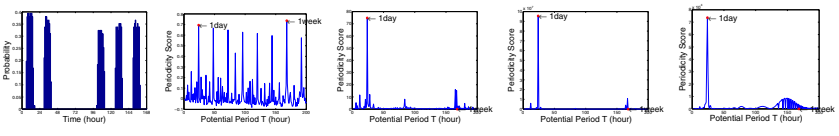
(a) Event (b) Our Method (c) FFT (d) Auto (e) Histogram

Fig. 19 [Sampling rate: 20 minutes] Comparison of period detection methods on a person's movement data



(a) Event (b) Our Method (c) FFT (d) Auto (e) Histogram

Fig. 20 [Sampling rate: 1 hour] Comparison of period detection methods on a person's movement data



(a) Event (b) Our Method (c) FFT (d) Auto (e) Histogram

Fig. 21 Comparison of methods on detecting long period, *i.e.* one week (168 hours)

Table 3 Periods reported by different methods at various sampling rates

Method	Sampling rate			
	20min	1hour	2hour	4hour
Our Method (pos)	24	24	24	8
Our Method	24	24	24	8
FFT(pos)	9.3	9	8	8
FFT	24	195	372	372
Auto(pos)	24	9	42	8
Auto	24	193	372	780
Histogram	66.33	8	42	48

location is his office which he only visits during weekdays. Our method correctly detects 7-day with the highest periodicity score and 1-day has second highest score. But all other methods are dominated by the short period of 1-day. Please note that, in the figures of other methods, 1-week point is not even on the peak. This shows the strength of our method at detecting both long and short periods.

6 Summary and Discussion

This chapter offers an overview of periodic pattern mining from spatiotemporal data. As movement data is widely available in larger volumes, the techniques of data mining nowadays play a crucial role in the semantic understanding and analysis of such data. The chapter first discusses the importance and challenges in mining periodic behaviors from movement data. We then review traditional time series methods for periodicity detection and discuss the disadvantages of directly applying these methods to movement data. To conquer these disadvantages, a novel approach, Periodica, is introduced. Periodica can detect multiple interleaved periodic behaviors from movement data by using the notion of reference spots. Next, we examine a common issue in real-world applications: the incomplete observations in spatiotemporal data. A robust period detection method for temporal events, Periodo, is then introduced to handle such sparse and incomplete movement data.

While experiment results on real movement data have already demonstrated the effectiveness of our methods, there are still many challenges that remain unsolved and new frontiers that would be interesting to explore. We list a few of them below.

First, in Periodica, there is a strong assumption that a reference spot must be a dense region on the map. However, a periodically visited place does not necessarily need to be dense in practice. For example, a person may go to Wal-Mart every Sunday afternoon. But compared with his home and office, Wal-Mart is not a densely visited location. If we use density-based method to find the reference spots, Wal-Mart is likely to be missed, even though this person has weekly periodic pattern with respect to it. Hence, designing a better method to identify such locations is a very interesting future direction.

Second, a more complicated yet more practical scenario in real data is the *irregular* periodic behavior. For example, the movement of fishing ships may follow the tides, which behave according to the cycles of the lunar phase. Hence, the movement of the ships may not have a strict monthly periodicity, which is defined based on the western calendar. Therefore, instead of simply saying "the ships roughly follow the monthly periodicity", it is desirable to develop new mechanisms which can explicitly model and detect such irregularity in the duration of a period.

Third, using periodic behaviors to predict future movements is a very important topic that deserves more in-depth study. Human and animals are highly dominated by a mixture of their routines. For example, if we observe that a person is at home at 8am, how should we predict his location at 9am based on his routines? The correct answer may be the following. If it is a weekday, the next location should be the office; if it is a weekend, the next location could still be home; however, if it is a holiday, the next location might be somewhere on the way to his hometown. As we can see, the person's behavior is not confined to a single periodic behavior, but rather determined by multiple routines and the semantics of the locations and time. Therefore, it is very important to develop principled methodology that can fuse information from various sources to make reliable predictions.

Acknowledgments. The work was supported in part by Boeing company, NASA NRA-NNH10ZDA001N, NSF IIS-0905215 and IIS-1017362, the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053 (NS-CTA) and startup funding provided by the Pennsylvania State University. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

1. Ahdesmäki, M., Lähdesmäki, H., Gracey, A., Yli-Harja, O.: Robust regression for periodicity detection in non-uniformly sampled time-course gene expression data. BMC Bioinformatics 8(1), 233 (2007)
2. Bar-Dvaid, S., Bar-David, I., Cross, P.C., Ryan, S.J., Getz, W.M.: Methods for assessing movement path recursion with application to african buffalo in south africa. Ecology 90 (2009)
3. Berberidis, C., Aref, W.G., Atallah, M.J., Vlahavas, I.P., Elmagarmid, A.K.: Multiple and partial periodicity mining in time series databases. In: Proc. 2002 European Conference on Artificial Intelligence, ECAI 2002 (2002)
4. Cao, H., Mamoulis, N., Cheung, D.W.: Discovery of periodic patterns in spatiotemporal sequences. IEEE Transactions on Knowledge and Data Engineering 19(4), 453–467 (2007)
5. Elfeky, M.G., Aref, W.G., Elmagarmid, A.K.: Periodicity detection in time series databases. IEEE Trans. Knowl. Data Eng. 17(7) (2005)
6. Elfeky, M.G., Aref, W.G., Elmagarmid, A.K.: Warp: Time warping for periodicity detection. In: Proc. 2005 Int. Conf. Data Mining, ICDM 2005 (2005)
7. Glynn, E.F., Chen, J., Mushegian, A.R.: Detecting periodic patterns in unevenly spaced gene expression time series using lomb-scargle periodograms. Bioinformatics (2005)
8. Han, J., Dong, G., Yin, Y.: Efficient mining of partial periodic patterns in time series database. In: Proc. 1999 Int. Conf. Data Engineering (ICDE 1999), Sydney, Australia, pp. 106–115 (April 1999)
9. Han, J., Gong, W., Yin, Y.: Mining segment-wise periodic patterns in time-related databases. In: Proc. 1998 Int. Conf. Knowledge Discovery and Data Mining (KDD 1998), York City, NY, pp. 214–218 (August 1998)
10. Jeung, H., Liu, Q., Shen, H.T., Zhou, X.: A hybrid prediction model for moving objects. In: Proc. 2008 Int. Conf. Data Engineering, ICDE 2008 (2008)
11. Junier, I., Herisson, J., Kepes, F.: Periodic pattern detection in sparse boolean sequences. Algorithms for Molecular Biology (2010)
12. Li, Z., Ding, B., Han, J., Kays, R., Nye, P.: Mining periodic behaviors for moving objects. In: Proc. 2010 ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD 2010), Washington D.C. (July 2010)
13. Liang, K.-C., Wang, X., Li, T.-H.: Robust discovery of periodically expressed genes using the laplace periodogram. BMC Bioinformatics 10(1), 15 (2009)
14. Liao, L., Fox, D., Kautz, H.: Location-based activity recognition using relational markov networks. In: Proc. 2005 Int. Joint Conf. on Artificial Intelligence (IJCAI 2005), pp. 773–778 (2005)
15. Lomb, N.R.: Least-squares frequency analysis of unequally spaced data. Astrophysics and Space Science (1976)
16. Ma, S., Hellerstein, J.L.: Mining partially periodic event patterns with unknown periods. In: Proc. 2001 Int. Conf. Data Engineering (ICDE 2001), Heidelberg, Germany, pp. 205–214 (April 2001)

17. Mamoulis, N., Cao, H., Kollios, G., Hadjieleftheriou, M., Tao, Y., Cheung, D.: Mining, indexing, and querying historical spatiotemporal data. In: Proc. 2004 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD 2004), Seattle, WA, pp. 236–245 (August 2004)
18. Priestley, M.B.: Spectral Analysis and Time Series. Academic Press, London (1981)
19. Scargle, J.D.: Studies in astronomical time series analysis. ii - statistical aspects of spectral analysis of unevenly spaced data. Astrophysical Journal (1982)
20. Schimmel, M.: Emphasizing difficulties in the detection of rhythms with lomb-scargle periodograms. Biological Rhythm Research (2001)
21. Vlachos, M., Yu, P.S., Castelli, V.: On periodicity detection and structural periodic similarity. In: Proc. 2005 SIAM Int. Conf. on Data Mining, SDM 2005 (2005)
22. Wang, C., Parthasarathy, S.: Summarizing itemset patterns using probabilistic models. In: Proc. 2006 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD 2006), pp. 730–735. ACM (2006)
23. Wang, W., Yang, J., Yu, P.S.: Meta-patterns: Revealing hidden periodic patterns. In: Proc. 2001 Int. Conf. Data Mining (ICDM 2001), San Jose, CA (November 2001)
24. Worton, B.J.: Kernel methods for estimating the utilization distribution in home-range studies. Ecology 70 (1989)
25. Xia, Y., Tu, Y., Atallah, M., Prabhakar, S.: Reducing data redundancy in location-based services. In: GeoSensor (2006)
26. Yan, X., Cheng, H., Han, J., Xin, D.: Summarizing itemset patterns: A profile-based approach. In: Proc. 2005 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD 2005), Chicago, IL, pp. 314–323 (August 2005)
27. Yang, J., Wang, W., Yu, P.S.: Mining asynchronous periodic patterns in time series data. In: Proc. 2000 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD 2000), Boston, MA, pp. 275–279 (August 2000)
28. Yang, J., Wang, W., Yu, P.S.: Infominer: mining surprising periodic patterns. In: Proc. 2001 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD 2001), San Francisco, CA, pp. 395–400 (August 2001)
29. Yang, J., Wang, W., Yu, P.S.: Infominer+: Mining partial periodic patterns with gap penalties. In: Proc. 2002 Int. Conf. Data Mining (ICDM 2002), Maebashi, Japan (December 2002)
30. Zhang, M., Kao, B., Cheung, D.W.-L., Yip, K.Y.: Mining periodic patterns with gap requirement from sequences. In: Proc. 2005 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD 2005), pp. 623–633 (2005)
31. Zheng, V.W., Zheng, Y., Xie, X., Yang, Q.: Collaborative location and activity recommendations with gps history data. In: Proceedings of the 19th International Conference on World Wide Web (WWW 2010), pp. 1029–1038. ACM (2010)

Spatio-temporal Data Mining for Climate Data: Advances, Challenges, and Opportunities

James H. Faghmous and Vipin Kumar

Abstract. Our planet is experiencing simultaneous changes in global population, urbanization, and climate. These changes, along with the rapid growth of climate data and increasing popularity of data mining techniques may lead to the conclusion that the time is ripe for data mining to spur major innovations in climate science. However, climate data bring forth unique challenges that are unfamiliar to the traditional data mining literature, and unless they are addressed, data mining will not have the same powerful impact that it has had on fields such as biology or e-commerce. In this chapter, we refer to spatio-temporal data mining (STDM) as a collection of methods that mine the data's spatio-temporal context to increase an algorithm's accuracy, scalability, or interpretability (relative to non-space-time aware algorithms). We highlight some of the singular characteristics and challenges STDM faces within climate data and their applications, and provide the reader with an overview of the advances in STDM and related climate applications. We also demonstrate some of the concepts introduced in the chapter's earlier sections with a real-world STDM pattern mining application to identify mesoscale ocean eddies from satellite data. The case-study provides the reader with concrete examples of challenges faced when mining climate data and how effectively analyzing the data's spatio-temporal context may improve existing methods' accuracy, interpretability, and scalability. We end the chapter with a discussion of notable opportunities for STDM research within climate.

1 Introduction

Our world is experiencing simultaneous changes in population, industrialization, and climate amongst other planetary-scale changes. These contemporaneous transformations, known as *global change*, raise pressing questions of significant

James H. Faghmous · Vipin Kumar
Department of Computer Science and Engineering, The University of Minnesota, Twin Cities
e-mail: {jfagh,kumar}@cs.umn.edu

W.W. Chu (ed.), *Data Mining and Knowledge Discovery for Big Data,*
Studies in Big Data 1,
DOI: 10.1007/978-3-642-40837-3_3, © Springer-Verlag Berlin Heidelberg 2014

scientific and societal interest [39]. For example, how will the continued growth in global population and persisting tropical deforestation, or global climate change, affect our ability to access food and water? Coincidentally, these questions are emerging at a time when data, specifically spatio-temporal climate data, are more available than ever before. In fact, climate science promises to be one of the largest sources of data for data-driven research. A recent lower bound estimate puts the size of climate data in 2010 at 10 Petabyytes (1 PB = 1,000 TB). This number is projected to grow exponentially to about 350 Patabytes by 2030 [69].

The last decades have seen tremendous growth in data-driven learning algorithms and their broad-range applications [46]. This rapid growth was fueled by the Internet's democratization of data production, access, and sharing. Merely observing these events unfold – the growth of climate data, a wide-range of challenging real-world research questions, and the emergence of data mining and machine learning in virtually every domain where data are reasonably available – one may assume that data mining is ripe to make significant contributions to these challenges.

Unfortunately, this has not been the case – at least not at the scale we have come to expect from the success of data mining in other domains, such as biology and e-commerce. At a high level, this lack of progress is due to the inherent *nature* of climate data as well as the *types* of research questions climate science attempts to address.

Although the size of climate data is a serious challenge, there are major research efforts to address the variety, velocity, and volume of climate data (commonly referred to as Big Data's 3Vs). Research efforts to address the *nature* of climate data, however, are severely lagging the rate of data growth. For instance, climate data tend to be predominantly spatio-temporal, noisy, and heterogeneous. The spatio-temporal nature of climate data emerges in the form of auto- and cross-correlation between input variables. Therefore, existing learning methods that make implicit or explicit independence assumptions about the input data will have limited applicability to the climate domain.

It is also important to study the *types* of research questions that climate science brings forth. Climate science is the study of the spatial and temporal variations of the atmospherehydrosphereland surface system over prolonged time periods. As a result, climate-related questions are inexorably linked to space and time. This means that climate scientists are interested in solutions that explain the evolution of phenomena in space and time. Furthermore, the majority of climate phenomena occur only within a specific region and time period. For example, hurricanes only take place in certain geographic regions and during a limited month range. However, due to the large datasets and the exponential number of space-time subsets within the data, we must reduce the complexity of problems by finding significant space-time subsets.

The combination of climate data's unique characteristics and associated research questions require the emergence of a new generation of space-time algorithms. Fortunately, climate data have intrinsic space and time information that, if insightfully leveraged, can provide a powerful computational framework to address many of the challenges listed above while significantly reducing the complexity of

computational problems. In this chapter we focus on the advances and opportunities for *spatio-temporal data mining*: a collection of methods that mine the data's spatio-temporal context to increase an algorithm's accuracy, scalability, or interpretability (relative to non-space-time aware algorithms). We begin by briefly reviewing the different types of climate data available and expand on notable challenges associated with them. We then proceed to a broad review of sample works in the STDM literature applied to climate spatio-temporal data. We then demonstrate the promise of STDM on a real-world application of tracking mesoscale ocean eddies in satellite data. We conclude the chapter with a review and future directions.

2 An Overview of Climate Data and Associated Challenges

In this section, we review the different types of climate data available to data mining researchers and the notable caveats when mining climate data.

2.1 Types of Climate Data

The majority of climate data available can be classified into four categories based on their source: in-situ, remote sensed, model output, and paleoclimatic.

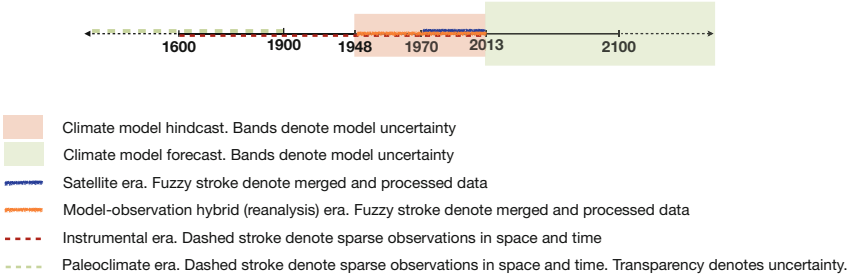


Fig. 1 Climate science has numerous types of data, each with its own challenges

In-situ records of climate data date back to the mid- to late 1600s [69]. Today, observational data are gathered from a plethora of in-situ instruments such as ships, buoys, and weather balloons. Such data tend to be sparse measurements in space and time since they are only available when measurements are gathered and where the instrument is physically located. For example, a weather balloon records frequent measurement only for a limited time duration and at its physical location. Additionally, raw measurements can be noisy due to measurement error or other phenomena temporarily impacting measurement (*e.g.* strong winds affecting temperature measurements). A final caveat is such data are dependent on the geopolitical state of where the instruments are deployed. For instance, the quality of sea surface temperatures along the Atlantic ocean decreased during World War II due to reduced reconnaissance.

Remote sensed satellite data became available in the late 1960s and are a great source of relatively high quality data for large portions of the earth. Although they are considered one of the best sources of global observational data, remote sensed satellite data have notable limitations. First, satellite data are subject to measurement noise and missing data due to obstructions from clouds or changes in orbit. Second, due to their short life-span ($\sim$ a decade) and evolving technology, satellite data can be heterogeneous.

Currently, the biggest contributors to climate data volume are climate model simulations. Climate models are used to simulate future climate change under various scenarios as well as reconstructing past climate (hindcasts). Such models run solely based on the thermodynamics and physics that govern the atmosphere-hydrosphere-land surface system, with observational data used for initialization. While these data tend to be spatio-temporally continuous, they are highly variable due to the output's dependence on parameterization and initial conditions. Furthermore, all model output come with inherent uncertainties given that not all the physics are resolved within models and our incomplete understanding of many physical processes. Therefore, the climate science community often relies on multi-model ensembles where numerous model outputs using various parameters and initial conditions are averaged to mitigate the uncertainty any single model output might have. For instance, the Nobel Peace Prize winning Intergovernmental Panel on Climate Change (IPCC) used multi-model ensembles to present its assessment of future climate change [86]. Finally, there still exist several theoretical and computational limitations that cause climate models to poorly simulate certain phenomena, such as precipitation.

To address the noisy and heterogeneous quality of in-situ and satellite observations, a new generation of simulation-observation hybrid data (or reanalyses) have emerged. Reanalysis datasets are assimilated remote and in-situ sensor measurements through a numerical climate model. Reanalyses are generated through an unchanging ("frozen") data assimilation scheme and models that take available observation from in-situ and remote sensed data every 6-12 hours over a pre-defined period being analyzed (*e.g.* 1948–2013)[1]. This unchanging framework provides a dynamically consistent estimate of the climate state at each time step. As a result, reanalysis datasets tend to be smoother than the raw observational records and have extended spatio-temporal coverage. While reanalyses are considered the best available proxy for global observations, their quality is still dependent on that of the observations, the (assimilation) model used, and processing methods. More domain specific quality issues for certain reanalysis data can be found at `http://www.ecmwf.int/research/era/do/get/index/QualityIssues`.

Finally, researchers have been reconstructing historical data using paleoclimatic proxy records such as trees, dunes, shells, oxygen isotope content and other sediments[2]. Such data are used to study climate variability at the centennial and millennial scales. Given the relatively short record of observational data, paleoclimate

[1] `http://climatedataguide.ucar.edu/reanalysis/atmospheric-reanalysis-overview-comparison-tables`

[2] `http://www.ncdc.noaa.gov/paleoclimate-data`

data are crucial for understanding pre-instrumental climate variability. It is important to note that paleoclimate data are proxies, such as using tree rings to infer rainfall or temperature trends. Furthermore, such records are used to infer climate over a wide time-span and the time of occurrence cannot be exact. Finally, paleoclimate techniques are still developing and quality testing methods continue to be an active area of research.

2.2 Unique Characteristics of Climate Data

In the introduction, we briefly mentioned some of the data's characteristics and in the previous subsection we discussed some of the issues that surround data quality and availability. In this section, we expand further on this subject to provide the reader with a more nuanced discussion of climate data characteristics.

From a modeling perspective, the most fundamental difference between traditional (categorical) data and spatio-temporal climate data is that data that are close in space and time tend to be more similar than data far apart. This "first law of geography" which is more commonly known as *autocorrelation* dictates that spatio-temporal data not be modeled as statistically independent [87]. As a result, models that assume independent and identically distributed (*i.i.d*) observations will be limited in modeling climate data and their underlying processes.

Another notable difference is that spatio-temporal phenomena in climate are not concrete "objects" but evolving patterns over space and time. For example, a hurricane doesn't simply appear and disappear, rather an atmospheric instability slowly evolves into a hurricane that gradually gains strength, plateaus, and gradually dissipates over a spatio-temporal span. This is a profound difference from traditional binary data mining where objects are either present or absent. Such spatio-temporal evolutionary processes are well captured by the differential equations used in climate models. While differential equations are costly to solve and have other well-known limitations, data mining has no (cost efficient) statistical analog to model the evolutionary nature of spatio-temporal phenomena [25]. This is becoming a significant challenge and efforts are emerging, especially within the spatio-temporal statistics community, to provide an alternative. However such methods have yet to gain wide applicability.

Another fundamental difference in climate data is the uncertainty, variability, and diversity inherent in such datasets. Uncertainty in climate data stems from the fact that many climate datasets have biases in sampling and measurement, along with some datasets being the product of merged (uncertain) data. Furthermore, researchers are seldom provided with the data's uncertainty information. For instance, there are datasets that span the past 150 years, and while it is reasonable to assume that older data are less reliable, often there is no way to objectively characterize such uncertainty. Alternatively, if one chooses to restrict their attention to the most reliable data periods (post 1979), then a data-driven research agenda becomes more challenging due to the short record.

Climate data tend to also be highly variable. Sources of variability include: (i) natural variability, where wide-range fluctuations within a single field exist between different locations on the globe, as well as at the same location across time; (ii) variability from measurement errors; (iii) variability from model parameterization; and (iv) variability from our limited understanding of how the world functions (*i.e.* model representation). Even if one accounts for such variability, it is not clear if these biases are additive and there are limited approaches to de-convolute such biases a posteriori.

We refer to data diversity as its heterogeneity in space and time. That is data are available at various spatio-temporal resolutions, from different sources, and for different uses. Often times, a researcher must rely on multiple sources of information and adequately integrating such diverse data remains a challenge. For example, one may have access to three different sea surface temperature datasets: one reanalysis dataset at a 2.5° resolution, another reanalysis dataset at 0.75° resolution, and a satellite dataset at 0.25° resolution. Given that each dataset has its own biases, it unclear what effect fusing these datasets would have on data mining tasks and knowledge extracted therein.

Additionally, climate phenomena operate and interact on multiple spatio-temporal scales. For example, changes in global atmospheric circulation patterns may have significant impacts on local infrastructures that cannot be unearthed if studying climate only at a global scale (*i.e* "will global warming cause a more rainy winter in California in year 2020?"). Understanding such multi-scale dependencies and interactions is of significant societal interest as there is a need to provide meaningful risk assessments about global climate's impact on local communities.

Finally, many climate phenomena have effects that are delayed in space and time. Although "long-range" relationships do exist in traditional data mining applications, such as a purchase occurring due to a distant acquaintance recommending a product, they are far more complex in a climate setting. Relationships in climate datasets can not only be long-range in both space and time as well as multivariate, there are exponentially many space-time-variable subsets where relationships may exist. As a result, identifying significant spatio-temporal patterns depends on knowing what to search for as much as *where* to search for such a pattern (*i.e.* which spatio-temporal resolution).

In the next section, we will provide the reader with a concise review of the STDM literature pertaining to climate data.

3 Advances in STDM Applications to Climate

Although the fields of temporal and spatial data mining research are relatively mature [77, 56], STDM is an emerging computer science field. The main driver for such emergence is the growth in spatio-temporal datasets and associated real-world challenges. Broadly speaking, STDM originated in the form of extending temporal capabilities to spatial data mining problems, or accommodating for space in temporal data mining applications. The former extension is a rather natural one given

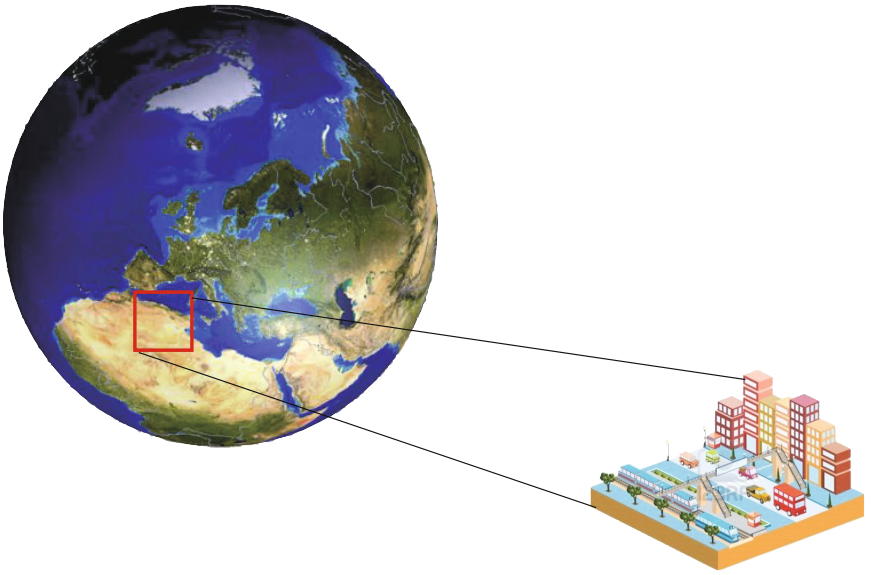


Fig. 2 A large amount of climate data is at at global spatial scale (∼250km), however many climate-related questions are at the regional (∼50km) or local (km or sum-km) scale. This multi-scale discrepancy is a significant data mining challenge.

the widespread availability of time-stamped geographic data. Intuitively, one may think of the spatio-temporal context of the data as *constraints* for a knowledge discovery algorithm. Expert constraints have been a staple of knowledge discovery algorithms as they have the potential to improve a model's scalability (by reducing the search space), accuracy (by discarding implausible models) and interpretability [22, 21, 57, 27]. In the same spirit, one may think of spatio-temporal information as expert constrains on traditional learning algorithms. However, a constraint point-of-view cannot be adopted for many existing algorithms given the strong assumptions such methods have on the nature of the data (*e.g.* i.i.d) or the data generation process (Gaussian, Poisson, *etc.*). In this case, an entire new generation of learning algorithms must be developed to account for the specific nature of *data* and *problems* STDM is trying to address. In this section, we expose the reader to a broad range of STDM application to climate. In the following subsections, we will provide a simple introduction and example for each broad type of applications as well as a sample of the literature within those applications.

3.1 Spatio-temporal Query Matching

Some of the earliest works in STDM were in the context of earth and climate sciences. Intuitively, the first step a data miner undertakes is exploring the data and its

characteristics. Given the large size of climate data, early priorities were focused on data exploration and collaborative analysis.

Mesrobian *et al.* [61] introduced CONQUEST, a parallel query processing system for exploratory research using geoscience data. The tool allowed scientists to formulate and mine queries in large datasets. This is one of the first works to track distortions in a continuous field. One application demonstrated in their work was the tracking of cyclones as local minima within a closed contour sea level pressure (SLP) field [61, 83]. As en extension to CONQUEST, Stolorz and Dean [82] introduced Quakefinder, an automatic application that detects and measures tectonic activity from remote sensing imagery. Mesrobian *et al.* [62] introduced Oasis, an extensible exploratory data mining tool for geophysical data. A similar application is the algorithm development and mining framework (ADaM) [73] which was developed to mine geophysical events in spatio-temporal data. Finally, Baldocchi *et al.* [4] introduced FLUXNET, a collaborative research tool to study the spatial and temporal variability of carbon dioxide, water vapor, and energy flux densities.

The early emphasis of all these works was on scalable query matching as well as abstracting the data and their formats to the researcher to focus more on exploratory research rather than data management. However, large-scale collaborative research efforts are costly and require extensive infrastructures and management, effectively increasing the risk associated with such endeavors. Furthermore, we often embark on exploratory research without prior knowledge of the patterns of interest making explicit query searches non-trivial. Finally, such exploratory efforts should capitalize on the recent advances in both spatial and temporal subsequence pattern mining (*e.g.* [36, 72]).

3.2 Pattern Mining

One of the fundamental applications of data mining is finding patterns within a dataset. Pattern mining refers to the insightful grouping of features that share similar characteristics such as statistical properties or frequency of occurrence. In this section we will review three notable pattern mining approaches within climate applications: empirical orthogonal function (EOF) analysis, clustering, and user-defined pattern mining.

One of the most fundamental tools in spatio-temporal pattern finding is empirical orthogonal function (EOF) analysis. EOFs are synonymous to the eigenvectors in traditional eigenvalue decomposition of a covariance matrix. As pointed out by Cressie and Wikle [25], in the discrete case, EOF analysis is simply principle component analysis (PCA). In the continuous case, it is a Karhunen-Loève (K-L) expansion. EOF analysis has been traditionally used to identify a low dimensional subspace that best explains the data's spatio-temporal variance. By taking the data's first principal component, researchers seek to identify dominant spatial structures and their evolution over time. For instance, Mestas-Nuñez and Enfield [63] analyzed the rotated[3] EOFs of global SST data and linked the first six principal components

[3] Rotation transforms the EOF into a non-orthogonal linear basis.

to ocean-atmospheric modes[4]. In another application, Basak *et al.* [6] used independent component analysis to discover the North Atlantic Oscillation index (NAO) [55] in SLP data. For a comprehensive discussion of EOF analysis for climate data please see [97].

Within clustering applications, Hoffman *et al.* [48] developed a spatio-temporal clustering algorithm to identify regions with similar environmental characteristics. White *et al.* [96] applied the techniques presented in [48] to generate climate and vegetation clusters that were subsequently used to infer phenological responses to climate change. Braverman and Fetzer [9] mined large-scale structures in climate data using a data compression technique based on entropy-constrained vector quantization [20] to generate multivariate distribution estimates of the data and monitored the changes of such distributions across space, time, and resolution. McGuire *et al.* [60] used spatial neighborhood and temporal discretization methods to identify spatio-temporal neighborhoods in SST data. In another clustering application, Gaffney *et al.* [42] clustered cyclone tracks using a regression mixture model and works by Camargo *et al.* [10] and Camargo *et al.* [11] further analyzed the clusters to discuss various properties of tracks belonging to each cluster. Although there are numerous works in the field, finding significant spatio-temporal clusters remains a major challenge because of both spatial and temporal variability. In particular, the physical meaning and significance of clusters are sometimes debatable. Furthermore, traditional feature similarity measures used to assign features to clusters, such as Euclidean distance from cluster centroids, might not have a physical meaning in climate applications.

Finally, sample works that mined climate data for user-defined patterns include: automatically identifying and tracking cyclones in the atmosphere as close contoured negative anomalies in SLP data. There are several techniques to find and monitor such patterns as storm monitoring is an active field of research. For a review please see [91]. Another dominant climate pattern is the InterTropical Convergence Zone (ITCZ), a phenomena on a daily time scale over the east Pacific. Bain *et al.* [3] developed a spatio-temporal Markov random field to detect the ITCZ in satellite data. Henke *et al.* [47] extended such methods by using a super- and semi-supervised method to track this dynamic phenomena and its properties in satellite and infrared data. Within pattern finding applications, a large number of climate phenomena tend to exist within specific spatio-temporal subsets. Naively searching for such subsets is prone to combinatorial explosion due to the exponentially-many subsets in both space and time. A notable emerging pattern mining application is that of identifying user-defined patterns in large data. Figure 3 shows an example of pattern mining in continuous spatio-temporal climate data. Ocean eddies (rotating whirlpools in the ocean) manifest in numerous climate datasets and extracting such a pattern from noisy climate data is an active field of research. In this case, the pattern of interest is localized sea surface height anomalies spanning 50 to 100s of kilometers over time-spans of weeks to months. The goal is to identify such patterns on a global scale. We will discuss this application in depth in the next section.

[4] Emanuel [33] points out that EOFs are *not* mathematically equivalent to modes.

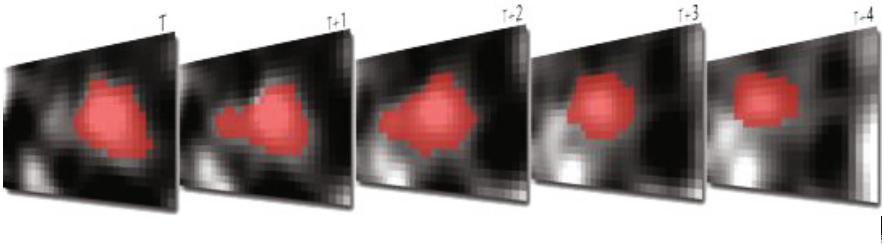


Fig. 3 An ocean eddy moving in time as detected in ocean data. One of the challenges of STDM is to identify significant patterns in continuous spatio-temporal climate data.

3.3 Event and Anomaly Detection

Automatic identification of climate events such as global changes in vegetation, droughts, and extreme rainfall is of interest to a variety of researchers. In climate applications, an event is an instance in time when a significant and persistent change occurs. In contrast, an anomaly (or outlier) is a short yet significant deviation from normal behavior. Figure 4 shows examples for an event and an anomaly. The time-series denote changes in vegetation over time as defined by remote sensed images. Panel (a) shows relatively stable vegetation from 2000 until 2003 when a distinctly new and persistent vegetation pattern emerged. Mid-2003 would be considered an event change point, where the vegetation level significantly and persistently changed from the previous period. Panel (b) shows a sudden drop in vegetation due to a forest fire in 2006. The vegetation level did recover after a few years. As a result the fire event can be considered an anomaly.

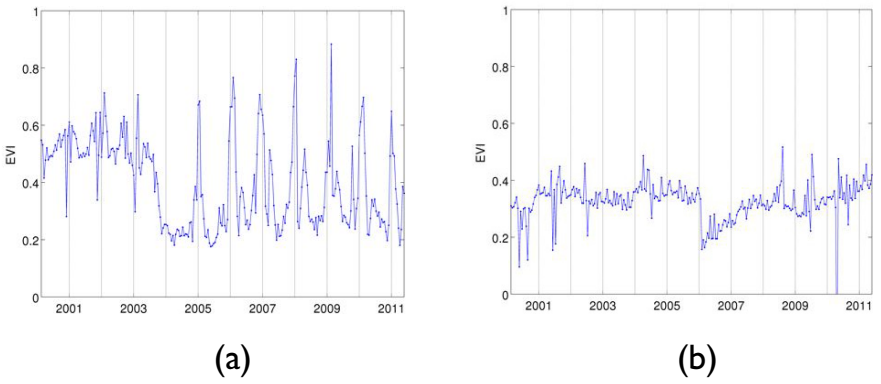


Fig. 4 An example of a spatio-temporal event (a) and anomaly (b). The time-series denote changes in vegetation over time. (a) A land-cover change event as seen in the decrease of vegetation due to agricultural expansion in 2003. (b) an abrupt drop in vegetation due to a forest fire in 2006, the vegetation gradually returned after the fire.

A number of studies have monitored event and anomaly changes in ecosystems data. Boriah *et al.* [8] proposed a recursive merging algorithm that exploited the data's seasonality to distinguish between locations that experienced a land cover change and those that did not. Mithal *et al.* [64] introduced a global land-cover change algorithm that accounted for the natural variability of vegetation levels. While the land-cover change literature is vast, especially within the remote sensing community, Mithal *et al.* [65] provide a concise discussion of STDM techniques and challenges related to land-cover change. In another global-scale event detection application, Fu *et al.* [41] extended the traditional Markov random field (MRF) model [93] used in spatial statistics by maintaining the spatio-temporal dependency structure of the MRF to autonomously detect droughts globally.

There is extensive STDM work for outlier detection for disease outbreaks [68, 67] and the climate applications base their work on that domain. To address the fact that atmospheric events occur at different scale in space and time, Cheng and Li [19] developed a multi-scale spatio-temporal outlier detection algorithm by evaluating the change between consecutive spatial and temporal scales to detect abnormal coastal changes. Barua and Alhajj [5] used a parallel wavelet transform to detect spatio-temporal outliers in SST data. Wu *et al.* [99, 100] detected spatio-temporal outliers in precipitation data by storing high discrepancy spatial regions over time in a tree. The authors were able to recover anomalous precipitation spatio-temporal spans that closely mimic the El-Niño Southern Oscillation cycle. Anbaroğlu [1] used a space-time autoregressive integrated moving average to define coherent spatio-temporal neighborhoods. An outlier was then defined if its value was significantly different from the mean that of nearby spatio-temporal neighborhoods.

Although traditional data mining has extensive research on event and outlier detection [13], there are notable differences that make such applications within climate extremely challenging. First, unlike traditional data mining where events are relatively unambiguous (*e.g.* a purchase, check in, *etc.*) the very pattern that represents an event is not known in advance or might vary based on a spatio-temporal context (*e.g.* different precipitation events could be labeled as a flood or drought depending on the time and location of occurrence). Second, climate data tends to be noisy and highly variable therefore one cannot simply label anomalous events as a large deviation from the mean. For instance, Ghosh *et al.* [43] used an extreme value theory method to highlight the fact that due to high spatial variability, anomaly detection must be in relation to space and time. Third, it is challenging to distinguish a measurement error (*i.e.* a spurious anomaly) from a low-probability event. Sugihara and May [84] proposed a method to distinguish between chaos and measurement error using short-term predictability, however additional advances might be needed. Finally, there is extreme societal interest in identifying prolonged dramatic changes in climate, known as climate state shifts [75]. Such events are critical because species tend to be less resilient to such severe abrupt changes (*e.g.* a region suddenly transforming into a desert). However, given the relatively small number of years with high quality data, it is difficult to establish with certainty whether an observed change is a significant shift or a mere fluctuation if taken into the proper spatio-temporal context. Therefore there is a need to develop novel event

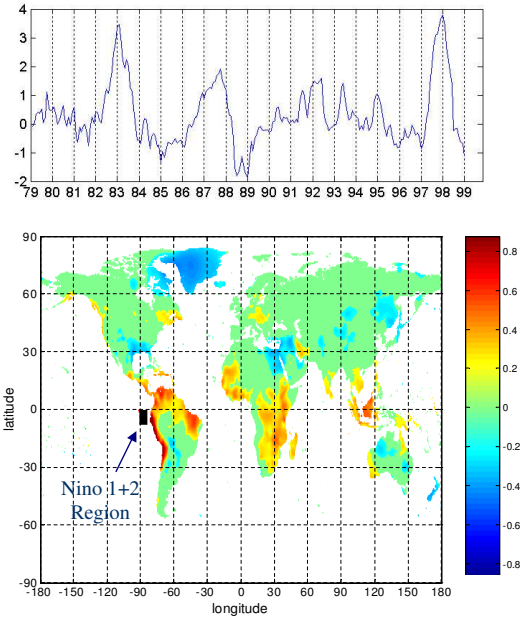


Fig. 5 Top: The NINO1+2 time-series which was constructed by averaging the sea surface temperatures (SST) of the box highlighted in the map below. Bottom: the linear correlation between the NINO1+2 index and global land surface temperature anomalies.

significance tests that would account for the limited number of reliable observations within certain datasets.

3.4 Relationship Mining

Within climate applications, researchers are interested in linking changes in one variables (e.g. global temperatures) to other phenomena (*e.g.* land cover or total number of hurricanes). A common example is relating changes in Pacific sea surface temperatures (SST), known as El-Niño Southern Oscillation (ENSO), to other global phenomena. To abstract the complex ENSO phenomenon, researchers use the mean SST of fixed regions in the Pacific to construct NINO indices and subsequently relate them to other phenomena. Figure 5 shows the linear correlation coefficients between one such NINO indices (NINO1+2) and global land surface temperature anomalies. The figure suggests that when the NINO1+2 is in a positive extreme, land temperatures tend to be high in South America, while land temperatures tend to be cooler in the south eastern United States. There are numerous works that analyze linear relationships between climate variables. Goldenberg and Shapiro [44] used linear and partial linear correlations to link vertical wind shear in the Atlantic to SST and Sahel rainfall patterns. Webster *et al.* [95] analyzed the linear correlation between basin-wide mean SST and seasonal TC counts in all the

major basins between 1970-2005 and concluded that the upward trend in Atlantic TC seasonal counts cannot be attributed to the increased SST. This was because not all basins that had an increase in SST, had a corresponding increase in TC counts. In another study, Chen *et al.* [18] used the sea surface temperatures and found different oceanic regions correlate with fire activity in different parts of Amazon. There are numerous other studies like the ones mentioned above, however detecting relationships in large climate datasets remains extremely challenging. For example, the data used in [18] only spanned 10 years (N=10). It is also impossible to isolate all confounding factors in global climate studies since many conditions can affect any given phenomenon.

One other limitation of linear correlation is its inability to capture nonlinear relationships. While there are studies that use nonlinear measures such as mutual information (*e.g.* [49]), climate scientist use *composite analysis* as a another way to quantify how well one variable explains another. Figure 6 shows an example of how composites are constructed. For a given anomaly index, in this case NINO3.4 index, we can identify extreme years as those that significantly deviate from the long-term mean (*e.g.* less/greater than one or two standard deviations). The time-series in Figure 6's upper panel highlights the extreme positive (red squares) and negative (blue squares) years within the NINO3.4 index from 1979 to 2010. Using the extreme positive and negative years, one can comment on how a variable responds to the extreme phases of a variable (in this case the NINO3.4 index). Take the June-October mean vertical wind shear over the Atlantic basin (Figure 6 bottom panel). The composite shows the difference between the mean June-October vertical wind shear during the 5 negative extreme years and the 5 positive extreme years. The bottom panel suggests that extreme negative years in NINO3.4 tend to have low vertical wind shear along the tropical Atlantic. One of the advantages of using composite analysis is that it does not make specific assumptions about the relationship between the two variables, it could be linear or non-linear. One must also use caution when analyzing composites. While we can test the significance in the difference in means between the positive and negative years, traditional significance tests assume independent observations which might not be the case for such data. Furthermore, the sample size of extreme events might be too small to be significant. For example, Kim and Han [54] constructed composites of Atlantic hurricane tracks based on the warming patterns in the Pacific ocean. One phase of their index had a sample size of 5 years (out of 39 years). To test the significance of the composite that summarized hurricane tracks during those years, the authors used a bootstrapping technique [31] to determine how significant was the mean of the small sample relative to random noise.

Finally, given that one searches for potential relationships (linear or non-linear) between a large number of observations, the likelihood of observing a strong relationship by random chance is higher than normal (known as multiple hypothesis testing or field significance). Figure 7 shows an example of the same dataset (geopotential height) correlated with a real index (left) and random noise (right). The figure shows how easily a random pattern can yield misleadingly high correlations with smooth spatial patterns.

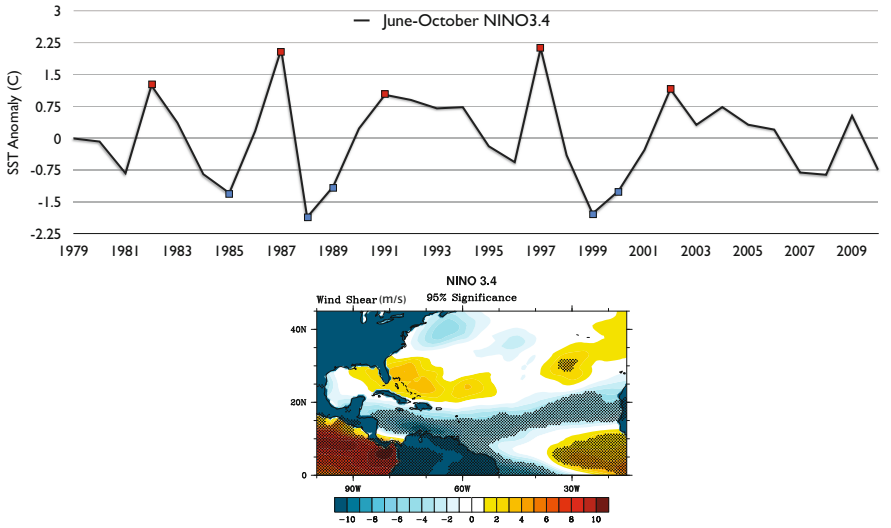


Fig. 6 An example on how composites un-earth non-linear relationships between variables. Top panel: time-series of SST anomalies in the NINO3.4 region. Bottom panel: Composite of June-October mean vertical wind shear, which was constructed by subtracting the top panel's mean of the negative extremes from the mean of the positive extremes. The figure shows that warming in the Pacific ocean has significant impact on an other variable in the tropical Atlantic.

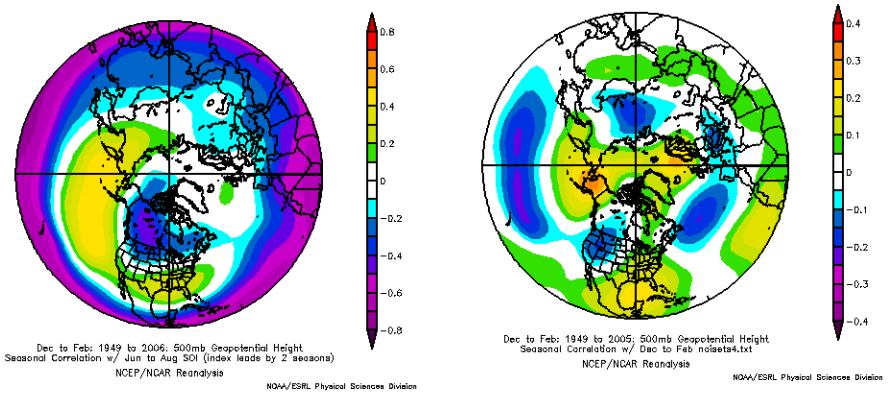


Fig. 7 Geopotential height correlated with the Southern Oscillation Index (SOI; left) and random noise (right). This is an example how high and spatially coherent correlations can be the result of random chance.

3.5 Spatio-temporal Predictive Modeling

One of the major applications to climate is the ability to model and subsequently predict future phenomena. Statistical models hold great promise to model phenomena

not well resolved in physics based models, such as precipitation. With the growth of statistical machine learning there have been numerous works on predictive modeling. In this section, we will mainly focus on some of the works that explicitly addressed the spatio-temporal nature of the data.

Coe and Stern [23] used a first- and second-order Markov chain to model precipitation. However scarce observations at the time almost certainly limit the generalization of such an approach. Cox and Isham [24] proposed a spatio-temporal model of rainfall where storm cells obey a Poisson process in space and time with each cell moving at random velocity and for a random duration. Additional reviews of precipitation models can be found in [98, 78, 79]. Huang and Cressie [50] improved on traditional spatial prediction models of water content in snow cover (also known as snow water equivalent) using a Kalman filter-based spatio-temporal model. The model effectively incorporated snow content from previous dates to make accurate snow water equivalent predictions for locations where such data was missing. Cressie *et al.* [26] designed a spatio-temporal prediction model to model precipitation over North America. Their work employed random sets to leverage data from multiple model realizations (*i.e.* multiple initial conditions, parameter settings *etc.*) of a North American regional climate model.

Van Leeuwen *et al.* [92] built a logistic regression-based model trained on land surface temperatures to detect changes in tropical forest cover. Karpatne *et al.* [51] extended the work in [92] by addressing the heterogeneous nature land cover data. Instead of training a single global model of land cover change based on a single variable (*e.g.* land surface temperature), they built multiple models based on land cover type to improve single-variable forest cover estimation models. A related application within the field of land cover change is autonomously identifying the different types of land-cover (urban, grass, corn, *etc.*) based on the pixel intensity of a remote sensed image. Traditional remote sensing techniques train a classifier to classify each pixel in an image to belong to certain land-cover class [85]. However, each pixel is classified independently of every other pixel without any regard for the spatio-temporal context. This causes highly variable class labels for the same pixel across time. Mithal *et al.* [66] improve the classification accuracy of existing models by considering the temporal evolution of the class labels of each pixel.

One of the major challenges in predictive modeling is that climate phenomena tend to have spatial and temporal lags where distant events in space and time affect seemingly unrelated phenomena far away (physically and temporally). Therefore identifying meaningful predictors in the proper spatio-temporal range is difficult. It is also important to note that certain extreme events that are of interest to the community (*e.g.* hurricanes) are so rare that the number of observations is much smaller than the data's dimensionality ($n << D$). In this case, a minimum number of predictors must be used to avoid overfitting and a poor generalized performance. For instance, Chatterjee *et al.* [14] used a sparse regularized regression method to identify the interplay between oceanic and land variables in several regions around the globe (*e.g.* how does warming in the South Atlantic affect rainfall in Brazil?). Their use of parsimony significantly improved the model's performance. Finally, model interpretability is crucial for spatio-temporal predictive modeling because the

majority of climate science applications need a physical explanation to be adopted by climate scientists.

3.6 *Network-Based Analysis*

For gridded climate data, numerous efforts have sought to abstract the large complex data and associated interactions into a simple network. Generally, nodes in the climate network are geographical locations on the grid and the edge weights measure a degree of similarity between the behavior of the time-series that characterize each node (*e.g.* linear correlation [88], mutual information [29], syntonization [2], *etc.*) Once a network is built, it is possible to apply the techniques previously discussed such as relationship mining [52], predictive modeling [81, 76], or pattern mining [80] on the transformed data.

Steinbach *et al.* [80] were one of the first to organize climate data into a network and applied a shared nearest neighbor algorithm on the network to discover the strongest climate indices: time-series that abstract the state of the atmosphere over large spatial and temporal spans. Kawale *et al.* [52] extended the work in [80] to allow for dynamic dipoles (strongly correlated distant spatial regions) in climate

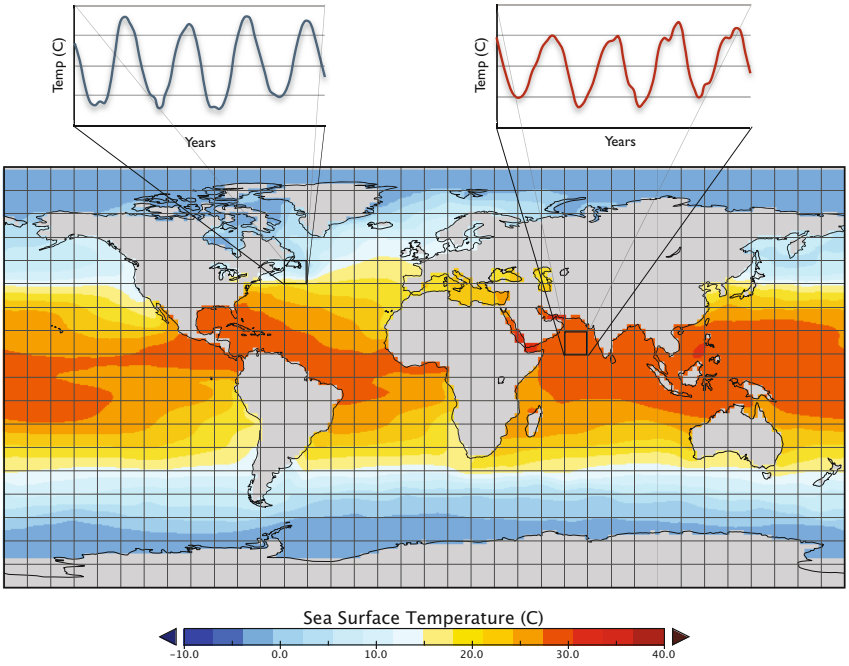


Fig. 8 Gridded spatio-temporal climate data can be analyzed in a network format. Each grid location is characterized by a time series. A network can be constructed between each location with an edge weight being the relationship between the time-series of each location.

data. Kawale *et al.* [53] proposed a bootstrapping method to test the significance of such long-range spatio-temporal patterns.

Inspired by complex networks, [88] were the first to propose the notion of a *climate network* and analyze its properties and how they relate to physical phenomena. For example, several studies have found the network structure to correlate with the dominant large-scale signals of global climate such as El-Niño [30, 102, 45]. Similarly, Tsonis *et al.* [89] showed that some climate phenomena and datasets obey a small-world network property [94]. Furthermore, several studies found distinct structural differences between the networks around tropical and extra-tropical regions [89, 29]. Berezin *et al.* [7] analyzed the evolution and stability of such networks over time and found that networks along the tropics tend to be more stable. Other studies have linked regions with high in-bound edges, known as supernodes, to be associated with major large-scale climate phenomena such as the North Atlantic Oscillation [89, 90].

Others have built networks using non-gridded discrete climate data. Elsner *et al.* [32] used seasonal hurricane time-series to construct a network to study interannual hurricane count variability. Fogarty *et al.* [38] built a network to analyze coastal locations (nodes) and their associated hurricane activity (edges) and found distinct connectivity difference between active and inactive regions. Furthermore, the authors connected various network topographies to phases of the El-Niño Southern Oscillation.

While network-based methods within climate are increasingly popular, these efforts are relatively young and several questions remain such as how to sparcify fully connected networks, the notion of multi-variate climate networks, and the distinction between statistical and physical connectivity [70].

We will spend the remainder of the chapter demonstrating a case study of spatio-temporal pattern mining with an autonomous ocean eddy monitoring application. This is because ocean eddies are a central part of ocean dynamics and impact marine and terrestrial ecosystems. Furthermore, identifying and tracking eddies form a new generation of data mining challenges where we are interested in tracking uncertain features in a continuous field.

4 STDM Application Case Study: Ocean Eddies Monitoring

In this section, we will provide an in-depth case study for mining patterns in continuous climate data, highlight some of the challenges discussed in previous sections, and provide possible ways to address them.

Very much like the atmosphere, our planet's oceans experience their own storms and internal variability. The ocean's kinetic energy is dominated by mesoscale variability: scales of tens to hundreds of kilometers over tens to hundreds of days [101, 74, 15]. Mesoscale variability is generally comprised of linear Rossby waves and as nonlinear ocean eddies (coherent rotating structures much like cyclones in the atmosphere; hereby eddies). Unlike atmospheric storms, eddies are a source of intense physical and biological activity (see Figure 9). In contrast to linear Rossby

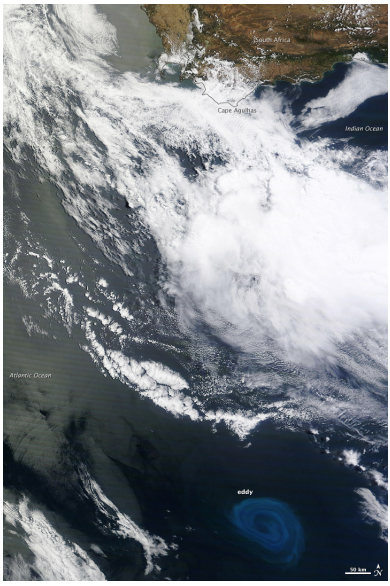


Fig. 9 Image from the NASA TERRA satellite showing an anti-cyclonic (counter-clockwise in the Southern Hemisphere) eddy that likely peeled off from the Agulhas Current, which flows along the southeastern coast of Africa and around the tip of South Africa. This eddy (roughly 200 km wide) is an example of eddies transporting warm, salty water from the Indian Ocean to the South Atlantic. We are able to see the eddy, which is submerged *under* the surface because of the enhanced phytoplankton activity (reflected in the bright blue color). This anti-cyclonic eddy would cause a depression in subsurface density surfaces in sea surface height (SSH) data. Image courtesy of the NASA Earth Observatory. Best seen in color.

waves, the rotation of nonlinear eddies transports momentum, mass, heat, nutrients, as well as salt and other seawater chemical elements, effectively impacting the ocean's circulation, large-scale water distribution, and biology. Therefore, understanding eddy variability and change over time is of critical importance for projected marine biodiversity as well as atmospheric and land phenomena.

Eddies are ubiquitous in both space and time, yet autonomously identifying them is challenging due to the fact that they are not objects moving within the environment, rather they are a distortion (rotation) evolving through a continuous field (see Figure 10). To identify and track such features, climate scientists have resorted to mining the spatial or temporal signature eddies have on a variety of ocean variables such as sea surface temperatures (SST) and ocean color. The problem is accentuated further given the lack of base-line data makes any learning algorithms unsupervised. While there exists extensive literature in traditional object tracking algorithms (*e.g.* see Yilmaz *et al.* [103] for a review), a comprehensive body of work tracking user-defined features in continuous climate data is still lacking despite the exponential increase in the volume of such data [69].

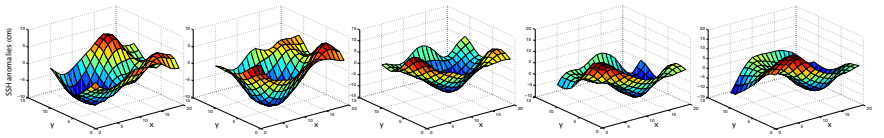


Fig. 10 An example of a cyclonic eddy traveling through a continuous sea surface height (SSH) field (from left to right). Unlike common feature mining and tracking tasks, features in physical sciences are often not self-defined with unambiguous contours and properties. Instead, they tend to be dynamic user-defined features. In the case of eddies, eddies manifest as a distortion traveling in space and time through the continuous field. A cyclonic eddy manifests as a negative SSH anomaly.

Our understanding of ocean eddy dynamics has grown significantly with the advent of satellite altimetery. Prior to then, oceanographers relied primarily on case studies using drifting floats in the open ocean to collect detailed information about individual eddies such as rotational speeds, amplitude, and salinity profiles. With the increased accessibility to satellite data, ocean surface temperatures and color have been used to identify ocean eddies based on their signatures on such fields [71, 37, 28]. While, these fields are impacted by eddy activity, there are additional phenomena, such as hurricanes or near-surface winds, that affect them as well; effectively complicating eddy identification in such data fields. More recently, sea surface height (SSH) observations from satellite radar altimeters have emerged as a better-suited alternative for studying eddy dynamics on a global scale given SSH's intimate connection to ocean eddy activity. Eddies are generally classified by their rotational direction. Cyclonic eddies rotate counter-clockwise (in the Northern Hemisphere), while anti-cyclonic eddies rotate clockwise. As a result, cyclonic eddies cause a decrease in SSH, while anti-cyclonic eddies cause an increase in SSH. Such impact allows us to identify ocean eddies in SSH satellite data, where cyclonic eddies manifest as closed contoured negative SSH anomalies and anti-cyclonic eddies as positive SSH anomalies. In Figure 11, anti-cyclonic eddies can be seen in patches of positive (dark red) SSH anomalies, while cyclonic eddies are reflected in closed contoured negative (dark blue) SSH anomalies.

In section 2.2, we discussed some general challenges that arise when mining climate data. Here we briefly review considerations one must take when specifically identifying and tracking eddies on a global scale. First, due to large-scale natural variability in global SSH data (Figure 12) complicate the task of finding a universal set of parameters to analyze the data. For example, the mean and standard of the data yield very little insight due to the high spatial and temporal natural variability. Second, unlike traditional data mining where objects are relatively well-defined, SSH data is prone to noise and uncertainty, making it difficult to distinguish between meaningful eddy patterns from spurious events and measurement errors. Third, although eddies generally have an ellipse-like shape, the shape's manifestation in gridded SSH data differs based on latitude. This is because of the stretch deformation of projecting spherical coordinates into a two-dimensional plane. As a result, one

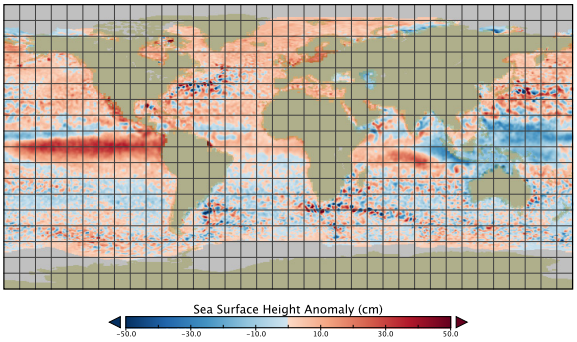


Fig. 11 Global sea surface height (SSH) anomaly for the week of October 10 1997 from the Version 3 dataset of the Archiving, Validation, and Interpretation of Satellite Oceanographic (AVISO) dataset. Eddies can be observed globally as closed contoured negative (dark blue; for cyclonic) or positive (dark red; for anti-cyclonic) anomalies. Best seen in color.

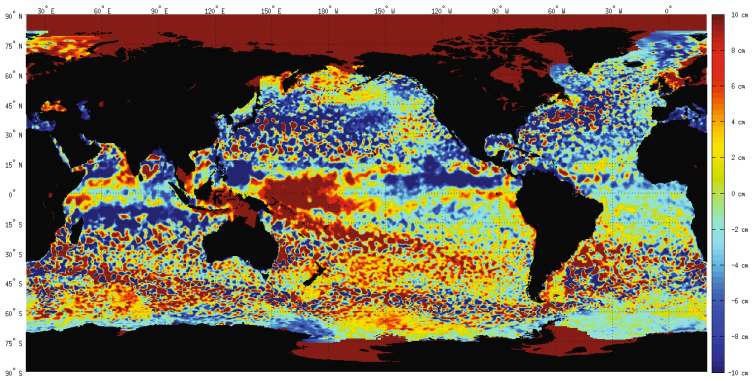


Fig. 12 Global unfiltered SSH anomalies. The data is characterized with high spatial and temporal variability, where values vary widely from one location to the next, as well as across time for the same location. Therefore traditional measures such as mean and standard deviations yield little insight in global patterns.

cannot restrict eddies by shape (*e.g.* circle, ellipse, *etc.*) Fourth, eddy heights and sizes vary by latitude, which makes having a global "acceptable" eddy size unfeasible [40]. Therefore, applying a single global threshold would wipe out many relevant patterns in the presence of spatial heterogeneity. A more subtle challenges is that eddies can manifest themselves as local minima (maxima) embedded in a large-scale background of negative (positive) anomalies [15] making numerous features unnoticeable. False positives are also an issue, as other phenomena such as linear Rossby waves or fronts can masquerade as eddy-like features in SSH data [59, 17]. Finally, given the global and ubiquitous nature of eddies, any learning must be unsupervised. One way to verify the performance of eddy identification and tracking algorithms is to use field-studies data, where floats and ships physical

sit on top of eddies. However, such datasets would only provide anecdotal evidence. Despite these non-trivial challenges, a more vexing challenge is that the majority of autonomous eddy identification schemes take the four-dimensional feature representation of eddies (latitude, longitude, time, and value where "value" depends on the field) and analyze that data orthogonally in either space or time only, effectively introducing additional uncertainty.

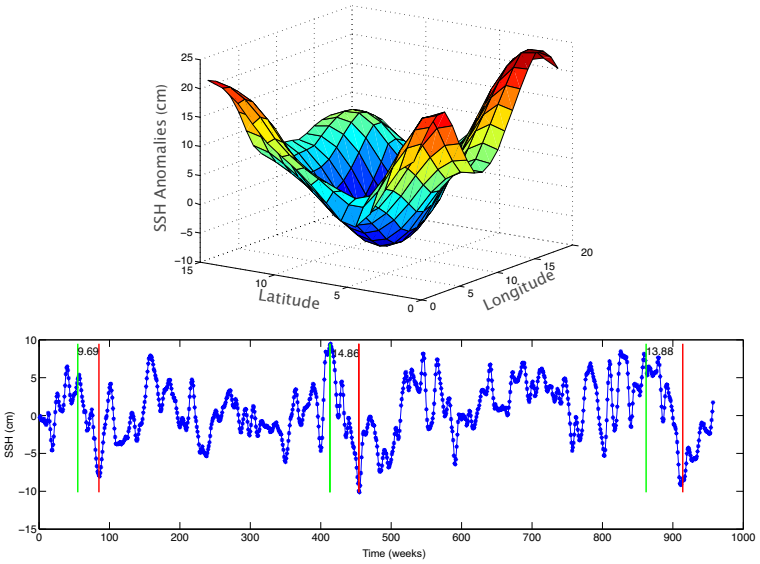


Fig. 13 Two different but complementary views of eddies' effect on SSH anomalies. *Top*: A three dimensional view of a cyclonic eddy in the SSH field. *Bottom*: an SSH time-series at single location. In both cases, the presence of an eddy is indicated through a sustained SSH depression.

Figure 13 shows two different yet complementary views of eddies and SSH. On the top panel are two anti-cyclonic eddies in the SSH field. The bottom panel shows the temporal profile of a single pixel in the SSH dataset. When taken alone each method has notable limitations. In the spatial view, thresholding the data top-down would force the application to return artificially larger size regions that the eddy occupies (since it favors the largest region possible). Furthermore, such a thresholding approach is known to merge eddies in close proximity [16]. A temporal view would allow us to identify eddy-like behavior by searching for segments of gradual decrease and increase denoted by the green and red lines [34]. However, a temporal only approach is not enough as multiple pixels must exhibit similar temporal behavior in space and time otherwise the approach would be vulnerable to noise and spurious signals. Our method attempts to combine bother approaches to address each method's limitations. We begin by discussing each approach in more detail.

4.1 Spatial Methods for Ocean Eddy Identification (Threshold-Based)

Spatial methods that identify eddies in the SSH field assign binary values to single-time SSH snapshots based on whether or not a varying threshold was exceeded, and subsequently saving the eddy-like connected component features that remain after thresholding. Subsequently the identified features are pruned based on physically-consistent criteria that define eddies. Given the noise in the SSH field, a second round of pruning occurs after tracking the features across time-frames and discarding any features that did not persist beyond four weeks. Figure 14 shows the ubiquitous cyclonic eddy features identified in a single SSH snapshot. Each snapshot contains a few thousand eddy-like features. However that number is often reduced by a variety of significance tests mentioned earlier.

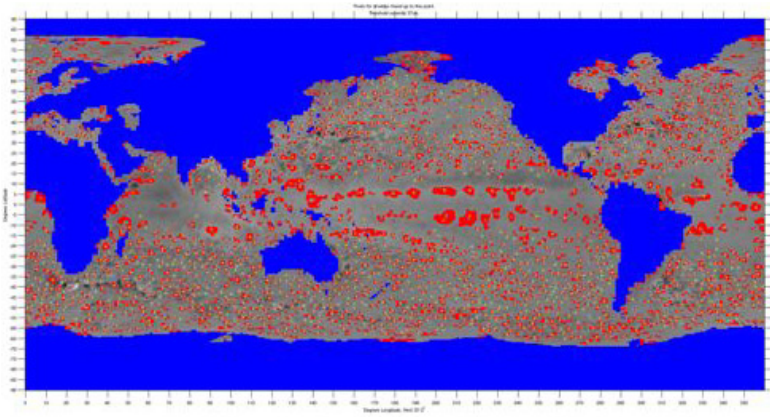


Fig. 14 Eddy-like features are ubiquitous in global SSH data. The challenges is in identifying and tracking such features within a continuous SSH field.

Chelton *et al.* [15] was the first to track eddies globally using a unified set of parameters. They also introduced the notion of eddy non-linearity (the ratio of rotational and transitional speeds) to differentiate between non-linear eddies and linear Rossby waves. In the most comprehensive SSH-based eddy tracking study to date, Chelton *et al.* [16] identified eddies globally as closed contoured smoothed SSH anomalies using a thresholding and nearest neighbor search approach. A similar algorithm was presented in [35] with a few modifications over [16] to improve the runtime complexity and accuracy of the threshold-based method.

At a high level, threshold-based algorithms extract candidate connected components from SSH data by gradually thresholding the data and finding connected component features at each threshold. For each connected component, we applied six criteria to determine if a feature is an eddy candidate: (i) A minimum eddy size of 9 pixels; (ii) a maximum eddy size of 1000 pixels; (iii) a minimum amplitude of

1 cm; (iv) the connected component must contain at least a minimum/maximum; (v) the distance between any two pixels along the contour of the feature must be less than a fixed maximum; and (vi) each connected component must have a predefined convex hull ratio as a function of the latitude of the eddy. The first five conditions are similar to those proposed by [16]. The convexity criterion is to ensure that we select the minimal set of points that can form a coherent eddy, and thus avoid mistakenly grouping multiple eddies together. Once the eddies are detected, the pixels representing the eddy are removed from consideration for the next threshold level. Doing so ensures that the algorithm does not over-count eddies. Removing the pixels will not compromise the accuracy of the algorithm given that the first instance an eddy is detected will be at its most likely largest size as a function of the threshold.

The main distinction between our implementation, *EddyScan*, and that of Chelton *et al.* [16] are two-fold: First, we use unfiltered data while Chelton *et al.* [16] pre-process the data. Second, to ensure the selection of compact rotating vortices, Chelton *et al.* [16] required that the maximum distance between any pairs of points within an eddy interior be less than a specified threshold, while EddyScan uses the convexity criterion to ensure compactness. The primary motivation to use convexity is to reduce the run time complexity of the algorithm from $O(N^2)$ to $O(N)$ in the number of features identified.

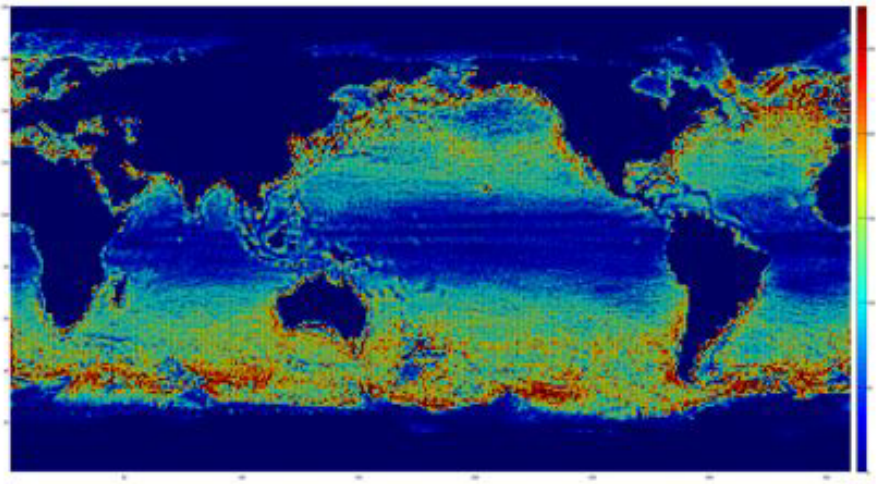


Fig. 15 Aggregate counts for eddy centroids that were observed through each $1^\circ \times 1^\circ$ region over the October 1992 - January 2011 period as detected EddyScan. These results show high eddy activity along the major currents such as the Gulf Stream (North Atlantic) and Kuroshio Current (North Pacific). Best seen in color.

There are instances, however, when the maximum distance criterion is unable to avoid merging several smaller eddies together. Figure 16 shows an example where the minimal distance between any pair of pixels in the blob is met despite there being several eddies. As a result CH11 (yellow cross) labels the entire feature as a

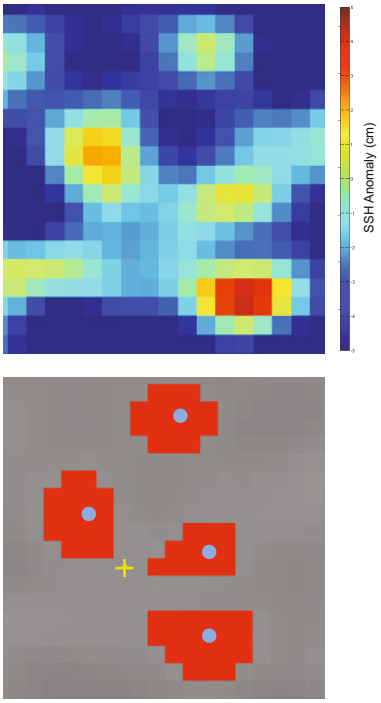


Fig. 16 An example of when Chelton *et al.* [16] maximum distance criterion is met, yet the large feature is in fact several eddies merged together. **Top:** a zoomed-in view on SSH anomalies in the Southern Hemisphere showing at least four coherent structures with positive SSH anomalies. **Bottom:** Chelton *et al.* [16] (yellow cross) identifies a single eddy in the region, while our convexity parameter allows EddyScan to successfully break the larger blob into four smaller eddies. The SSH data are in grayscale to improve visibility of the identified eddies. Best seen in color.

single eddy. EddyScan, however, is able to break the large blob into coherent small eddies.

4.2 Temporal Method for Ocean Eddy Identification

Spatial-based eddy identification schemes often have computational and application-specific limitations. Such algorithms are highly parameterized and rely on complex data-filtering schemes that make reproducibility challenging. More importantly, they fail to capitalize on a critical fact: eddies manifest as coherent SSH distortions in both space and time. When an eddy travels through the SSH field, it leaves a distinctive signature in SSH anomalies in space and time that is wasted when applying a single time-step thresholding method since all features are evaluated in the binary space. Therefore, instead of tracking eddies directly in images of SSH anomalies, an

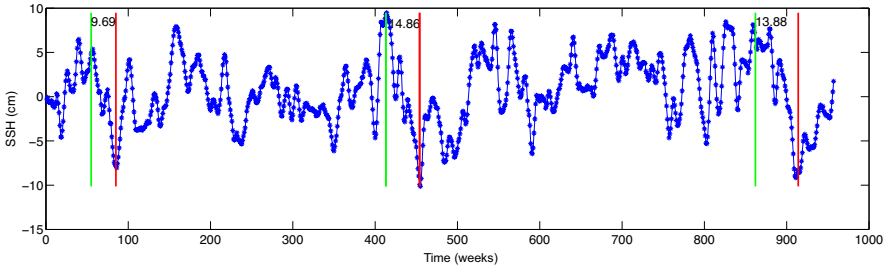


Fig. 17 A sample time-series analyzed by PDELTA with gradually decreasing segments enclosed between each pair of green and red lines. These segments were obtained after discarding segments of very short length or insignificant drop that are atypical signatures of an eddy.

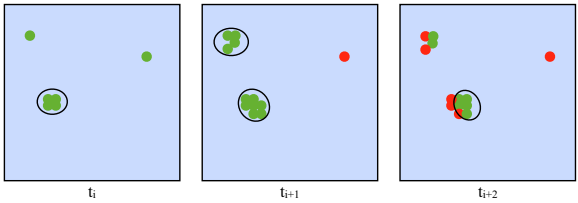


Fig. 18 An illustration to show PDELTA's spatial analysis component. At any given time t_i only a subset of all time-series are labeled as candidates for being part of an eddy (green points). Only when a sufficient number of similarly behaving neighbors are detected (in this case four) PDELTA labels them as an eddy (black circle). As time passes, some time-series are removed from the eddy (red points) as they are no longer exhibiting a gradual change; while others are added. If the number of similarly behaving time-series falls below (above) the minimum (maximum) number of required time-series, the cluster is no longer an eddy (*e.g.* top left corner at t_{i+2} frame).

alternative approach could leverage the fundamental spatio-temporal characteristics of eddies.

Eddies form and sustain their energy over a timescale of weeks to months, resulting in gradual changes in SSH on the order of a few centimeters over regions between 50-200 kilometers within the regions where the eddy move. Given the large time-scales within which eddies operate, eddies will manifest as a connected group of gradually increasing/decreasing SSH time-series. We leverage this information to track eddies directly from the SSH time-series as opposed to the SSH heat-maps.

We present an algorithm (adapted from [12]) that monitors the SSH time-series for the unique temporal signal eddies have on SSH. The algorithm operates in three main steps, first we identify individual time-series that have the previously described "eddy-like" behavior. Each candidate time-series will be labeled with a start and end time (t_s and t_e respectively) where a significant gradual increase/decrease occurred. Second, given that an eddy must operate over a large enough region, for each time step t we scan the neighbors of any candidate time-series (where $t_s \leq t \leq t_e$); if

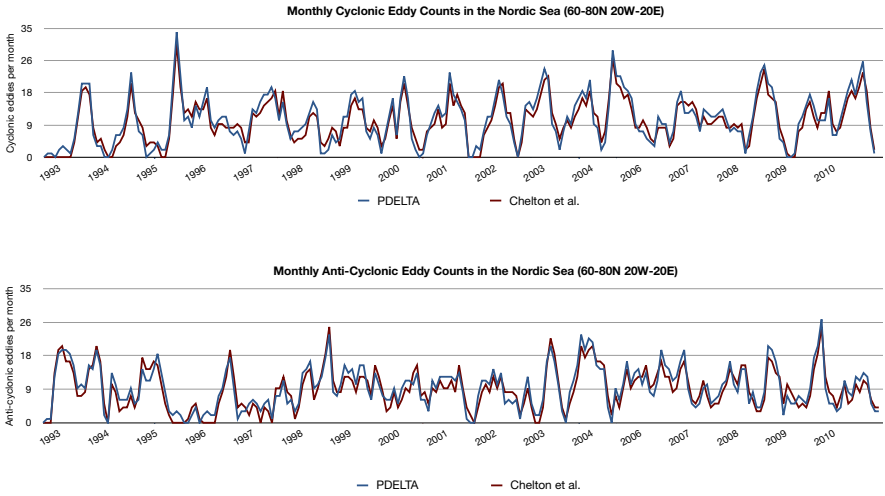


Fig. 19 Monthly eddy counts (lifetime ≥ 16 weeks). **Top:** Monthly counts for cyclonic eddies as detected by our automated algorithm PDELTA (blue) and CH11 (red). **Bottom:** Monthly counts for anti-cyclonic eddies as detected by our automated algorithm PDELTA (blue) and Chelton *et al.* [16] (red).

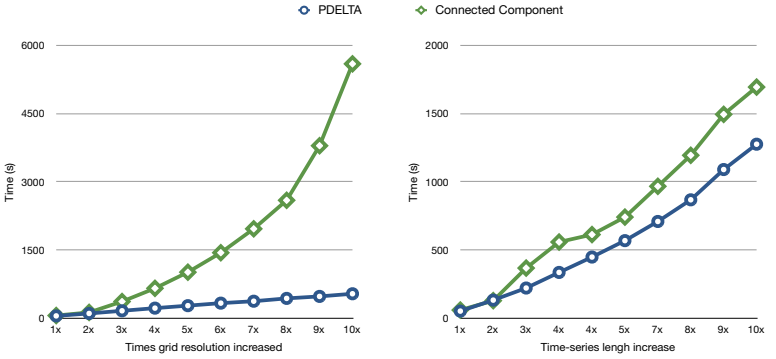


Fig. 20 Scalability comparison between our algorithm PDELTA (blue) and a connected component algorithm (green) similar to CH11. **Left:** time required to track all eddies in the dataset as a function of the grid resolution. **Right:** time required to track all eddies in the dataset as a function of the time-series length (*i.e.* number of weekly observations). Our algorithm PDELTA (blue) scales better than the connected component algorithm in both time and space.

a sufficient number of neighbors are also candidate time-series at time t then the identified group is labeled as an eddy. Finally, as the eddy moves from one time-step to the next, we keep adding new candidate time-series as their t_s is reached and remove other time-series as their t_e is passed. We count the duration of each eddy as the number of weeks the minimum number of clustered candidate time-series is met.

Figure 17 demonstrates how our approach detects candidate time-series. The top panel shows the SSH anomaly time-series for one grid point in the Nordic Sea. For this particular location, our algorithm PDELTA, identified three segments where a significant gradual decrease in SSH occurred over a long time period starting at approximatively weeks 60, 410, and 870 respectively. During each decreasing segment, we search this location's neighborhood for time-series with similar gradual decrease. Once the significant decreasing segment ends, either there will be other neighbors that will continue to form a coherent eddy or the eddy has dissipated if the minimum eddy size is no longer met.

PDELTA detected slightly more cyclonic (9.89 per month) than anti-cyclonic (9.48 per month) eddies. These differences are consistent with the findings of Chelton *et al.* [16]. Overall, we identified a total of 9.08 eddies per month versus 8.87 for Chelton *et al.* [16][5]. This could be due to the fact that eddies tend to be smaller in the region analyzed, and thus could have been ignored by CH11's algorithm once the data were filtered. Figure 19 shows the monthly cyclonic (top) and anti-cyclonic (bottom) counts for PDELTA (blue curve) and CH11 (red curve). We find that although the counts match well, PDELTA detected fewer eddies than CH11 during winter months, but more eddies during summer months.

One major advantage of considering the spatio-temporal context of the SSH data is that such an approach scales well with respect to the data's resolution and time-series length (*i.e.* number of satellite snapshots). Figure 20 shows empirical results comparing the computation time of PDELTA and the connected component algorithm as the number of grid cells ($M \times N$) and time-series length (K) are increased; the figure shows quadratic increase in computation time for the connected component algorithm as $M \times N$ is increased, while PDELTA's computation time increases linearly. This difference is particularly germane since data from future climate models and satellite observations will be of much higher resolution.

5 Conclusion and Future Directions

We presented a broad review of some of the unique characteristics of climate data along with a sample of STDM applications. We encourage interested readers to refer to the references and citations within for further reading.

Based on some of the information presented in this chapter, there may be several traditional data mining concepts that might need rethinking as we explore new applications within spatio-temporal climate data. One such re-thinking might deal with significance testing. The challenge of quantifying statistical significance in climate applications stems from both the exploratory nature of the work as well as a the autocorrelation in the data. While traditional randomization tests (e.g. [58]) may address some of the concerns stemming from multiple hypothesis testing, there is an acute need to develop spatio-temporal randomization test where the randomization procedure does not break the data's inherent characteristics such as autocorrelation.

[5] Data available at: `http://cioss.coas.oregonstate.edu/eddies/`

We might also have to re-think the definition of anomalies and extremes beyond that of abnormal deviation from the mean. Climate extremes may be better analyzed in a multi-variate fashion, where multiple relatively normal conditions may lead to a "cumulative" extreme. For instance, while hurricane Katrina was a Category 5 hurricane, it was the breaking of the levee that accentuated its horrific impact. Finally, traditional evaluation metrics for learning algorithms may need to be extended for STDM. A large number of climate problems have no reliable "ground truth" data and therefore rely on unsupervised learning techniques. Hence, it is crucial to develop objective performance measures and experiments that allow to compare the performance of different unsupervised STDM algorithms. Furthermore, traditional performance measures such root mean square error might need to be adjusted to account for spatio-temporal variability.

There are also great opportunities for novel STDM applications within climate science. Within the applications of user-defined pattern mining, the majority of features of interest are usually defined by domain experts. Such an approach is not always feasible since we have significant knowledge gaps in many domains where such data exists. Therefore developing unsupervised feature extraction techniques that autonomously identify significant features based on spatio-temporal variability (*i.e.* how different is a pattern from random noise) might be preferable, especially in large datasets. Additionally, given the large number of climate datasets, each at a different spatio-temporal resolution, there is a high demand for spatio-temporal relationship mining and predictive modeling techniques, that take data at a low, global resolution and infer impact on a higher, local resolution (and vice versa). Finally, one fundamental quantification might need to emerge between uncertainty and risk. Data mining and machine learning have used probabilities as a measure of uncertainty. However, numerous climate-related questions are interested in risk as opposed to uncertainty. Providing decision-makers with tools to convert statistical uncertainty to risk quantities based on available information is has the potential to be a major scientific and societal contribution.

Answers to some of these questions will emerge over time as we continue to see new STDM applications to climate data. Others, such as significance tests, might require diligent collaborations with adjacent fields such as statistics. Nonetheless, there is an exciting (and challenging) road ahead for STDM researchers.

Acknowledgements. Part of the research presented in this chapter was funded by an NSF Graduate Research Fellowship, an NSF Nordic Research Opportunity Fellowship, a University of Minnesota Doctoral Dissertation Fellowship, and an NSF Expeditions in Computing Grant (IIS-1029711). Access to computing resources was provided by the University of Minnesota Supercomputing Institute. The authors thank Varun Mithal for generating Figure 4 and Dr. Stefan Sobolowski for generating Figure 7. We also thank Dr. Stefan Liess for constructive comments that improved the quality of the manuscript.

References

[1] Anbaroğlu, T.C.B.: Spatio-temporal outlier detection in environmental data. Spatial and Temporal Reasoning for Ambient Intelligence Systems, 1–9 (2009)

[2] Arenas, A., Díaz-Guilera, A., Kurths, J., Moreno, Y., Zhou, C.: Synchronization in complex networks. Physics Reports 469(3), 93–153 (2008)

[3] Bain, C.L., De Paz, J., Kramer, J., Magnusdottir, G., Smyth, P., Stern, H., Wang, C.-C.: Detecting the itcz in instantaneous satellite data using spatiotemporal statistical modeling: Itcz climatology in the east pacific. Journal of Climate 24(1), 216–230 (2011)

[4] Baldocchi, D., Falge, E., Gu, L., Olson, R., Hollinger, D., Running, S., Anthoni, P., Bernhofer, C., Davis, K., Evans, R., et al.: Fluxnet: a new tool to study the temporal and spatial variability of ecosystem-scale carbon dioxide, water vapor, and energy flux densities. Bulletin of the American Meteorological Society 82(11), 2415–2434 (2001)

[5] Barua, S., Alhajj, R.: Parallel wavelet transform for spatio-temporal outlier detection in large meteorological data. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) IDEAL 2007. LNCS, vol. 4881, pp. 684–694. Springer, Heidelberg (2007)

[6] Basak, J., Sudarshan, A., Trivedi, D., Santhanam, M.: Weather data mining using independent component analysis. The Journal of Machine Learning Research 5, 239–253 (2004)

[7] Berezin, Y., Gozolchiani, A., Guez, O., Havlin, S.: Stability of climate networks with time. Scientific Reports 2 (2012)

[8] Boriah, S., Kumar, V., Steinbach, M., Potter, C., Klooster, S.: Land cover change detection: a case study. In: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 857–865. ACM (2008)

[9] Braverman, A., Fetzer, E.: Mining massive earth science data sets for large scale structure. In: Proceedings of the Earth-Sun System Technology Conference (2005)

[10] Camargo, S.J., Robertson, A.W., Gaffney, S.J., Smyth, P., Ghil, M.: Cluster analysis of typhoon tracks. part i: General properties. Journal of Climate 20(14), 3635–3653 (2007a)

[11] Camargo, S.J., Robertson, A.W., Gaffney, S.J., Smyth, P., Ghil, M.: Cluster analysis of typhoon tracks. part ii: Large-scale circulation and enso. Journal of Climate 20(14), 3654–3676 (2007b)

[12] Chamber, Y., Garg, A., Mithal, V., Brugere, I., Lau, M., Krishna, V., Boriah, S., Steinbach, M., Kumar, V., Potter, C., Klooster, S.A.: A novel time series based approach to detect gradual vegetation changes in forests. In: CIDU 2011: Proceedings of the NASA Conference on Intelligent Data Understanding, pp. 248–262 (2011)

[13] Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection for discrete sequences: A survey. IEEE Transactions on Knowledge and Data Engineering 24(5), 823–839 (2012)

[14] Chatterjee, S., Steinhaeuser, K., Banerjee, A., Chatterjee, S., Ganguly, A.: Sparse group lasso: Consistency and climate applications. In: SDM (2012)

[15] Chelton, D., Schlax, M., Samelson, R., de Szoeke, R.: Global observations of large oceanic eddies. Geophysical Research Letters 34, L15606 (2007)

[16] Chelton, D., Schlax, M., Samelson, R.: Global observations of nonlinear mesoscale eddies. Progress in Oceanography (2011a)

[17] Chelton, D.B., Gaube, P., Schlax, M.G., Early, J.J., Samelson, R.M.: The influence of nonlinear mesoscale eddies on near-surface oceanic chlorophyll. Science 334(6054), 328–332 (2011b)

[18] Chen, Y., Randerson, J.T., Morton, D.C., DeFries, R.S., Collatz, G.J., Kasibhatla, P.S., Giglio, L., Jin, Y., Marlier, M.E.: Forecasting fire season severity in south america using sea surface temperature anomalies. Science 334(6057), 787–791 (2011)
[19] Cheng, T., Li, Z.: A multiscale approach for spatio-temporal outlier detection. Transactions in GIS 10(2), 253–263 (2006)
[20] Chou, P.A., Lookabaugh, T., Gray, R.M.: Entropy-constrained vector quantization. IEEE Transactions on Acoustics, Speech and Signal Processing 37(1), 31–42 (1989)
[21] Clark, P., Matwin, S.: Using qualitative models to guide inductive learning. In: Proceedings of the Tenth International Conference on Machine Learning, pp. 49–56 (1993)
[22] Clearwater, S.H., Provost, F.J.: Rl4: A tool for knowledge-based induction. In: Proceedings of the 2nd International IEEE Conference on Tools for Artificial Intelligence, pp. 24–30. IEEE (1990)
[23] Coe, R., Stern, R.: Fitting models to daily rainfall data. Journal of Applied Meteorology 21(7), 1024–1031 (1982)
[24] Cox, D., Isham, V.: A simple spatial-temporal model of rainfall. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences 415(1849), 317–328 (1988)
[25] Cressie, N., Wikle, C.K.: Statistics for spatio-temporal data, vol. 465. Wiley (2011)
[26] Cressie, N., Assunçao, R., Holan, S.H., Levine, M., Zhang, J., Samsi, C.-N.: Dynamical random-set modeling of concentrated precipitation in north america. Statistics and its Interface (2011)
[27] Domingos, P.: Occam's two razors: The sharp and the blunt. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pp. 37–43. AAAI Press (1998)
[28] Dong, C., Nencioli, F., Liu, Y., McWilliams, J.: An automated approach to detect oceanic eddies from satellite remotely sensed sea surface temperature data. IEEE Geoscience and Remote Sensing Letters (99), 1–5 (2011)
[29] Donges, J.F., Zou, Y., Marwan, N., Kurths, J.: The backbone of the climate network. EPL (Europhysics Letters) 87(4), 48007 (2009a)
[30] Donges, J.F., Zou, Y., Marwan, N., Kurths, J.: Complex networks in climate dynamics. The European Physical Journal-Special Topics 174(1), 157–179 (2009b)
[31] Effron, B., Tibshirani, R.: Statistical data analysis in the computer age. Science 253(5018), 390–395 (1991)
[32] Elsner, J., Jagger, T., Fogarty, E.: Visibility network of united states hurricanes. Geophysical Research Letters 36(16), L16702 (2009)
[33] Emanuel, K.: The hurricane-climate connection. Bulletin of the American Meteorological Society 89(5) (2008)
[34] Faghmous, J., Chamber, Y., Vikebø, F., Boriah, S., Liess, S., dos Santos Mesquita, M., Kumar, V.: A novel and scalable spatio-temporal technique for ocean eddy monitoring. In: Twenty-Sixth Conference on Artificial Intelligence, AAAI 2012 (2012a)
[35] Faghmous, J.H., Styles, L., Mithal, V., Boriah, S., Liess, S., Vikebo, F., dos Santos Mesquita, M., Kumar, V.: Eddyscan: A physically consistent ocean eddy monitoring application. In: 2012 Conference on Intelligent Data Understanding (CIDU), pp. 96–103 (2012b)
[36] Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases, vol. 23. ACM (1994)
[37] Fernandes, A.M.: Identification of oceanic eddies in satellite images. In: Bebis, G., et al. (eds.) ISVC 2008, Part II. LNCS, vol. 5359, pp. 65–74. Springer, Heidelberg (2008)

[38] Fogarty, E.A., Elsner, J.B., Jagger, T.H., Tsonis, A.A.: Network analysis of us hurricanes. Hurricanes and Climate Change, 153–167 (2009)
[39] Foley, J.A.: Can we feed the world & sustain the planet? Scientific American 305(5), 60–65 (2011)
[40] Fu, L., Chelton, D., Le Traon, P., Morrow, R.: Eddy dynamics from satellite altimetry. Oceanography 23(4), 14–25 (2010)
[41] Fu, Q., Banerjee, A., Liess, S., Snyder, P.K.: Drought detection of the last century: An mrf-based approach. In: Proceedings of the SIAM International Conference on Data Mining (2012)
[42] Gaffney, S.J., Robertson, A.W., Smyth, P., Camargo, S.J., Ghil, M.: Probabilistic clustering of extratropical cyclones using regression mixture models. Climate Dynamics 29(4), 423–440 (2007)
[43] Ghosh, S., Das, D., Kao, S.-C., Ganguly, A.R.: Lack of uniform trends but increasing spatial variability in observed indian rainfall extremes. Nature Climate Change (2011)
[44] Goldenberg, S., Shapiro, L.: Physical mechanisms for the association of el niño and west african rainfall with atlantic major hurricane activity. Journal of Climate 9(6), 1169–1187 (1996)
[45] Guez, O., Gozolchiani, A., Berezin, Y., Brenner, S., Havlin, S.: Climate network structure evolves with north atlantic oscillation phases. EPL (Europhysics Letters) 98(3), 38006 (2012)
[46] Hastie, T., Tibshirani, R., Friedman, J., Franklin, J.: The elements of statistical learning: data mining, inference and prediction. The Mathematical Intelligencer 27(2), 83–85 (2005)
[47] Henke, D., Smyth, P., Haffke, C., Magnusdottir, G.: Automated analysis of the temporal behavior of the double intertropical convergence zone over the east pacific. Remote Sensing of Environment 123, 418–433 (2012)
[48] Hoffman, F.M., Hargrove Jr., W.W., Erickson III, D.J., Oglesby, R.J.: Using clustered climate regimes to analyze and compare predictions from fully coupled general circulation models. Earth Interactions 9(10), 1–27 (2005)
[49] Hoyos, C., Agudelo, P., Webster, P., Curry, J.: Deconvolution of the factors contributing to the increase in global hurricane intensity. Science 312(5770), 94 (2006)
[50] Huang, H.-C., Cressie, N.: Spatio-temporal prediction of snow water equivalent using the kalman filter. Computational Statistics & Data Analysis 22(2), 159–175 (1996)
[51] Karpatne, A., Blank, M., Lau, M., Boriah, S., Steinhaeuser, K., Steinbach, M., Kumar, V.: Importance of vegetation type in forest cover estimation. In: CIDU, pp. 71–78 (2012)
[52] Kawale, J., Steinbach, M., Kumar, V.: Discovering dynamic dipoles in climate data. In: SIAM International Conference on Data mining, SDM. SIAM (2011)
[53] Kawale, J., Chatterjee, S., Ormsby, D., Steinhaeuser, K., Liess, S., Kumar, V.: Testing the significance of spatio-temporal teleconnection patterns. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 642–650. ACM (2012)
[54] Kim, M., Han, J.: A particle-and-density based evolutionary clustering method for dynamic networks. Proceedings of the VLDB Endowment 2(1), 622–633 (2009)
[55] Lamb, P.J., Peppler, R.A.: North atlantic oscillation: Concept and an application. Bulletin of the American Meteorological Society 68, 1218–1225 (1987)
[56] Laxman, S., Sastry, P.S.: A survey of temporal data mining. Sadhana 31(2), 173–198 (2006)

[57] Lee, Y., Buchanan, B.G., Aronis, J.M.: Knowledge-based learning in exploratory science: Learning rules to predict rodent carcinogenicity. Machine Learning 30(2), 217–240 (1998)
[58] Livezey, R., Chen, W.: Statistical field significance and its determination by monte carlo techniques (in meteorology). Monthly Weather Review 111, 46–59 (1983)
[59] McGillicuddy Jr., D.: Eddies masquerade as planetary waves. Science 334(6054), 318–319 (2011)
[60] McGuire, M.P., Janeja, V.P., Gangopadhyay, A.: Spatiotemporal neighborhood discovery for sensor data. In: Gaber, M.M., Vatsavai, R.R., Omitaomu, O.A., Gama, J., Chawla, N.V., Ganguly, A.R. (eds.) Sensor-KDD 2008. LNCS, vol. 5840, pp. 203–225. Springer, Heidelberg (2010)
[61] Mesrobian, E., Muntz, R., Shek, E., Santos, J., Yi, J., Ng, K., Chien, S.-Y., Mechoso, C., Farrara, J., Stolorz, P., et al.: Exploratory data mining and analysis using conquest. In: Proceedings of the IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing, pp. 281–286. IEEE (1995)
[62] Mesrobian, E., Muntz, R., Shek, E., Nittel, S., La Rouche, M., Kriguer, M., Mechoso, C., Farrara, J., Stolorz, P., Nakamura, H.: Mining geophysical data for knowledge. IEEE Expert 11(5), 34–44 (1996)
[63] Mestas-Nuñez, A.M., Enfield, D.B.: Rotated global modes of non-enso sea surface temperature variability. Journal of Climate 12(9), 2734–2746 (1999)
[64] Mithal, V., Garg, A., Brugere, I., Boriah, S., Kumar, V., Steinbach, M., Potter, C., Klooster, S.: Incorporating natural variation into time-series based land cover change identification. In: Proceeding of the 2011 NASA Conference on Intelligent Data Understanding, CIDU (2011a)
[65] Mithal, V., Garg, A., Boriah, S., Steinbach, M., Kumar, V., Potter, C., Klooster, S., Castilla-Rubio, J.C.: Monitoring global forest cover using data mining. ACM Transactions on Intelligent Systems and Technology (TIST) 2(4), 36 (2011b)
[66] Mithal, V., Khandelwal, A., Boriah, S., Steinhauser, K., Kumar, V.: Change detection from temporal sequences of class labels: Application to land cover change mapping. In: SIAM International Conference on Data mining, SDM. SIAM (2013)
[67] Neill, D., Moore, A., Cooper, G.: A bayesian spatial scan statistic. In: Advances in Neural Information Processing Systems 18, p. 1003 (2006)
[68] Neill, D.B., Moore, A.W., Sabhnani, M., Daniel, K.: Detection of emerging space-time clusters. In: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 218–227. ACM (2005)
[69] Overpeck, J., Meehl, G., Bony, S., Easterling, D.: Climate data challenges in the 21st century. Science 331(6018), 700 (2011)
[70] Paluš, M., Hartman, D., Hlinka, J., Vejmelka, M.: Discerning connectivity from dynamics in climate networks. Nonlinear Processes Geophys. 18 (2011)
[71] Pegau, W., Boss, E., Martínez, A.: Ocean color observations of eddies during the summer in the gulf of california. Geophysical Research Letters 29(9), 1295 (2002)
[72] Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., Keogh, E.: Searching and mining trillions of time series subsequences under dynamic time warping. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 262–270. ACM (2012)
[73] Ramachandran, R., Rushing, J., Conover, H., Graves, S., Keiser, K.: Flexible framework for mining meteorological data. In: Proceedings of the 19th Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology (2003)

[74] Richardson, P.: Eddy kinetic energy in the north atlantic from surface drifters. Journal of Geophysical Research 88(C7), 4355–4367 (1983)
[75] Scheffer, M., Carpenter, S., Foley, J.A., Folke, C., Walker, B., et al.: Catastrophic shifts in ecosystems. Nature 413(6856), 591–596 (2001)
[76] Sencan, H., Chen, Z., Hendrix, W., Pansombut, T., Semazzi, F.H.M., Choudhary, A.N., Kumar, V., Melechko, A.V., Samatova, N.F.: Classification of emerging extreme event tracks in multivariate spatio-temporal physical systems using dynamic network structures: Application to hurricane track prediction. In: IJCAI, pp. 1478–1484 (2011)
[77] Shekhar, S., Vatsavai, R.R., Celik, M.: Spatial and spatiotemporal data mining: Recent advances. Data Mining: Next Generation Challenges and Future Directions (2008)
[78] Smith, R., Robinson, P.: A bayesian approach to the modeling of spatial-temporal precipitation data. In: Case Studies in Bayesian Statistics, pp. 237–269. Springer (1997)
[79] Srikanthan, R., McMahon, T., et al.: Stochastic generation of annual, monthly and daily climate data: A review. Hydrology and Earth System Sciences Discussions 5(4), 653–670 (2001)
[80] Steinbach, M., Tan, P.-N., Kumar, V., Klooster, S., Potter, C.: Discovery of climate indices using clustering. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 446–455. ACM (2003)
[81] Steinhaeuser, K., Chawla, N.V., Ganguly, A.R.: Complex networks in climate science: progress, opportunities and challenges. In: NASA Conf. on Intelligent Data Understanding, Mountain View, CA (2010)
[82] Stolorz, P., Dean, C.: Quakefinder: A scalable data mining system for detecting earthquakes from space. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD 1996), pp. 208–213 (1996)
[83] Stolorz, P., Mesrobian, E., Muntz, R., Santos, J., Shek, E., Yi, J., Mechoso, C., Farrara, J.: Fast spatio-temporal data mining from large geophysical datasets. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining, pp. 300–305 (1995)
[84] Sugihara, G., May, R.: Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. Nature 344(19), 734–741 (1990)
[85] Taubenböck, H., Esch, T., Felbier, A., Wiesner, M., Roth, A., Dech, S.: Monitoring urbanization in mega cities from space. Remote Sensing of Environment (2011)
[86] Team, C.W.: Climate Change 2007: Synthesis Report. Contribution of Working Groups I, Ii and Iii to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change. Ipcc, Geneva, Switzerland (2007)
[87] Tobler, W.R.: A computer movie simulating urban growth in the detroit region. Economic Geography 46, 234–240 (1970)
[88] Tsonis, A., Roebber, P.: The architecture of the climate network. Physica A: Statistical Mechanics and its Applications 333, 497–504 (2004)
[89] Tsonis, A.A., Swanson, K.L., Roebber, P.J.: What do networks have to do with climate? Bulletin of the American Meteorological Society 87(5), 585–596 (2006)
[90] Tsonis, A.A., Swanson, K.L., Wang, G.: On the role of atmospheric teleconnections in climate. Journal of Climate 21(12), 2990–3001 (2008)
[91] Ulbrich, U., Leckebusch, G., Pinto, J.: Extra-tropical cyclones in the present and future climate: a review. Theoretical and Applied Climatology 96(1), 117–131 (2009)
[92] Van Leeuwen, T.T., Frank, A.J., Jin, Y., Smyth, P., Goulden, M.L., van der Werf, G.R., Randerson, J.T.: Optimal use of land surface temperature data to detect changes in tropical forest cover. Journal of Geophysical Research 116(G2), G02002 (2011)

[93] Wainwright, M.J., Jordan, M.I.: Graphical models, exponential families, and variational inference. Foundations and Trends® in Machine Learning 1(1-2), 1–305 (2008)
[94] Watts, D., Strogatz, S.: The small world problem. Collective Dynamics of Small-World Networks 393, 440–442 (1998)
[95] Webster, P.J., Holland, G.J., Curry, A., Chang, H.: Changes in tropical cyclone number, duration, and intensity in a warming environment. Science 309(5742), 1844–1846 (2005)
[96] White, M.A., Hoffman, F., Hargrove, W.W., Nemani, R.R.: A global framework for monitoring phenological responses to climate change. Geophysical Research Letters 32(4), L04705 (2005)
[97] Wilks, D.S.: Statistical methods in the atmospheric sciences. Academic press (2006)
[98] Woolhiser, D.A.: Modeling daily precipitation progress and problems. In: Walden, A., Guttorp, P. (eds.) Statistics in the Environmental and Earth Sciences. Edward Arnold, London (1992)
[99] Wu, E., Liu, W., Chawla, S.: Spatio-temporal outlier detection in precipitation data. In: Gaber, M.M., Vatsavai, R.R., Omitaomu, O.A., Gama, J., Chawla, N.V., Ganguly, A.R. (eds.) Sensor-KDD 2008. LNCS, vol. 5840, pp. 115–133. Springer, Heidelberg (2010)
[100] Wu, E., Liu, W., Chawla, S.: Spatio-temporal outlier detection in precipitation data. In: Gaber, M.M., Vatsavai, R.R., Omitaomu, O.A., Gama, J., Chawla, N.V., Ganguly, A.R. (eds.) Sensor-KDD 2008. LNCS, vol. 5840, pp. 115–133. Springer, Heidelberg (2010)
[101] Wyrtki, K., Magaard, L., Hager, J.: Eddy energy in the oceans. Journal of Geophysical Research 81(15), 2641–2646 (1976)
[102] Yamasaki, K., Gozolchiani, A., Havlin, S.: Climate networks around the globe are significantly affected by el nino. Physical Review Letters 100(22), 228501 (2008)
[103] Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys (CSUR) 38(4), 13 (2006)

Mining Discriminative Subgraph Patterns from Structural Data

Ning Jin and Wei Wang

Abstract. Many scientific applications search for patterns in complex structural information; when this structural information is represented as graphs, a powerful tool is efficiently mining discriminative subgraphs. For example, the structures of chemical compounds can be stored as graphs, and with the help of discriminative subgraphs, chemists can predict which compounds are potentially toxic; 3D protein structures can be stored as graphs, and with the help of discriminative subgraphs, pharmacologists can predict which proteins are able to bind certain ligands and which are not; program flow information can be represented as graphs and with the help of discriminative subgraphs, computer scientists can identify program bugs and predict which program flows are successful and which are not. Many research studies have been devoted to developing efficient discriminative subgraph pattern mining algorithms. Higher efficiency allows users to process larger graph datasets and higher effectiveness enables users to achieve better results in applications. In this chapter, we introduce several existing discriminative subgraph pattern mining algorithms, including LEAP, CORK, graphSig, COM, GAIA and LTS. We evaluate the algorithms with real protein and chemical structure data.

1 Introduction

1.1 Why Mining Discriminative Subgraphs?

Discriminative subgraphs are subgraphs that appear frequently in one set of graphs but infrequently in another set of graphs. Discriminative subgraphs can capture

Ning Jin
Catalog Quality Department, Amazon, Seattle, WA 98109, USA
e-mail: jinning1985@gmail.com

Wei Wang
Computer Science Department, University of California, Los Angeles, USA
e-mail: weiwang@cs.ucla.edu

W.W. Chu (ed.), *Data Mining and Knowledge Discovery for Big Data*,
Studies in Big Data 1,
DOI: 10.1007/978-3-642-40837-3_4, © Springer-Verlag Berlin Heidelberg 2014

feature substructures that are specific to a chosen set of graphs (feature substructure identification). In addition, they can be used to differentiate one set of graphs from another (graph classification). As a result, discriminative subgraphs have a wide range of applications in structured data such as 3D protein structures, chemical compound structures, program execution traces and social networks. This section lists some applications of discriminative subgraphs.

Protein Active Site Identification and Function Prediction

Proteins are biological macromolecules that perform important functions such as catalyzing chemical reactions and binding ligands. Many protein functions are performed through active sites, substructures of proteins. Active sites are of great interest to scientists in studying mechanisms of protein functions and designing protein structures with desired functions. Traditionally, active sites are identified through expensive and time-consuming experiments. Therefore, there is a strong need for computational approaches for protein active site identification [25, 4, 7, 13]. One approach is to utilize discriminative subgraphs found in protein graphs (graph representations of 3D protein structures) [11, 12].

A 3D protein structure can be represented by an undirected graph. The protein graph of a 3D protein structure may be generated by creating a node for each amino acid and connecting two nearby amino acids with an edge. Nodes can be labeled with amino acid types and edges can be labeled with distances between amino acids.

Given protein graphs and a chosen function, the protein graphs can be grouped into two sets based on whether the corresponding proteins have the function. Discriminative subgraphs are subgraphs that are frequent in protein graphs of proteins with the chosen function but infrequent in other protein graphs. Such subgraphs are very likely to be parts of or nearby active sites because they are specific to proteins with the chosen function.

In active site identification, it is already known whether a protein has a certain function or not. However, most proteins have unknown functions. Therefore, predicting whether a protein has a certain function is another interesting problem in studying proteins [2, 3]. One solution to protein function prediction is to convert the problem to a graph classification problem [6, 21]. In a graph classification problem, the input is two sets of graphs and the output is a computational model that predicts which graph set a graph belongs to. The prediction model is then used to make predictions for graphs that are not present in the input. To convert a protein function prediction problem to a graph classification problem, proteins with known functions are represented by protein graphs and the graphs are grouped into two sets based on whether they have the function. Then prediction models are generated with discriminative subgraphs being features or even building blocks of the models.

Chemical Compound Activity Prediction

In drug discovery, the search space of candidate chemical compounds is prohibitively large and it is expensive and time-consuming to perform experiments to test activity. Therefore, computational methods are strongly needed to predict chemical compound activity and screen candidate compounds [8, 19, 20]. Such computational methods usually calculate prediction models based on a set of selected molecular descriptors that help quantitatively characterize chemical compounds [18]. Some of the molecular descriptors can be derived from discriminative subgraphs frequently found in graphs of active compounds but infrequently in graphs of inactive compounds.

A chemical compound structure can be represented by an undirected graph. One way to generate a graph representation for a chemical compound structure is to create a node for each atom and connect two bonded atoms with an edge. Nodes are labeled with atom types and edges are labeled with bond types. Stereochemical information may be embedded in node and edge labels if needed.

Program Bug Localization

Program bugs are inevitable in software development and localizing program bugs is a painstaking task. Therefore there have been many studies in automated bug localization [10]. Automated bug localization usually takes two sets of program execution traces as input: one set of traces for correct executions and another set of traces for faulty executions. A program execution trace is generated by logging method invocation and statement execution. The output of bug localization is candidate locations of bugs. One way of performing automated bug localization is to represent program execution traces by graphs and find discriminative subgraphs that appear frequently in graphs of faulty executions but infrequently in graphs of correct executions.

A program execution trace can be represented by a directed graph. In a graph representation of a program execution, each node may correspond to a method, a basic block or a statement and each edge describes how the control/data flow moves from one node to another. Nodes are labeled with basic descriptions of the corresponding methods, basic blocks or statements.

Utilizing Social Network Information

A social network describes how people are related to or interact with each other. It provides structural information for each individual person in addition to numeric and categorical attributes such as age, gender and interest [17]. Such structural information is abundant and can be easily collected via online social network platforms, such as Facebook and LinkedIn. The online platforms and their collaborators can utilize such information to improve their personalized recommendations of friends, articles, products and services [5]. One way to utilize social network information for personalized recommendations is to convert the

problem to graph classification and use discriminative subgraphs found in social network graphs.

Each person can be described by the social network that he/she is directly involved in and a social network can be represented by a graph. In a graph representation of a social network, each node corresponds to one person and each edge describes whether two persons have interaction/relationship. Nodes can be labeled with basic attributes, such as age and gender, of the corresponding persons. Edges can have directions if the interactions/relationships are asymmetric.

To convert personalized recommendation to graph classification, social network graphs of people with known interests are grouped into two sets based on whether the corresponding person is interested in a certain recommendation or not. Then a prediction model is generated and can be applied to social network graphs of people with unknown interests. High accuracy of such graph prediction models can be achieved with the help of highly discriminative subgraphs.

1.2 Definitions

Definition 1 (Graph). A graph is denoted as $g = (V, E)$ where V is a set of nodes and E is a set of edges connecting the nodes. Both nodes and edges can have labels.

For example, in Figure 1, there are two graphs in the graph database. The text in each node is in the form of (node ID: node label). Two nodes in a graph may have the same label but they cannot have the same node ID. Two nodes in two different graphs can have the same node ID but they do not necessarily represent the same entity and may have different labels.

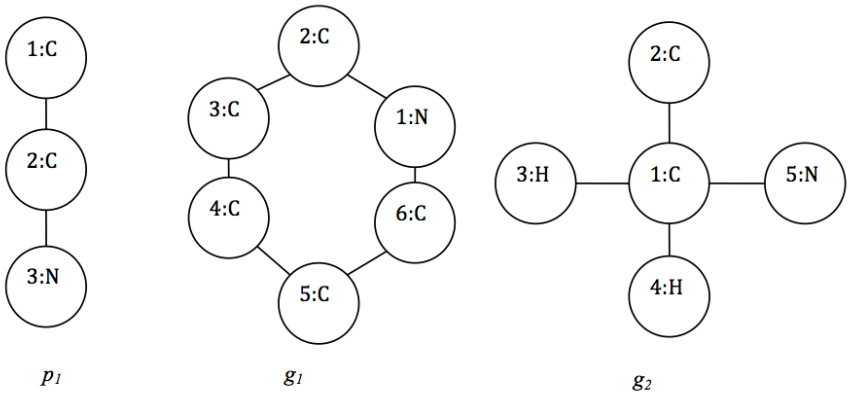


Fig. 1 An example of two graphs and a subgraph pattern

Definition 2 (Subgraph Isomorphism and Graph Isomorphism). The label of a node u is denoted by $l(u)$ and the label of an edge (u, v) is denoted by $l((u, v))$. For two graphs g and g', if there exists an injection f: $g.V \rightarrow g'.V$ such that for any node $u \in g.V$, $l(u) = l(f(u))$ and for any edge $(u, v) \in g.E$, $l((u, v)) = l((f(u), f(v)))$, then g is a subgraph of g' ($g \subseteq g'$) and g' is a supergraph of g. in other words, g' supports or contains g. If g is a subgraph of g' and g' is a subgraph of g, then g is isomorphic to g'.

For example, in Figure 1, pattern p_1 is a subgraph of g_1 and g_2.

Definition 3 (Frequency). Given a graph set G, the frequency of a subgraph pattern p is the ratio of the number of graphs supporting p in G to the total number of graphs in G.

The input of a discriminative subgraph pattern mining problem is composed of two sets of graphs: a positive set G_p and a negative set G_n.

Definition 4 (Positive Set and Negative Set). Given a certain property A, a positive set is a set of objects with property A (i.e. having positive test results for the property); the corresponding negative set is a set of objects without property A (i.e. having negative test results for the property).

The frequency of pattern p in the positive set is denoted by $pfreq(p)$ and the frequency in the negative set is denoted by $nfreq(p)$.

Definition 5 (Discrimination Score). The discrimination score of a subgraph pattern p is a user-specified objective function of its frequencies in the positive and negative sets. The more discriminative the pattern, the larger the discrimination score. For example, here is one of the scoring functions used in [24]:

$$G\text{-test score of } p = pfreq(p) * \log\frac{pfreq(p)}{nfreq(p)} + (1 - pfreq(p)) * \log\frac{1 - pfreq(p)}{1 - nfreq(p)}$$

Its rationale is that if two patterns have the same positive frequency (negative frequency) then the one with lower negative frequency (higher positive frequency) is better for discriminating positive graphs from negative graphs.

Definition 6 (Discriminative Subgraph Pattern Mining). We define discriminative subgraph pattern mining as a process to search a positive graph set and negative graph set for the subgraph pattern with the highest discrimination score for each positive graph.

1.3 Overview of Existing Mining Techniques

One straightforward solution to find discriminative subgraphs is to first enumerate all the subgraphs that are frequent in one set of graphs and then among the frequent subgraphs select those that are infrequent in the other set of graphs. This exhaustive enumeration and selection approach guarantees to find all discriminative subgraphs. However, the enumeration step typically generates an

enormous quantity of candidate subgraphs and computing frequency in the selection step involves subgraph isomorphism, which is known to be an NP-complete problem. These two limitations inhibit the ability of this straightforward solution to handle large real-world graph datasets. In addition, many subgraphs that are frequent in one set are also frequent in the other set. As a result, the straightforward solution is inefficient because much of the computation to enumerate frequent subgraphs is wasted.

To overcome the limitations, recent approaches search directly for discriminative subgraph patterns. LEAP [24] is a pioneer in discriminative subgraph pattern mining. It looks for the optimal subgraph pattern in terms of discrimination power with a branch-and-bound technique, taking advantage of the fact that structurally similar subgraphs tend to have similar discrimination power. It also uses a technique called "frequency descending mining" to exploit the correlation between subgraph frequency and subgraph discrimination power.

Another algorithm CORK [22] proposes to use correspondence to measure the discrimination power of subgraph patterns and thereby achieves a theoretically near-optimal solution. Given a set of subgraph patterns, the number of correspondences is the total number of pairs of graphs that these subgraphs cannot discriminate.

GraphSig [18] is an algorithm that utilizes frequent subgraph mining to find discriminative subgraphs but in a different way than the straightforward solution. It first converts graphs to feature vectors by performing Random Walk with Restarts on each node. Then it divides graphs into small groups such that graphs in the same group have similar vectors. It mines frequent subgraphs in each group with high frequency thresholds because high similarity in vectors in the same group indicates that the corresponding graphs in the group share highly frequent subgraphs. Using high frequency thresholds in frequent subgraph mining avoids enumerating a prohibitively large number of candidate subgraphs and enables graphSig to process relatively large datasets efficiently.

COM [14] searches for discriminative co-occurrences of small subgraph patterns instead of individually discriminative subgraph patterns. It employs a pattern exploration order such that the complementary discriminative subgraph patterns are examined first. Subgraph Patterns are grouped into co-occurrences during the pattern exploration. By taking advantage of co-occurrence information, COM can generate strong features by assembling weak features.

GAIA [15] employs a novel subgraph encoding approach to support an arbitrary subgraph pattern exploration order and explores the subgraph pattern space in a heuristic mining process resembling biological evolution. In this mining process, new candidate patterns are calculated by extending old candidate patterns and candidate patterns with lower discrimination power are more likely to be pruned by the algorithm. In this manner, GAIA is able to find discriminative subgraph patterns much faster than other algorithms. Additionally, it takes advantage of parallel computing to further improve the efficiency of the mining process.

LTS [16] is based on an observation that search history of discriminative subgraph mining is very useful in computing empirical upper bounds of discrimination power of subgraphs. LTS begins with a greedy algorithm that first samples the search space and then calculates a model to estimate upper bounds of discrimination power of subgraphs based on the samples. In the end, LTS explores the search space again in a branch and bound fashion leveraging the upper bound estimation model.

2 Mining Techniques

2.1 Simplifying the Problem

The main idea of algorithms in this category is to simplify the mining problem.

2.1.1 Dividing the Input Dataset (graphSig)

Feature Vector Generation

GraphSig predefines a set of simple structural features such as nodes and edges with specific labels. It represents each node in each graph with a feature vector based on the predefined features. As a result, a graph with n nodes is represented with n feature vectors. The feature vector for a node reflects the distribution of features around the node.

The algorithm begins with generating feature vectors. A feature vector is generated by performing RWR (Random Walk with Restarts) on a node in each graph. RWR simulates the trajectory of a walker that begins from the starting node and moves from one node to a randomly selected neighbor. Each neighbor has the same probability of being selected for next move. In addition, graphSig limits the distance of random walk by having a restart probability to bring the walker back to the starting node. Each feature value is the probability of it being traversed in RWR. Therefore, a high feature value means the feature is close to the starting node. When the feature values converge, RWR terminates.

Significant Sub-feature Vectors

Given two feature vectors $x = <x_1, x_2, ..., x_m>$ and $y = <y_1, y_2, ..., y_m>$, graphSig defines that x is called a sub-feature vector of y if and only if $x_i \leq y_i$, for i = 1, ..., m.

The probability of a feature vector x occurring in a random feature vector y is calculated as follows:

$$P(x) = \prod_{i=1}^{m} P(y_i \geq x_i)$$

Once the probability of feature vector x occurring in a random feature vector is known, the p-value of feature vector x can be calculated. The smaller the p-value of feature vector x, the more statistically significant the vector is.

GraphSig searches all feature vectors generated by RWR for common statistically significant sub-feature vectors. A statistically significant sub-feature vector indicates potential existence of discriminative subgraph patterns around the corresponding node. Therefore, for each significant sub-feature vector, graphSig invokes frequent subgraph pattern mining to search the supporting graphs for highly frequent subgraph patterns around the node associated with the sub-feature vector. This frequent subgraph mining process is highly efficient because the number of supporting graphs is very small and the search is limited to the neighborhood around the node associated with the sub-feature vector. In the end, discriminative subgraph patterns can be selected from the frequent subgraph patterns.

Overall Framework

The overall framework of graphSig is as follows:

Step 1: Calculate feature vectors for all graphs with Random Walk with Restarts,
Step 2: Use feature vector mining to find significant and frequent sub-feature vectors with user-specified frequency and p-value thresholds,
Step 3: Use frequent subgraph pattern mining to search graphs that share significant sub-feature vectors for discriminative subgraph patterns.

The algorithm is described as below.

Algorithm: *graphSig*
Input:
G_p: a set of positive graphs
G_n: a set of negative graphs
minFreq: frequency threshold for frequent subgraph mining
maxPvalue: p-value threshold for selecting significant sub-feature vectors
radius: radius for extracting subgraphs around a given node
Output:
P: a set of discriminative subgraph patterns

1. $P = \emptyset$;
2. $F = \emptyset$;
3. **for each** $g \in G_p \cup G_n$
4. R = {feature vectors generated by RWR in g with p-values < maxPvalue};
5. $F = F \cup R$;
6. $S = S - \{p\}$;
7. **for each** node label a in $G_p \cup G_n$
8. $F_a = \{f \mid f \in F, f$ was generated by RWR on nodes whose labels were $a\}$;

```
9.  S = {significant sub-feature vectors in F_a};
10. for each f ∈ S
11.     V_f = {v | ∃ g ∈ G_p ∪ G_n, v ∈ g.V, l(v) equals a and f is a sub-feature
        vector of the feature vector of v};
12.     C = ∅;
13.     for each v ∈ V_f
14.         C = C ∪ {g' | g' is a subgraph centered at v within distance radius};
15.     P = P ∪ frequent_subgraph_mining(C, minFreq);
16. return P;
```

2.1.2 Choosing Discrimination Power Measurement (CORK)

A Submodular Discrimination Score Function

The goal of CORK is to find a subgraph pattern set that can discriminate two sets of graphs instead of individually discriminative subgraph patterns. Therefore, the discrimination score function used by CORK evaluates the discrimination power of a set of subgraph patterns rather than individual patterns.

CORK uses a greedy algorithm to search for the target pattern set. It begins with an empty pattern set and gradually adds one subgraph pattern to the set at a time. Each time it adds a subgraph pattern to the pattern set, CORK chooses the subgraph pattern that can maximize the discrimination score function of the new pattern set. In general, this greedy algorithm does not guarantee the optimal solution. However, it can guarantee a near-optimal solution if the discrimination score function is submodular. Submodularity is defined as follows:

Given a search space D, a pattern $p \in D$, and two candidate pattern set T and T', $T' \subset T \subseteq D$, a scoring function *score* is submodular if:

$$score(T' \cup \{p\}) - score(T') \geq score(T \cup \{p\}) - score(T)$$

If the scoring function is submodular, it has been proved that the greedy algorithm yields a near-optimal solution and its score achieves at least $(1-\frac{1}{e}) \approx 63\%$ of the score of the optimal solution.

Therefore, CORK uses the product of -1 and the number of correspondences as its scoring function, which is submodular. Given a set of subgraph patterns, the number of correspondences is the total number of pairs of graphs that these subgraphs cannot discriminate. As a result, pattern sets with higher discrimination power have less number of correspondences and thus higher scores. For example, in Figure 2, for pattern set {A-B, B-C}, the pairs of graphs it cannot discriminate are: (g_1, g_5), (g_1, g_6), (g_2, g_5), (g_2, g_6), (g_3, g_5), (g_3, g_6), so the number of correspondences is 6 and the score is -6. For another example, for pattern set {A-B-C}, the pairs of graphs it cannot discriminate are: (g_1, g_5), (g_2, g_5), (g_3, g_5), so the number of correspondences is 3 and the score is -3.

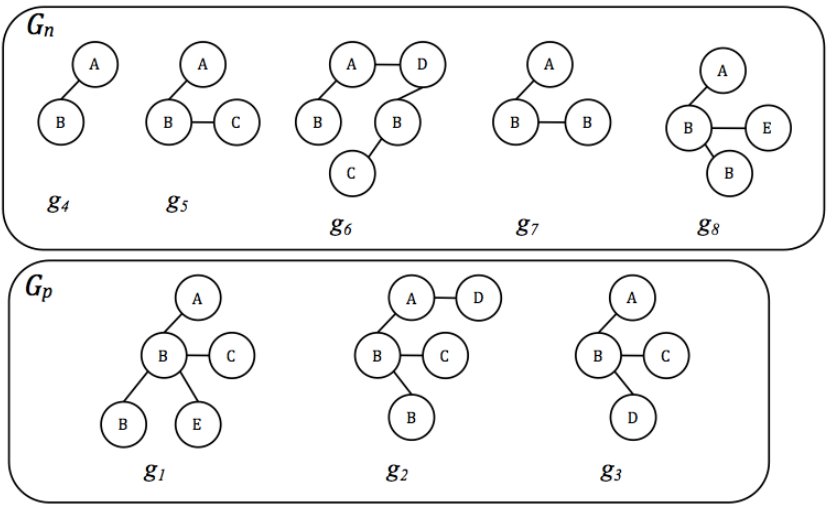


Fig. 2 An example of input positive set and negative set

Overall Framework

The overall framework of CORK is as follows:

Step 1: Initialize the resulting pattern set T as empty,

Step 2: Select subgraph pattern p that maximizes score(T $\cup$ $\{p\}$),

Step 3: If $score(T \cup \{p\}) > score(T)$, then insert p into T and go to Step 2; otherwise, return the resulting subgraph pattern set T

2.2 Branch and Bound Mining

The main idea of algorithms in this category is to use branch-and-bound search with different techniques for upper-bound estimation.

2.2.1 Leap Search and Frequency Descending Mining

Structural Leap Search

Yan et al. [24] made an observation in branch and bound search for discriminative subgraphs: if two subgraph patterns are highly similar in their structures, then there is usually strong similarity in their positive and negative frequencies as well. This is an inevitable result of huge redundancy existing in the graph pattern space. Given $|G_p|$ positive graphs and $|G_n|$ negative graphs, the possible positive

frequency values are in $\{0, \frac{1}{|G_p|}, \frac{2}{|G_p|}, ..., \frac{|G_p|-1}{|G_p|}, 1\}$ and the possible negative frequency values are in $\{0, \frac{1}{|G_n|}, \frac{2}{|G_n|}, ..., \frac{|G_n|-1}{|G_n|}, 1\}$. As we know, the number of subgraphs could easily reach an astronomic number when their frequency decreases. Therefore, an exponential number of subgraphs have to be crowded in a small frequency rectangle area $|G_p|*|G_n|$.

Figure 3 depicts the subgraph frequency distribution of the AIDS anti-viral dataset[2] with minimum frequency (0.03, 0.03). The color represents the number of subgraphs in each cell. As we can see, most of subgraphs are crowded in the left-lower corner, sharing the same frequency and the same G-test score.

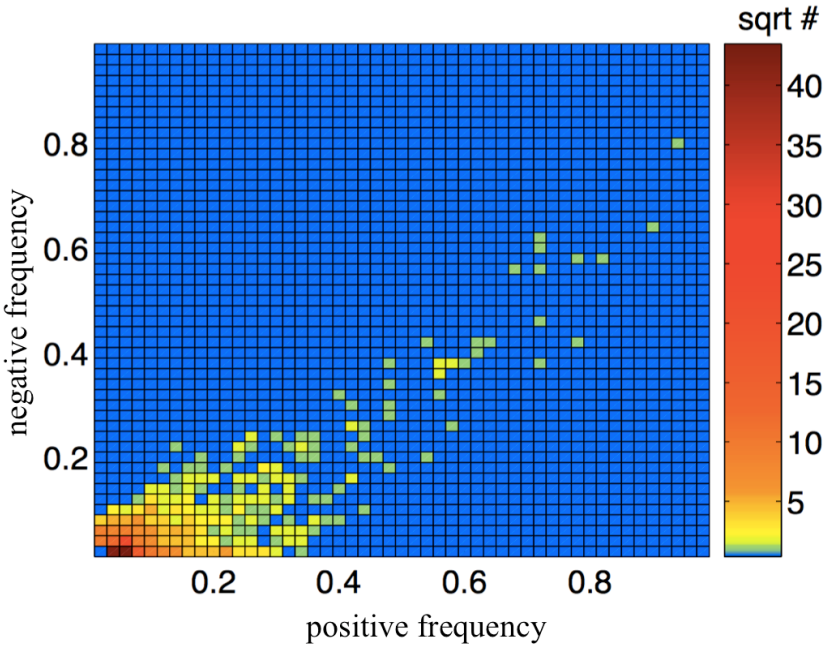


Fig. 3 Frequency distribution

According to this observation, if a subgraph pattern p has already been explored and subgraph pattern q is similar to p, pattern q can be skipped because the frequencies of q are similar to the frequencies of p and thus the G-test score of q is similar to the G-test score of p.

[2] http://pubchem.ncbi.nlm.nih.gov

The similarity between two subgraph patterns p and q is measured by the ratio of the maximum frequency difference that p and q can have to the sum of frequencies of p and q. If the ratio is less than a user specified threshold σ, then the two subgraph patterns are considered highly similar and there is no need to explore the other if one is already explored. Let $\Delta_p(p, q)$ be the maximum positive frequency difference that p and q can have and $\Delta_n(p, q)$ be the maximum negative frequency difference that p and q can have. After one pattern is explored, the other can be skipped if:

$$\frac{2\Delta_p(p,q)}{pfreq(p)+pfreq(q)} \le \sigma \text{ and } \frac{2\Delta_n(p,q)}{nfreq(p)+nfreq(q)} \le \sigma$$

This subgraph pattern pruning can be further extended to prune a whole search branch instead of an individual subgraph pattern.

The algorithm of structural leap search is described as below.

Algorithm: *Structural_Leap_Search*
Input:
G_p: a set of positive graphs
G_n: a set of negative graphs
σ: difference threshold
Output:
p^*: optimal subgraph pattern candidate

17. S = {1-edge subgraph patterns};
18. $p^* = \varnothing$;
19. *G-test*(p^*) = -∞;
20. **while** S ≠ ∅
21. p = next subgraph pattern in S;
22. $S = S - \{p\}$;
23. **if** p was examined
24. **continue**;
25. **if** ∃ q, q was examined and $\frac{2\Delta_p(p,q)}{pfreq(p)+pfreq(q)} \le \sigma$ and $\frac{2\Delta_n(p,q)}{nfreq(p)+nfreq(q)} \le \sigma$
26. **continue**;
27. **if** *G-test*(p) > *G-test*(p^*)
28. $p^* = p$;
29. **if** upper bound of *G-test*(p) ≤ *G-test*(p^*)
30. **continue**;
31. $S = S \cup$ {supergraphs of p with one more edge};
32. **return** p^*;

Frequency Descending Mining

Yan et al. [24] discovered that if all subgraphs are sorted in ascending order of their frequency, discriminative subgraph patterns are often in the high-end range.

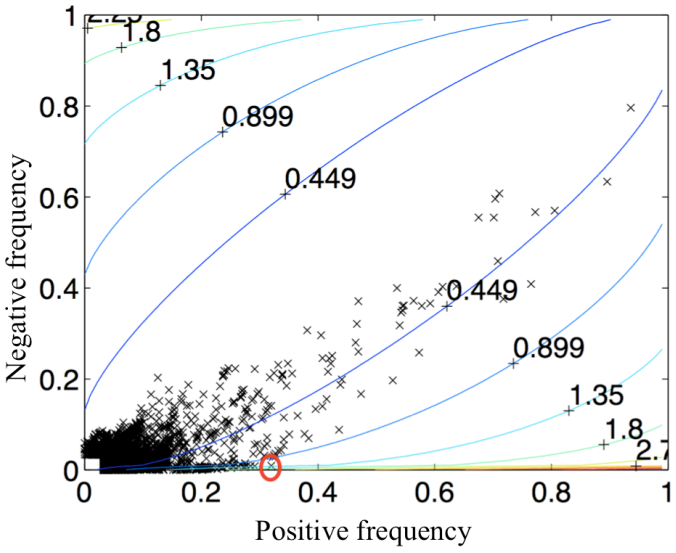


Fig. 4 Frequency vs. G-test score

Figure 4 illustrates the relationship between frequency and G-test score for the AIDS Anti-viral dataset[3]. It is a contour plot displaying isolines of G-test score in two dimensions. The X axis is the frequency of a subgraph in the positive dataset, while the Y axis is the frequency of the same subgraph in the negative dataset. The curves depict G-test score (to avoid infinite G-test score, a default minimum frequency is assumed for any pattern whose frequency is 0 in the data). Left upper corner and right lower corner have the higher G-test scores. The "circle" marks the highest G-test score subgraph discovered in this dataset. As one can see, its positive frequency is higher than most of subgraphs. Similar results are also observed in other graph datasets.

To profit from this discovery, the authors proposed an iterative frequency descending mining method.

Frequency descending mining begins the mining process with high frequency threshold $\theta = 1.0$ and it searches for the most discriminative subgraph pattern p^* whose frequency is at least θ. Then frequency descending mining repeatedly lower the frequency threshold θ to check whether it can find better p^* whose frequency is at least θ. It terminates when θ reaches either 0 or a user-specified threshold.

[3] `http://pubchem.ncbi.nlm.nih.gov`

The algorithm of frequency descending mining is described as below.

Algorithm: *Frequency_Descending_Mine*
Input:
G_p: a set of positive graphs
G_n: a set of negative graphs
ε: converging threshold
Output:
p^*: optimal subgraph pattern candidate
1. θ = 1;
2. $p = \varnothing$;
3. *G-test*(p) = -∞;
4. **do**
5. $p^* = p$;
6. S = {subgraph patterns in G_p and G_n with frequency no less than θ};
7. $p = \text{argmax}_{p' \in S} G-test(p')$;
8. θ = θ / 2;
9. **while** (*G-test*(p) - *G-test*(p) ≥ ε);
10. **return** $p^* = p$;

Overall Framework

The overall framework of LEAP is as follows:

Step 1: Use structural leap search to find the most discriminative subgraph pattern p^* with frequency threshold θ = 1.0,

Step 2: Repeat Step 1 with θ = θ / 2 until *score*(p^*) converges,

Step 3: Take *score*(p^*) as a seed score; use structural leap search to find the most discriminative subgraph pattern without frequency threshold.

2.2.2 Upper-Bound Estimation by Learning from Search History (LTS)

Pattern Exploration Order

Almost all efficient subgraph pattern exploration methods, such as gSpan [23] and FFSM [11], start with subgraphs having only one edge and extend them to larger subgraphs by adding one edge at a time. Each large subgraph pattern can be directly extended from more than one smaller subgraph patterns. For example, in Figure 5, subgraph pattern *A-B-C* can be extended from either *A-B* or *B-C*.

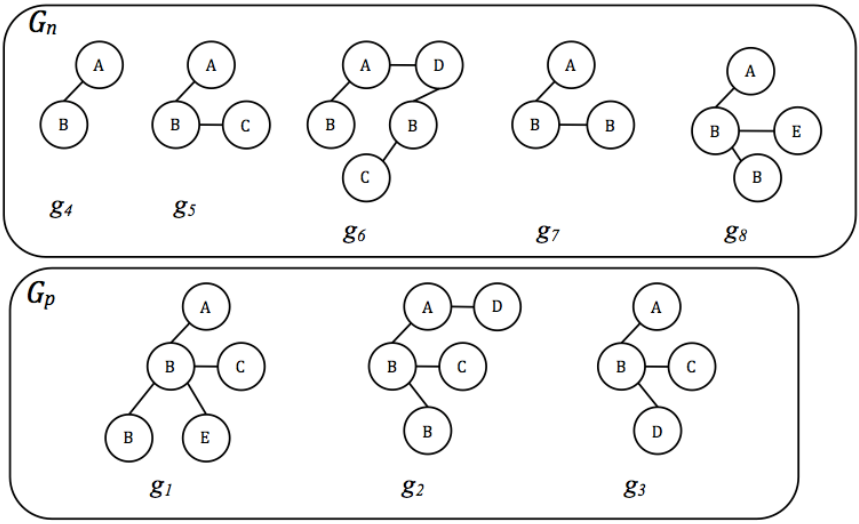


Fig. 5 An example of input positive set and negative set

Definition (Lineage). In a pattern exploration method M, a lineage of pattern p is a sequence of patterns: $l(p) = p_1p_2...p_{k-1}p_k$, where $p_k = p$, p_1 has only one edge and $\forall i \in [1, k\text{-}1]$, p_{i+1} can be directly extended from p_i by adding one more edge.

If a pattern exploration method allows a pattern to have multiple lineages, this exploration method is called a multiple-lineage exploration; otherwise, it is called a single-lineage exploration. Figure 6 shows examples of multiple-lineage exploration and single-lineage exploration for the graph sets in Figure 5. Each node represents a subgraph pattern (only patterns with less than 3 edges are shown for illustration) in the graph sets and there is a directed edge from node p to node q if in the exploration method pattern q is allowed to be reached by extending p.

A subgraph pattern may have multiple possible lineages. Thus, multiple-lineage exploration is more natural than single-lineage exploration. To achieve single-lineage exploration, an algorithm needs to define an enumeration order $<$ on all subgraph patterns in the search space. If pattern $p <$ pattern q, then p is enumerated before q. The resulting lineages become the canonical lineages of the respective patterns. Both gSpan and FFSM are single-lineage exploration methods.

The major advantage of single-lineage exploration is that it is more efficient than multiple-lineage exploration in subgraph pattern enumeration without missing any pattern. In a single-lineage exploration method, each subgraph pattern is enumerated only once while a multiple-lineage exploration method may visit a pattern multiple times through different lineages. For example, in Figure 6, the

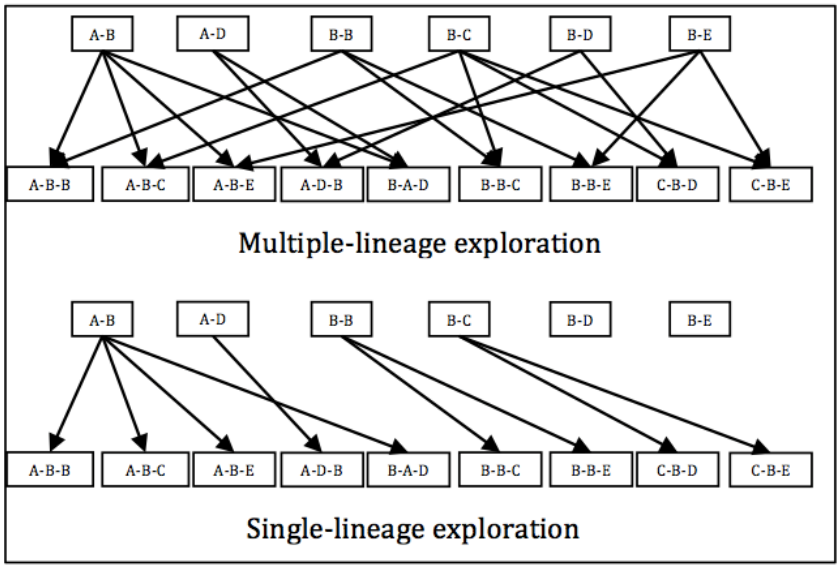


Fig. 6 An example of multiple-lineage exploration and single-lineage exploration for graphs in Figure 5

multiple-lineage exploration visits pattern *A-B-B* twice while the single-lineage exploration visits it only once. In addition, the average number of subgraph extensions performed for each subgraph pattern in single-lineage exploration is less than that in multiple-lineage exploration. For example, in Figure 6, each subgraph pattern with one edge performs three extension operations on average in multiple-lineage exploration while the average number of extension operations in single-lineage exploration is 1.5. Extension operation is the most costly operation in subgraph enumeration, thus algorithms requiring fewer extensions are highly favorable. In applications where subgraph patterns are much larger and more complex, the difference in number of extension operations becomes even larger. As a result, single-lineage exploration is preferred in most subgraph mining algorithms.

However, single-lineage exploration has the problem that its result is sensitive to subgraph pruning. Since each subgraph pattern can only be reached through a single lineage, the algorithm will miss a subgraph pattern if any subgraph on its lineage is pruned. On the contrary, multiple-lineage exploration is much more tolerant of subgraph pruning because a subgraph pattern can be reached through more than one lineage. This difference does not create any problem for using single-lineage exploration in frequent subgraph mining because of the antimonotonicity property of pattern frequency. In frequent subgraph mining, if

pattern p is in the lineage of pattern q and q is a frequent pattern, then p must also be frequent by the mining algorithm. However, in discriminative subgraph pattern mining, the redundancy in multiple-lineage exploration becomes its advantage over single-lineage exploration. Objective functions to measure discrimination power of subgraphs are usually not antimonotonic. If pattern p is in the lineage of pattern q and q is a discriminative pattern, p is not necessarily discriminative. Under such circumstances, multiple-lineage exploration can be aggressive in pruning patterns with low discrimination scores while single-lineage exploration cannot afford to prune any pattern unless it is absolutely certain that the pattern will not lead to any discriminative pattern.

For example, in Figure 5, *A-B-C* is a highly discriminative subgraph pattern in the positive set while *A-B* is not discriminative as it appears in every positive and negative graph. The single-lineage exploration shown in Figure 6 cannot prune *A-B* because otherwise *A-B-C* will be missed. The multiple-lineage exploration in Figure 6 can afford to prune *A-B* since *A-B-C* can also be reached from *B-C*.

In the proposed algorithm, LTS, Jin et al. adopted multiple-lineage exploration to reduce the risk of missing the most discriminative subgraph patterns due to pruning. Jin et al. used CCAM code to encode subgraph patterns and maintain a lookup table for subgraph patterns that have been extended to avoid extending a subgraph pattern repeatedly. Embeddings of subgraph patterns in the graph sets are also maintained to facilitate subgraph extension and frequency calculation.

Fast Probing Subgraph Pattern Space

As mentioned earlier in this chapter, a greedy algorithm can often reach a (relatively) discriminative subgraph quickly. Even though it may not be the optimal one, its score can be used to prune the search space. The higher the score, the better the pruning power. For example, let the estimated upper-bound for descendants of p be 1.0. By the time p is visited, if the best score so far is 1.2, all descendants of p can be pruned. But if the best score found so far is only 0.5, the algorithm is unable to perform any pruning.

In [16], Jin et al. proposed a greedy algorithm called *fast-probe* to generate a good sample of discriminative subgraphs to facilitate the subsequent branch-and-bound search. *Fast-probe* maintains a list of candidate subgraph patterns to be processed. The candidate list is initialized with all single-edge subgraph patterns in G_p. It repeatedly draws and processes a candidate pattern p from the list as long as the list is not empty. If pattern p is the optimal pattern for any positive graph at the time it is processed, *fast-probe* computes all extensions of p in the positive set with one more edge and put an extension into the candidate list if the extension has not be generated before; otherwise, p is discarded. *Fast-probe* terminates when the candidate list becomes empty. This process is efficient since only the best subgraphs are extended to generate candidate patterns.

The *fast-probe* algorithm is described as below.

Algorithm: *fast-probe*
Input:
G_p: positive graph set
G_n: negative graph set
Output:
the optimal pattern for each positive graph

```
1.  Put all single-edge subgraph patterns into candidate set C
2.  while (C is not empty)
3.     p ← get next pattern and remove it from C
4.     updated ← false
5.     for each graph g in G_p
6.        if score(p) > optimal score for g so far
7.           update the optimal pattern and optimal score for g
8.           updated ← true
9.     if (not updated)
10.       continue
11.    C ← all subgraph patterns with one more edge attached to p
12.    for each pattern q in C
13.       if q has not been generated before
14.          put q into C
15. return the optimal pattern for each g in G_p
```

An indicator function for a subgraph pattern p is defined as follows:

$$d(p) = \begin{cases} 1, \exists g \in G_p, score(p) > \text{optimal score for } g \text{ so far} \\ 0, \text{otherwise} \end{cases}$$

If function d is antimonotonic as patterns are extended, then when a pattern p is visited and it fails to be the optimal pattern for any positive graph (i.e. $d(p)= 0$), the search process can safely prune p and any lineages extended from p. No supergraph of p will be the optimal pattern for any positive graph because of the antimonotonicity property that once $d(p) = 0$ no supergraph q of p will have $d(q) = 1$. If this assumption is true, then the search process would become very efficient as only good patterns need to be considered. And single lineage exploration would have been sufficient.

However, this assumption is not always true because discrimination scores of patterns may increase as patterns become larger. Therefore, even if a pattern p is not the optimal pattern for any positive graph, a supergraph of p may be the optimal pattern for some positive graph because its score is greater than the score of p.

Nevertheless, the assumption does not have to hold for all subgraph patterns to make *fast-probe* work. In fact, for the most discriminative subgraph pattern, as

long as the assumption holds for at least one of its lineages, the optimal pattern will be found. Using multiple-lineage exploration helps because the likelihood of the assumption being true for at least one lineage is much larger in multiple-lineage exploration than in single-lineage exploration. In addition, the most discriminative subgraph pattern will not be missed as long as patterns in its lineages are optimal patterns for one positive graph at the time they are visited. This is very likely to be true: it is typical that some positive graphs are covered by multiple highly discriminative subgraphs while others do not have highly discriminative subgraphs. The former are called as "rich" graphs and the latter are called as "poor" graphs. Ancestors for the highly discriminative subgraphs for "rich" graphs may cover "poor" graphs when their positive frequencies are still high. Let p be the most discriminative subgraph for a "rich" graph g and q be another highly discriminative subgraph for g. Let q be visited before any ancestor of p is visited. Patterns in the lineages of p may not be the optimal patterns for g when they are visited because they may not be as discriminative as q. However they may be the optimal patterns for some "poor" graphs and thus survive and produce a lineage to p. The most discriminative subgraphs for "poor" graphs may be missed when there are no "poorer" graphs for their ancestors to survive. In this case, a subsequent (branch and bound) search may be needed to recover the most discriminative subgraphs missed by *fast-probe*.

Upper-Bound Estimation by Learning from Search History

A tight estimated upper-bound of scores may improve the efficiency of branch-and-bound algorithms. For example, when p is visited, let the optimal score of any patterns visited so far be 1.2. If the estimated upper-bound is 1.5 (loose), then the algorithm cannot prune any descendants of p; but if the estimated upper-bound is 1.1 (tight), then the algorithm can prune all descendants of p.

In [16], the authors first studied a simple way for upper-bound estimation. According to the definition, the discrimination score increases as the negative frequency decreases, and decreases as the positive frequency decreases. A simple estimation of upper-bound for scores of descendants of p is achieved when the positive frequency remains the same as that of p and the negative frequency is zero:

$$\hat{B}(p) = \log \frac{pfreq(p)}{\varepsilon}, \text{ where } \varepsilon \text{ is a small value to replace 0}$$

This is a very loose upper-bound especially when the positive and negative frequencies of p are high. In most cases, adding edges to p causes both positive and negative frequencies to decrease. If the negative frequency decreases faster than the positive frequency, then the pattern becomes more and more discriminative; otherwise, the pattern becomes less discriminative. If the negative frequency of p is high, many edges need to be added to it to achieve zero negative frequency and as a result the positive frequency drops significantly as well. For example, in chemical compound graphs, *C-C* has positive and negative

frequencies almost equal to 100% as it is prevalent in chemical compounds. However, its most discriminative descendants typically have positive frequency less than 15% and negative frequency close to zero. Therefore, the optimal discrimination score is much lower than the estimated upper-bound, which results in inefficient pruning.

Previous discriminative subgraph mining algorithms takes advantage of the correlations between score (or frequency) of a pattern and the largest score that the descendants of the pattern may have in designing exploration orders, in order to approach the optimal score as fast as possible. However, such correlations are qualitative and can only serve as a heuristic guidance. Jin et al. proposed to learn quantitative correlations from search history and use them to estimate tight upper-bounds.

Definition (score record). Given a lineage of pattern p, $l(p) = p_1p_2...p_{k-1}p_k$, the score record for $l(p)$ is a sequence of scores for the patterns in the lineage:

$$h(p) = score(p_1), score(p_2), ..., score(p_{k-1}), score(p_k)$$

A discriminative subgraph mining process always generates many score records, which can be organized into a prefix tree, called *prediction tree*. Figure 7 shows an example of score records and the corresponding *prediction tree*.

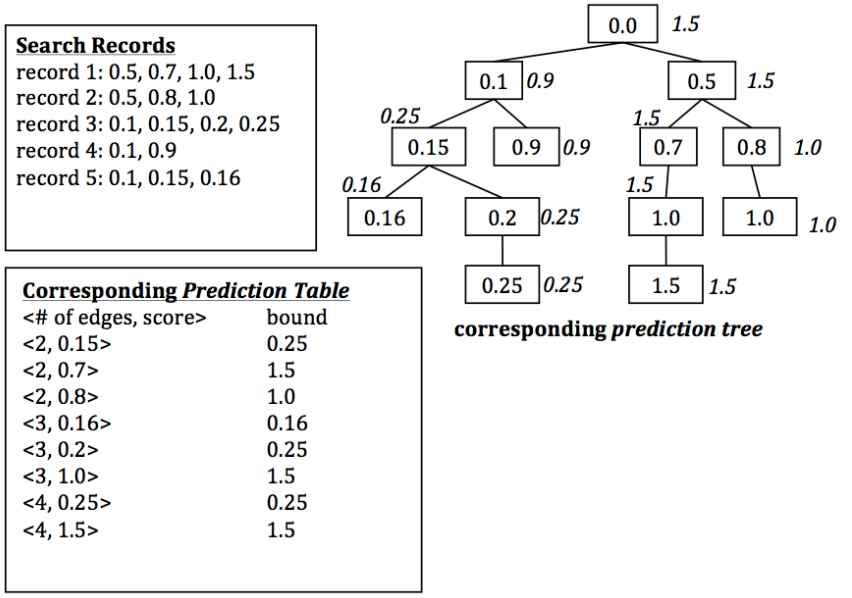


Fig. 7 An example of search records and the corresponding *prediction tree* and *prediction table*

Each tree node is labeled with score and the root node is labeled with 0.0, which is the score of an empty subgraph. In their implementation, Jin et al. discretized scores evenly into 10 bins and used the discretized scores as labels. In the example, the original scores are used as labels for the sake of intuitive illustration. In addition to the score label, each tree node is also associated with the maximum score in the sub-tree rooted at this node. The score records and the corresponding *prediction tree* can be considered as a sample of the whole search space. Therefore, the maximum score at each tree node is an estimated upper-bound in the search space. For example, for a pattern p with score record (0.5, 0.7, 1.0), its maximum score in the *prediction tree* is 1.5 and thus its estimated upper-bound in the search space is 1.5. LTS organizes the sample space by scores (rather than by subgraph structures in the search space) because it is much easier to compare scores than structures. Sometimes the score record of a pattern p is absent in the tree, so LTS additionally generates a lookup table, named *prediction table*, to aggregate the information in the tree. The key for each entry in the *prediction table* is composed of the number of edges in the pattern and the score of the pattern. The value stored at each entry is the maximum score of the descendants of the patterns with the corresponding size and score in the sample space. For example, if the score record of p is (0.4, 0.8), which cannot be found in the *prediction tree*, then LTS uses the key <2, 0.8> to look for an upper-bound estimation in the *prediction table*, which returns 1.0. The search history H is composed of the *prediction tree* and the *prediction table*. If neither the score record nor the <size, score> pair of p can be found in H, then LTS uses the loose upper-bound estimation discussed earlier in this section.

Using search history to estimate upper-bound bears the risk of underestimating upper-bound if the discriminative subgraph mining process, which provides the score records, fails to capture a good sample of high discrimination scores. This will result in inefficient pruning and thus prolonged execution time. However, there is little impact to the mining process if the greedy sampling misses many low discrimination scores because, although these score records may be absent in the *prediction tree*, the *prediction table* can still provide a reasonably tight upper-bound estimation and the algorithm always has the last resort to the loose estimation.

LTS first uses *fast-probe* to collect score records and generates search history H, which includes a *prediction tree* of score records and a *prediction table* aggregating the score records. LTS utilizes a vector F to keep track of the optimal pattern for each positive graph: $F[i]$ stores the optimal pattern for positive graph g_i. Vector F is updated with the optimal patterns found by *fast-probe*, which compose a better starting point than single-edge subgraphs, before the following branch-and-bound search. Then LTS performs a branch-and-bound search in the subgraph search space and uses a candidate list to keep track of candidate subgraph patterns. Its goal is to find the most discriminative subgraph for each positive graph. When the branch-and-bound search begins, the candidate list is initialized with all subgraphs with one edge. LTS repeatedly pops one subgraph from the candidate list at a time until the candidate list becomes empty. LTS uses

CCAM code [15] to encode subgraphs and maintains a lookup table to keep track of processed subgraphs. For each subgraph p from the candidate list, LTS updates $F[i]$ if positive graph g_i supports p and $score(p)$ is greater than $score(F[i])$. Meanwhile, LTS estimates the upper-bound of p based on search history H and checks whether the upper-bound is greater than any $score(F[i])$ with g_i supporting p. If the upper-bound is not greater than the optimal score of any positive graph supporting p, then p is discarded from further extension. Note that for each pattern, the algorithm only considers the positive graphs supporting this pattern when updating optimal scores and pruning with the estimated upper-bound because the algorithm is looking for the optimal pattern for each positive graph. If p is preserved, LTS computes all of its extensions with one more edge in the positive set. The extensions that have not been visited before are put into the candidate list.

2.3 Heuristic Search

The main idea of algorithms in this category is to use heuristic search.

2.3.1 Using Evolutionary Computation (GAIA)

Mining Discriminative Subgraph Patterns Using Evolutionary Computation

Evolutionary computation can be viewed as a generic search process for solutions of high quality or fitness, which begins with a set of sample points in the search space and gradually biases to regions of high fitness. In the problem of discriminative pattern mining, discrimination score is used to evaluate the fitness of a subgraph pattern. As a result, the evolutionary search process here is directed toward subgraph patterns with high discrimination power.

Framework of the Pattern Evolution: Organization and Resources

For each graph g_i in the positive graph set G_p, the algorithm stores a representative subgraph pattern and a list of up to s candidate subgraph patterns, where s is bounded (from above) by the available memory space divided by the number of graphs. Figure 8 illustrates the organization of candidate patterns and representative patterns. Only subgraphs of g_i with discrimination scores greater than 1 can be its representative or in its candidate list. The representative pattern has the highest discrimination score among all patterns that are subgraphs of g_i found during pattern evolution. Although one pattern can be subgraphs of several positive graphs, each pattern can only be in one candidate list at any time. The candidate lists are initialized with one-edge patterns.

The total number of subgraph patterns that the candidate lists can hold at any time is the product of s and $|G_p|$. The motivation of the design of this framework is to cause selection pressure which can significantly speed up the convergence of evolutionary search. When the total size of candidate lists is less than the total

number of patterns that can be found in positive graphs, not all patterns can be held in the candidate lists at the same time. As a result, one resource that candidate patterns need to compete for is a slot in candidate lists. In other words, patterns have to compete for survival and not all patterns are considered in the search process. Generally speaking, the larger the candidate lists are, the less selection pressure there is and thereby the more patterns are considered in the search. When the candidate lists are infinitely large, the search process becomes an exhaustive search.

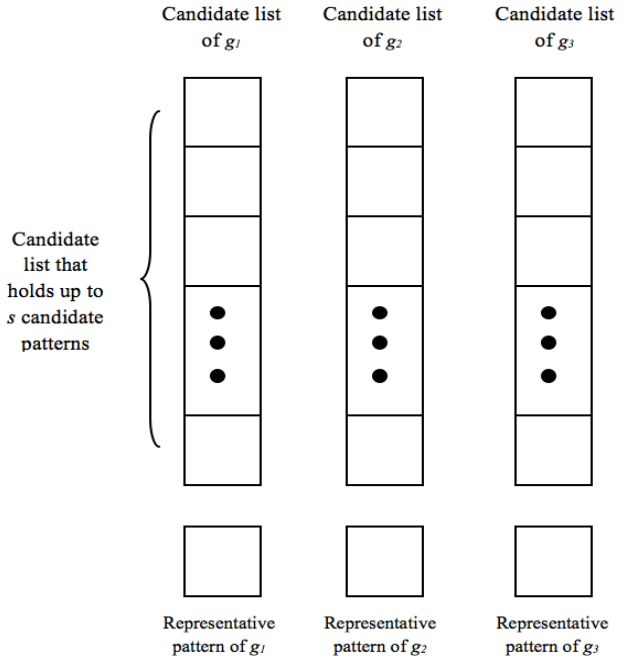


Fig. 8 An illustration of candidate pattern organization

Another resource that candidate patterns compete for is the opportunity to extend or, analogous to biological evolution, to produce offspring. All subgraph pattern mining algorithms start with small subgraph patterns and then extend them into larger patterns. However, pattern extension is a costly operation and not every pattern extension leads to a discriminative pattern. In an evolutionary search process, candidate patterns compete for the opportunity of pattern extension according to their fitness, which enables the search process to focus on candidate patterns that are more likely to lead to discriminative patterns. Although it does not guarantee that it reaches the globally optimal solution faster because of the existence of local optimal solutions, experiments show that in reality it has significant speed advantage over other methods [15].

Pattern Extension

All candidate patterns currently in the candidate lists have a non-zero probability of being selected for pattern extension. To perform pattern evolution, GAIA runs for n iterations, where n is a parameter set by the user. During each iteration, GAIA selects one pattern from each candidate list for extension. The probability of pattern p in candidate list of g_i to be selected for extension is proportional to the log ratio score of p and is calculated as follows:

$$Probability(p\ is\ selected) = \frac{score(p)}{\sum_{p' is\ in\ the\ candidate\ list\ of\ g_i} score(p')}$$

The probability is always between 0 and 1 because only patterns with positive log ratio scores are allowed in candidate lists as described in Subsection 3.2. This selection method is commonly used in evolutionary algorithms. The intuition here is that candidate patterns with higher scores are more likely to be extended to patterns with high scores because structurally similar subgraph patterns have similar discrimination power [24]. Note that when $s = 1$, each candidate list only holds 1 pattern. The probability of this pattern being selected for extension is 1. When $s > 1$, multiple patterns may be held in a candidate list. A random number generator is used to determine which pattern is selected for extension according to their probabilities.

For an extension operation of pattern p, GAIA generates a pattern set $X(p)$ and each pattern p' in $X(p)$ has one new edge attached to p. This new edge is not present in p and it can be either between two existing nodes in p or between one node in p and a new node. Unlike many previous subgraph pattern mining algorithms that only extend patterns with certain types of edges in order to efficiently maintain their canonical codes, GAIA considers all one-edge extensions of pattern p that occur in the positive graphs. This difference in extension operation is essential to GAIA because evolutionary computation is essentially a heuristic search for optimal solution. This difference enables GAIA to explore the candidate pattern space in any direction that appears promising.

Extensions of different patterns can produce the same pattern because a pattern p with k edges can be directly extended from all of its subgraphs with k-1 edges. Therefore, a lookup table is needed by GAIA to determine whether a pattern has already been generated to avoid repetitive examination of the same pattern.

Pattern Migration and Competition

In most cases, an extension operation on one pattern generates many new patterns and as a result the number of patterns found by the algorithm grows. Sooner or later the number of patterns will exceed the number of available positions in the candidate lists. It is also possible that the number of one-edge patterns already exceeds the number of available positions in the candidate lists at the very beginning if s is small. Therefore some rules are needed to determine which

patterns should survive in the candidate lists and which candidate list they should dwell in.

First, a pattern that has already been extended should not "live" in the candidate lists any longer because it has served its role in generating new patterns.

Second, some pattern in the candidate list may migrate to the candidate list of another graph if such migration will increase its chance of survival. Let p be the candidate pattern for migration and $G(p)$ be the set of graphs containing p. Let g_i be the graph in $G(p)$ which has the lowest value of $\sum_{p' is\ in\ the\ candidate\ list\ of\ g_i} score(p')$. p will migrate to the candidate list of g_i. The rationale for this pattern migration is that if a pattern wants to survive then it should go to a candidate list with the least fierce competition. In GAIA, the fierceness of competition of a candidate list is measured by the sum of scores of patterns in the list.

If the candidate list of g_i still has vacant positions, then p can move into one vacant position directly. However, if the candidate list is already full, then p has to compete with the "resident" patterns in the list. One straightforward approach to let p compete with "resident" patterns is to compare the log ratio score of p and the minimum log ratio score among "resident" patterns. If the score of p is greater than the minimum score among "resident" patterns, then p takes the position of pattern p' with the minimum score and p' no longer exists in any candidate list; otherwise, p fails to survive and will not exist in any candidate list. The disadvantage of this greedy approach is that it ignores the fact that patterns with low log ratio scores may still have some potential to extend into patterns with high log ratio scores and patterns with high log ratio scores at the time may have reached their limits and will never extend to better patterns. Therefore, GAIA adopts a randomized method for pattern competition which is commonly used by evolutionary algorithms. The score of p is compared against the score of a pattern p', which is randomly selected with probability $1/s$ from the candidate list. If the score of p is higher, then p' is eliminated and p takes the position of p'; otherwise, p is eliminated. By doing so, GAIA can at least have a chance to protect some of the "weak" patterns and give them an opportunity to extend into "strong" patterns. The benefit of this randomized approach is more evident when s is reasonably large. Note that when $s = 1$ the randomized strategy is essentially the same as the greedy strategy.

Again, the exhaustive extension operation is of great importance to allow pattern competition and elimination. When GAIA eliminates a pattern p, the real loss is not only this pattern but also the patterns generated by extending p. In previous subgraph pattern mining algorithms, such as gSpan [23] and FFSM [11], a pattern p can only be extended from one of its subpatterns, p'. If p' is lost, then the algorithms will never find p. As a result, for these algorithms, allowing pattern elimination will surely lose many patterns, some of which are discriminative patterns. But in GAIA, eliminating p' does not necessarily lead to the loss of p because the exhaustive extension operation allows p to be extended from many different patterns. As a result, the risk of missing discriminative patterns is much lower than other subgraph mining algorithms.

The algorithms of pattern migration and evolution are described as below.

Algorithm: *Pattern_Migrate*
Input:
p: a subgraph pattern
T: candidate lists
1. $g = argmin_{g'} (\sum_{p \text{ is in the candidate list of } g'} score(p))$
2. **if** (the candidate list of *g* has vacant positions)
3. insert *p* into the candidate list of *g*
4. **else**
5. randomly select a pattern *p'* in the candidate list of *g*
6. **if** (*score* (*p*) > *score* (*p'*))
7. replace *p'* with *p*

Algorithm: *Pattern_Evolution*
Input:
G_p: positive graph set
G_n: negative graph set
s: maximum size of each candidate list, by default equal to available_space/|G_p|
n: maximum number of iterations, by default the maximum interger value in the system
Output:
representative patterns: the best pattern for each positive graph
D = {all edges that occur in G_p }
1. **for each** edge *e* in *D*
2. *Pattern_Migrate* (*e, T*)
3. **for** *k* = 1:*n*
4. **if** (all candidate lists are empty)
5. **break**
6. **for each** *g* in G_p
7. randomly select a pattern *p* in the candidate list of *g*
8. *X* (*p*) = {all patterns in G_p with one more edge attached to *p*}
9. **for each** pattern *p'* in *X* (*p*)
10. **if** (CCAM code of *p'* is in *H*)
11. **continue**
12. insert *p'* into *H*
13. *Migrate* (*p'*, *T*)
14. update representative patterns

Generating Consensus Results

Because GAIA is a randomized algorithm (when $s > 1$), each single run of pattern evolution may generate different representative patterns and consume varying amount of CPU time. Some runs of pattern evolution may find better representative patterns than others and thus lead to classifiers with higher normalized accuracy. Therefore, if GAIA runs many instances of pattern evolution in parallel and selects the best subgraph patterns from all representative patterns found by these instances of pattern evolution, it is very likely that GAIA can get a better set of discriminative subgraph patterns than using representative patterns from one instance of pattern evolution alone. Therefore, by generating a consensus model based on many parallel instances of pattern evolution and only using the fastest instances of pattern evolution, GAIA can improve the discrimination power of its results and achieve faster expected response by taking advantage of parallel computing.

2.3.2 Mining Co-occurrences of Small Patterns Instead of Large Patterns (COM)

Pattern Exploration Order Based on CAM

All patterns in a graph set can be organized in a tree structure. Each tree node represents a pattern and is a supergraph of its parent node, with the root node being an empty graph. Traversing this tree can enumerate all distinct patterns without repetition. To facilitate this, a graph canonical code is often employed. Several graph-coding methods have been proposed for this purpose. COM adopted the CAM (Canonical Adjacency Matrix) code [11], but this method can be easily applied to other graph coding strategies.

The code of a graph *g* is not unique because g may have up to (n!) different adjacency matrices. So COM used standard lexicographic order on sequences to define a total order on all possible codes. The matrix that produces the maximal code for a graph g is called the Canonical Adjacency Matrix of g and the corresponding code is the CAM code of *g*. The CAM code of a graph g is unique. It is proved that exploring a pattern tree with the CAM codes can enumerate all patterns without repetition.

For example, *A-D-E* is a pattern in graph *P1* in Figure 9. Figure 10 shows two different adjacency matrices of *A-D-E*. A "1" indicates the existence of an edge between two nodes while a "0" indicates the absence of an edge. Adjacency matrix M leads to code A1D01E and adjacency matrix N leads to code D1A10E. Although both of them are correct codes of *A-D-E*, D1A10E is less than A1D01E lexicographically. In fact, A1D01E is the largest code for *A-D-E*, so it is the CAM code and adjacency matrix M is the canonical adjacency matrix.

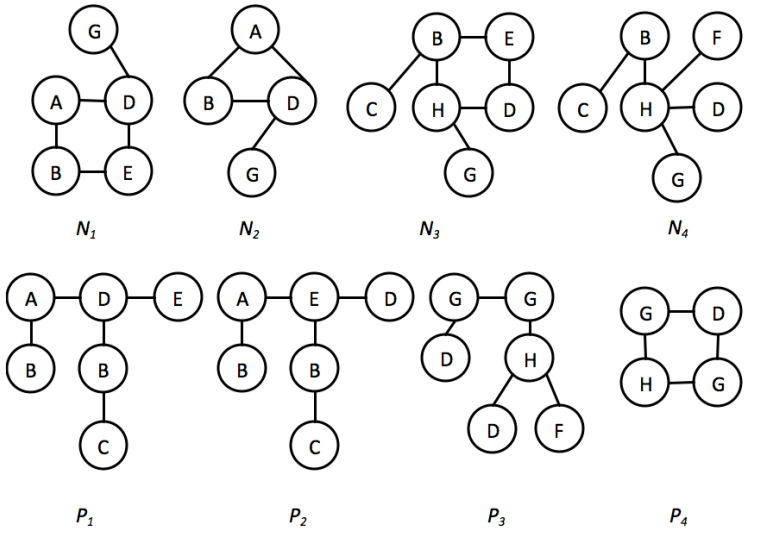


Fig. 9 An example of two sets of graphs

A	1	0
1	D	1
0	1	E

adjacency matrix M

D	1	1
1	A	0
1	0	E

adjacency matrix N

Fig. 10 An example of adjacency matrices

A Better Pattern Exploration Order

With a given scoring function, COM can rank all patterns by their scores. COM reorganizes the pattern tree to increase the probability that COM visits patterns with higher score ranks earlier than those with lower score ranks. The need for a more effective pattern exploration order is due to the fact that most pattern enumeration algorithms tend to visit patterns with similar conformations together since they usually have similar codes. This does not cause any side effect on effectiveness of pattern enumeration, but it has a huge negative impact on finding complementary discriminative patterns because patterns with similar conformations are much more likely to have overlapping supporting sets.

COM takes advantage of the following observation: let p be a pattern in the pattern tree and p' be the parent pattern of p, the score rank of p is correlated with the value of $\Delta(p)=score(p) - 2score(p')$. For patterns with two nodes, COM sets their Δ values equal to their scores $score(p)$.

Therefore, when COM explores the pattern space, it first enumerates all patterns with 2 nodes as candidates and inserts them into a heap structure with the candidate having the highest Δ value at the top. Ties are broken by favoring higher positive frequency and then by CAM code order. Then COM always takes the pattern at the top of the heap and generates all of its super-patterns with one more edge by performing the CAM extension operation. COM inserts new patterns into the heap structure. In this way, COM is able to visit patterns with high score ranks early and patterns with overlapping supporting sets late. The enumeration algorithm is described as follows.

Algorithm: *COM_enumerate_subgraphs*
Input:
G: input graph dataset
1. $P \leftarrow$ {all subgraphs with 2 nodes in G}
2. $p \leftarrow argmax_{p' \in P}(\Delta(p'))$
3. **while** ($p \neq NULL$)
4. $e \leftarrow \{CAM\_extension(p)\}$
5. **for each** $p' \in e$
6. **if** p' has not been visited
7. $P \leftarrow P \cup \{p'\}$
8. $P \leftarrow P - \{p\}$
9. $p \leftarrow argmax_{p' \in P}(\Delta(p'))$

Generating Co-occurrences

Any set of subgraph patterns can form a co-occurrence, but not all of them have high discrimination power. Ideally, the algorithm can find co-occurrences consisting of subgraph patterns with high frequency in the positive graph set and low frequency in the negative graph set. Therefore, COM used two user-specified parameters t_p and t_n to quantify the quality of a co-occurrence, where t_p is the minimal positive frequency allowed for a resulting co-occurrence and t_n is the maximal negative frequency permitted. The goal of COM is to find a co-occurrence set R such that each positive graph contains at least one co-occurrence, where each co-occurrence in R has positive frequency no less than t_p and negative frequency no greater than t_n.

This problem can be proved to be equivalent to the set cover problem and is therefore NP complete. It is intractable to find an optimal solution in the enormous pattern space. Therefore, COM adopted a greedy approach for rule generation. Let the candidate co-occurrence set be R_t and the resulting co-occurrence set be R. The

algorithm explores the pattern space with the heuristic order in Chapter 0 and whenever it comes to a new pattern p that has not been processed before, if there exists one positive graph that contains p but none of the existing co-occurrences, the algorithm generates a new candidate co-occurrence containing only p and examines the possibility of merging this new co-occurrence into existing candidate co-occurrences. Given a new pattern p and a candidate co-occurrence r_t, $\Delta(p, r_t) = score(r_t \cup \{p\}) - score(p)$. Pattern p is to be inserted into candidate co-occurrence r', $r' = argmax(\Delta(p, r_t))$, $\Delta(p, r_t) \geq 0$. If there are patterns in r' whose supporting sets are supersets of the supporting set of p, then inclusion of p into r' will make these patterns redundant. These patterns will be removed from r' when p is inserted. Then, for either the newly generated co-occurrence $\{p\}$ or the updated r', if it has $pfreq(p) \geq t_p$ and $nfreq(p) \leq t_n$ and it is present in at least one positive graph that does not contain any co-occurrence in R, it will be removed from R_t and inserted into R. The algorithm terminates either when all patterns are explored or when each positive graph contains at least one resulting co-occurrence. Although in the worst case the algorithm is still exhaustive, experiments show that it is time efficient in practice.

For example, let $t_p = 50\%$ and $t_n = 0\%$, in Figure 9, the frequent subgraphs of 2 nodes in the positive set are *A-B, B-C, D-E, D-G,* and *G-H*. Only positive patterns with frequency no less than t_p need to be considered because (1) as mentioned earlier the algorithm only considers positive patterns and (2) the frequency of a co-occurrence with patterns less frequent than t_p must be less than t_p as well. The algorithm initializes the rule sets to be empty: $R' = \{\}$ and $R = \{\}$.

According to the pattern exploration order introduced in Chapter 0, *A-B* is the first pattern to process. For simplicity, the example is designed so that these edges cannot extend to any larger patterns with positive frequency no less than t_p. A new candidate co-occurrence $\{A\text{-}B\}$ is added into R'. Note that R' was empty and thus there does not exist any rule in R' to insert *A-B*. Next, $\{B\text{-}C\}$ is added into R' and *B-C* is added into candidate co-occurrence $\{A\text{-}B\}$ because $\Delta(\{B\text{-}C\}, \{A\text{-}B\})$ is no less than 0. The modified candidate co-occurrence $\{A\text{-}B, B\text{-}C\}$ have $pfreq \geq t_p$ and $nfreq \leq t_n$, therefore it is removed from R' and added into R. Next, *D-E* is at the top of the heap, but there is no need to consider it because both of its supporting graphs, P_1 and P_2, contain co-occurrence $\{A\text{-}B, B\text{-}C\}$ and therefore considering *D-E* cannot lead to a better classifier. Then, following a similar procedure, the algorithm can generate co-occurrence $\{D\text{-}G, G\text{-}H\}$ and add it into R. Now the algorithm terminates because: 1) the heap structure for candidate patterns is empty and 2) $\{A\text{-}B, B\text{-}C\}$ and $\{D\text{-}G, G\text{-}H\}$ are sufficient to cover all graphs in the positive set. For each step, the initial status of R', R, the pattern at the heap top and the set of positive graphs that contain none of the co-occurrences in R are shown in Figure 11.

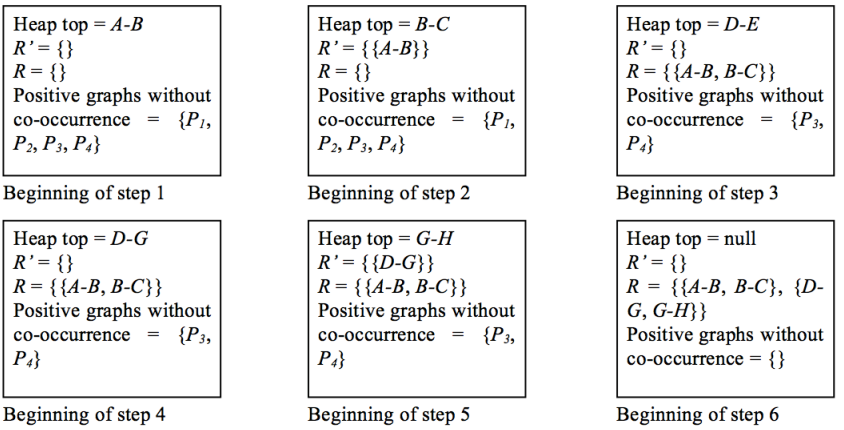


Fig. 11 An example of co-occurrence generation

3 Evaluation

3.1 Experiment Setup

We used protein datasets and chemical compound datasets in our experiments for evaluation. We evaluated the performance of GAIA, COM, LTS and LEAP, whose efficient implementations are available, by their run time efficiency and the g-test scores of the best patterns they found. The g-test score of a pattern p is defined as:

$$G\text{-test score of } p = pfreq(p) * \log\frac{pfreq(p)}{nfreq(p)} + (1 - pfreq(p)) * \log\frac{1 - pfreq(p)}{1 - nfreq(p)}$$

The protein datasets consist of protein structures from Protein Data Bank[4] classified by SCOP[5] (Structural Classification of Proteins). We selected all large SCOP families with more than 25 members. In each dataset, protein structures in a selected family are taken as the positive set. Unless otherwise specified, we randomly selected 250 outsider proteins (i.e., not members of the 16 families) as a common negative set used by all 16 protein datasets. To generate a protein graph, each graph node denotes an amino acid, whose location is represented by the location of its alpha carbon. There is an edge between two nodes if the distance between the alpha carbons of two amino acids is less than 11.5 angstroms. Nodes are labeled with their amino acid type and edges are labeled with the discretized distance between the alpha carbons. On average, each protein graph has approximately 250 nodes and 2700 edges.

[4] http://www.rcsb.org/pdb
[5] http://scop.mrc-lmb.cam.ac.uk/scop/

The chemical compound datasets consist of chemical compound structures from PubChem[6] classified by their biological activities. Each compound can be either active or inactive in a bioassay. For each bioassay, we randomly selected 400 active compounds as the positive set and 400 inactive compounds as the negative set. In chemical compound graphs, each atom is represented by a graph node labeled with the atom type and each chemical bond is represented by a graph edge labeled with the bond type. On average, each compound graph has 54.76 nodes and 57.24 edges.

Table 1 List of selected SCOP families

SCOP ID	Family name	# of proteins
46463	Globins	51
47617	Glutathione S-transferase (GST)	36
48623	Vertebrate phospholipase A2	29
48942	C1 set domains	38
50514	Eukaryotic proteases	44
51012	alpha-Amylases, C-terminal beta-sheet domain	26
51487	beta-glycanases	32
51751	Tyrosine-dependent oxidoreductases	65
51800	Glyceraldehyde-3-phosphate dehydrogenase-like	34
52541	Nucleotide and nucleoside kinases	27
52592	G proteins	33
53851	Phosphate binding protein-like	32
56251	Proteasome subunits	35
56437	C-type lectin domains	38
88634	Picornaviridae-like VP	39
88854	Protein kinases, catalytic subunit	41

Table 2 List of selected bioassay IDs

Bioassay ID	Tumor description	# of actives	# of inactives
1	Non-Small Cell Lung	2047	38410
33	Melanoma	1642	38456
41	Prostate	1568	25967
47	Central Nerv Sys	2018	38350
81	Colon	2401	38236
83	Breast	2287	25510
109	Ovarian	2072	38551
123	Leukemia	3123	36741
145	Renal	1948	38157
167	Yeast anticancer	9467	69998
330	Leukemia	2194	38799

[6] http://pubchem.ncbi.nlm.nih.org

3.2 Comparison

For chemical datasets, LEAP finds the most discriminative subgraph patterns among the four algorithms, but it is almost two orders of magnitude slower than the other three algorithms. Therefore, if optimizing pattern discrimination power is crucial and dataset is relatively small, LEAP is the best choice for discriminative subgraph pattern mining. However, when the dataset is large, LEAP cannot finish in a reasonable amount of time. GAIA and LTS offer better trade-off between pattern quality and runtime efficiency. Between the two, LTS finds better patterns in less time than GAIA. COM is faster than LEAP, but it does not find competitive subgraph patterns compared with LEAP.

Table 3 Performance comparison between GAIA, LTS, COM and LEAP

		GAIA	LTS	COM	LEAP
protein datasets	runtime (sec)	**2.63**	3.27	5.70	421
	best score	7.06	**7.45**	5.15	5.55
chemical datasets	runtime (sec)	1.21	**0.720**	5.25	62.9
	best score	0.803	0.813	0.06	**0.847**

For protein datasets, LEAP does not find the most discriminative subgraph patterns among the four algorithms, even though it still takes much longer time because its structural leap search is less efficient when the candidate patterns are less similar to each other. Both LTS and GAIA are significantly faster than LEAP and find more discriminative subgraph patterns. LTS finds more discriminative subgraph patterns than GAIA, but takes slightly longer time than GAIA. COM is faster than LEAP, but its patterns are not as discriminative as that of LEAP.

In general, the strength of LEAP and CORK is to find subgraph patterns with optimal discrimination scores and the cost is significantly longer runtime. In addition, experiments show that LEAP is more capable at processing graphs whose subgraphs are more similar to each other. We did not have access to the original implementation of CORK and thus were unable to evaluate CORK in experiments, but CORK can only use a specific measurement for discrimination power and the measurement can be undesirable in certain applications.

The strength of GAIA, LTS, graphSig and COM is to provide better trade-off between subgraph discrimination power and runtime efficiency. Among the four, graphSig has the advantage of making good use of domain knowledge because well studied substructures can be used as features to facilitate the mining process. Experiments show that COM is faster than LEAP but its pattern quality is not

competitive, at least for the protein and chemical datasets we tested. The advantage of LTS is its fast speed and highly competitive pattern quality. In fact, it outperforms LEAP for the protein datasets and only trails LEAP slightly for the chemical datasets in terms of pattern discrimination power. The advantage of GAIA is that it can be run in parallel to further improve runtime efficiency.

4 Summary

Discriminative subgraph patterns can be used to identify feature substructures and perform structure classification. Many research studies have been devoted to developing efficient and effective discriminative subgraph mining algorithms. Higher efficiency allows users to process larger graph datasets and higher effectiveness enables users to achieve better results in applications including protein classification, protein active site identification, chemical compound activity prediction, etc. Various techniques to improve efficiency and effectiveness are introduced and evaluated in this chapter.

References

1. Bandyopadhyay, D., Huan, J., Liu, J., Prins, J., Snoeyink, J., Wang, W., Tropsha, A.: Structure-based function inference using protein family-specific fingerprints. Protein Science 15, 1537–1543 (2006)
2. Bandyopadhyay, D., Huan, J., Prins, J., Snoeyink, J., Wang, W., Tropsha, A.: Identification of family-specific residue packing motifs and their use for structure-based protein function prediction: I. Method development. J. Comput. Aided Mol. Des. (2009)
3. Bandyopadhyay, D., Huan, J., Prins, J., Snoeyink, J., Wang, W., Tropsha, A.: Identification of family-specific residue packing motifs and their use for structure-based protein function prediction: II. Case studies and applications. J. Comput. Aided Mol. Des. (2009)
4. Chen, B.Y., et al.: Geometric sieving: Automated distributed optimization of 3D motifs for protein function prediction. In: Apostolico, A., Guerra, C., Istrail, S., Pevzner, P.A., Waterman, M. (eds.) RECOMB 2006. LNCS (LNBI), vol. 3909, pp. 500–515. Springer, Heidelberg (2006)
5. Chen, W.-Y., Zhang, D., Chang, E.: Combinational Collaborative Filtering for Personalized Community Recommendation. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 115–123 (2008)
6. Fei, H., Huan, J.: Structure Feature Selection For Graph Classification. In: ACM 17th International Conference of Knowledge Management 2008 (CIKM 2008), Napa Valley, California (2008)
7. Fei, H., Huan, J.: Boosting with Structure Information in the Functional Space: an Application to Graph Classification. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, SIGKDD (2010)

8. Fröhlich, H., Wegner, J.K., Sieker, F., Zell, A.: Optimal Assignment Kernels for Attributed Molecular Graphs. In: Proceedings of the 22nd International Conference on Machine Learning (ICML), pp. 225–232 (2005)
9. Helma, C., Cramer, T., Kramer, S., Raedt, L.D.: Data mining and machine learning techniques for the identification of mutagenicity inducing substructures and structure activity relationships of noncongeneric compounds. J. Chem. Inf. Comput. Sci. 44, 1402–1411 (2004)
10. Hsu, H., Jones, J.A., Orso, A.: RAPID: Identifying bug signatures to support debugging activities. In: ASE (Automated Software Engineering) (2008)
11. Huan, J., Wang, W., Prins, J.: Efficient mining of frequent subgraph in the presence of isomorphism. In: Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM), pp. 549–552 (2003)
12. Huan, J., Wang, W., Bandyopadhyay, D., Snoeyink, J., Prins, J., Tropsha, A.: Mining spatial motifs from protein structure graphs. In: RECOMB, pp. 308–315 (2004)
13. Huan, J., Bandyopadhyay, D., Prins, J., Snoeyink, J., Tropsha, A., Wang, W.: Distance-based identification of spatial motifs in proteins using constrained frequent subgraph mining. In: Proceedings of the LSS Computational Systems Bioinformatics Conference (CSB), pp. 227–238 (2006)
14. Jin, N., Young, C., Wang, W.: Graph Classification Based on Pattern Co-occurrence. In: Proceedings of the ACM 18th Conference on Information and Knowledge Management (CIKM), pp. 573–582 (2009)
15. Jin, N., Young, C., Wang, W.: GAIA: graph classification using evolutionary computation. In: Proceedings of the ACM SIGMOD International Conference on management of Data, pp. 879–890 (2010)
16. Jin, N., Wang, W.: LTS: Discriminative subgraph mining by learning from search history. In: ICDE 2011, pp. 207–218 (2011)
17. Khan, A., Yan, X., Wu, K.-L.: Towards Proximity Pattern Mining in Large Graphs. In: SIGMOD 2010 (Proc. 2010 Int. Conf. on Management of Data) (June 2010)
18. Ranu, S., Singh, A.K.: GraphSig: A Scalable Approach to Mining Significant Subgraphs in Large Graph Databases. In: Proceedings of the 25th International Conference on Data Engineering (ICDE), pp. 844–855 (2009)
19. Smalter, A., Huan, J., Lushington, G.: A Graph Pattern Diffusion Kernel for Chemical Compound Classification. In: Proceedings of the 8th IEEE International Conference on Bioinformatics and BioEngineering, BIBE 2008 (2008)
20. Smalter, A., Huan, J., Lushington, G.: Graph Wavelet Alignment Kernels for Drug Virtual Screening. Journal of Bioinformatics and Computational Biology 7(3), 473–497 (2009)
21. Saigo, H., Kraemer, N., Tsuda, K.: Partial Least Squares Regression for Graph Mining. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008), pp. 578–586 (2008)
22. Thoma, M., Cheng, H., Gretton, A., Han, J., Kriegel, H., Smola, A., Song, L., Yu, P., Yan, X., Borgwardt, K.: Near-optimal supervised feature selection among frequent subgraphs. In: SDM 2009, Sparks, Nevada, USA (2009)
23. Yan, X., Han, J.: gSpan: graph-based substructure pattern mining. In: Proceedings of the 2002 IEEE International Conference on Data Mining, pp. 721–724 (2002)
24. Yan, X., Cheng, H., Han, J., Yu, P.S.: Mining significant graph patterns by leap search. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 433–444 (2008)

25. Yao, H., Kristensen, D.M., Mihalek, I., Sowa, M.E., Shaw, C., Kimmel, M., Kavraki, L., Lichtarge, O.: An accurate, sensitive, and scalable method to identify functional sites in protein structures. J. Mol. Biol. 326, 255–261 (2003)
26. Zhang, X., Wang, W., Huan, J.: On demand Phenotype Ranking through Subspace Clustering. In: Proceedings of SIAM International Conference on Data Mining, SDM (2007)
27. Zhang, S., Yang, J.: RAM: Randomized Approximate Graph Mining. In: Proceedings of the 20th International Conference on Scientific and Statistical Database Management, pp. 187–203 (2008)

Path Knowledge Discovery: Multilevel Text Mining as a Methodology for Phenomics

Chen Liu, Wesley W. Chu, Fred Sabb, D. Stott Parker, and Robert Bilder

Abstract. Transdisciplinary research is a rapidly expanding part of science and engineering, demanding new methods for connecting results across fields. In biomedicine for example, modeling complex biological systems requires linking knowledge across multi-level of science, from genes to disease. The move to multilevel research requires new strategies; in this discussion we present *path knowledge discovery*, a novel methodology for linking published research findings.

The development of path knowledge discovery was motivated by problems in neuropsychiatry, where researchers need to discover interrelationships extending across brain biology that link genotype (such as dopamine gene mutations) to phenotype (observable characteristics of organisms such as cognitive performance measures). To advance an understanding of the complex bases of neuropsychiatric diseases, researchers need to search and discover relations among the many manifestations of these diseases across multiple biological and behavioral levels (i.e., genotypes and phenotypes at levels from molecular expression through complex syndromes). Phenomics — the study of phenotypes on a genome-wide scale — requires close collaboration among specialists in multiple fields. We developed a computer-aided path knowledge discovery methodology to accomplish this goal.

Path knowledge discovery consists of two integral tasks: 1) association path mining among concepts in multipart phenotypes that cross disciplines, and 2) fine-granularity knowledge-based content retrieval along the path(s) to permit deeper analysis. Implementing this methodology with our PhenoMining tools has required development of innovative measures of association strength for pairwise associations, as well as the strength for sequences of associations, in addition to powerful

Chen Liu · Wesley W. Chu · D. Stott Parker
Computer Science Department, University of California, Los Angeles, USA
e-mail: {chenliu,wwc,stott}@cs.ucla.edu

Fred Sabb · Robert Bilder
Semel Institute for Neuroscience and Human Behavior, University of California, Los Angeles, USA
e-mail: {fsabb,rbilder}@mednet.ucla.edu

W.W. Chu (ed.), *Data Mining and Knowledge Discovery for Big Data,*
Studies in Big Data 1,
DOI: 10.1007/978-3-642-40837-3_5, © Springer-Verlag Berlin Heidelberg 2014

lexicon-based association expansion to increase the scope of matching. In our discussions we describe the validation of the methodology using a published heritability study from cognition research, and we obtain comparable results. We show how PhenoMining tools can greatly reduce a domain expert's time (by several orders of magnitude) when searching and gathering knowledge from the published literature, and can facilitate derivation of interpretable results.

We built these PhenoMining tools on an existing knowledge base (PhenoWiki.org), now called PhenoWiki+, which can greatly speed up the knowledge acquisition process. Further, using the Resource Description Framework (RDF) data model in the PhenoWiki knowledge repository allows us to connect with different knowledge sources to enlarge the knowledge scope. The knowledge base also supports annotation, an important capability for collaborative knowledge discovery.

1 Introduction

Increasingly, scientific discovery requires the connection of concepts across disciplines, as well as systematizing their interrelationships. Doing this can require linking vast amounts of knowledge from very different domains. Experts in different fields still publish their discoveries in specialized journals, and even with the increasing availability of scientific literature in electronic media, it remains difficult to connect these discoveries. For example, an expert in cognitive assessment may know little about signaling pathways or genes, while an expert in genetics may lack knowledge about cognitive phenotypes. Although informatics tools such as search engines are very successful when it comes to helping people search for and retrieve information, these systems unfortunately lack the capability to connect the knowledge. To overcome this basic problem, new methodologies are needed for scalable and effective knowledge discovery and integration.

This work was motivated specifically by research on complex neuropsychiatric syndromes such as schizophrenia, ADHD and bipolar disorder. A multilevel framework has been proposed by the Consortium for Neuropsychiatric Phenomics at UCLA (www.phenomics.ucla.edu) to help systematize discovery [9]. Figure 1 presents a multilevel concept schema, with sample concepts at different levels. Under such a multilevel framework, it is important to understand the relationships among concepts at different levels, which form "paths" across the multilevel structure. For example, under the multilevel framework, one may be interested in a set of related questions as the following: What symptoms are related to schizophrenia? Which parts of the brain would be affected? What signaling pathway is related? And finally, which genes are related to this pathway? In recent studies, researchers discovered that schizophrenia patients usually have deficits in their working memory function, and working memory is related to neuroanatomic concepts such as prefrontal cortex. Further, genes such as COMT affect dopamine signaling and thus affect working memory functionality [14]. We can describe such a sequence of relationships with a path "schizophrenia → working memory → prefrontal cortex →

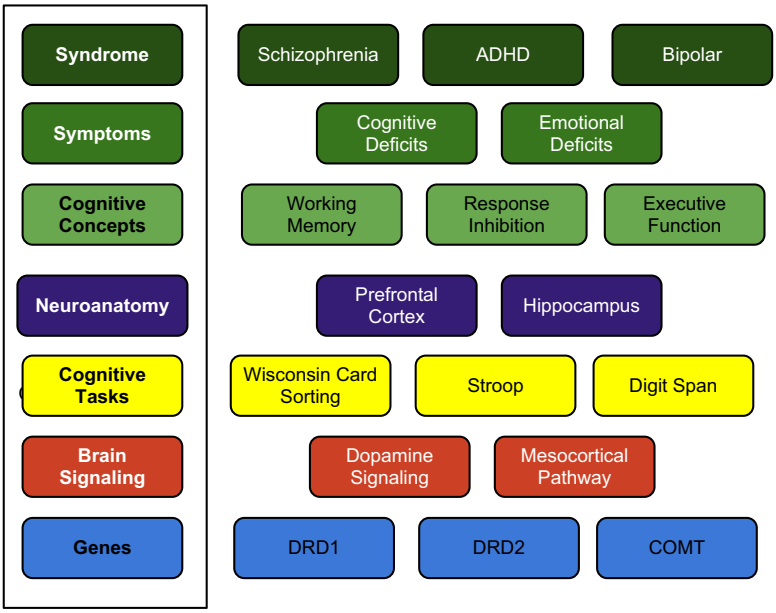


Fig. 1 The multilevel schema proposed by the Consortium for Neuropsychiatric Phenomics, at left, with a sample hierarchy of concepts at right that include phenotypes related to three syndromes

dopamine → COMT." Thus, paths are able to describe interactions among concepts and associations across disciplines.

The path schema shown in Figure 1 is both a hierarchy of relevant concepts and a hierarchy of phenotypes. Phenotypes are observable characteristics of organisms — such as color, shape, and experimentally-measured quantities. Although the space of human genetic variation is large, the space of human phenomic variation is much larger and more diverse, ranging across many science disciplines. Furthermore, phenotypes are influenced by the environment. Phenomics — the systematic study of phenotypes on a genome-wide scale [8] — requires consideration of experimental findings across a broad schema of phenotypes. By nature, then, phenomics is a transdisciplinary undertaking that requires new methodologies.

Perhaps the most essential result of this work is that paths can serve as the basis of a scalable methodology for multilevel knowledge discovery, so that in particular, when the path schema defines a hierarchy of phenotypes, the path knowledge discovery methodology can be useful for phenomics.

The path knowledge discovery problem is challenging for the following reasons. First, a path describes a sequence of associations across multiple levels of knowledge. Although existing data mining methods such as Apriori [3] perform well when identifying high-confidence pairwise associations, mining interrelated associations

still remains an open problem. Second, associations alone do not provide sufficient information for knowledge discovery. It is important to understand how the concepts are interrelated and necessary for retrieving information that can support the associations. However, a traditional information retrieval system is unable to answer such a specific query. As described above, the path knowledge discovery problem can be decomposed into two integral parts: 1) identifying paths describing relations among concepts at multiple concept levels, and 2) retrieving content corresponding to the paths from the corpus to explain the interrelations.

We developed PhenoMining tools to solve the path knowledge discovery problem in phenomics. The tools are built based on a multilevel phenotype lexicon that is constructed using domain knowledge from experts, and on a corresponding corpus of scientific literature selected by experts. Two tools have been developed to solve the two problems in path knowledge discovery above: 1) the PathMining tool is able to identify associations among concepts in the lexicon in order to construct a path based on their co-occurrence in the corpus, and it provides a quantitative way to measure the strength of associations. 2) The Document Content Explorer tool finds relevant published information for a specific path at fine granularity, so as to explain the interrelations.

These PhenoMining tools can aid in constructing a phenotype knowledge base such as PhenoWiki [25] by providing efficient path knowledge discovery. A knowledge base extension called PhenoWiki+ that integrates mining results with the knowledge base has also been developed to facilitate storage, retrieval and update of path knowledge discovered with PhenoMining tools. Figure 2 describes the process of path knowledge discovery and management using our methodology.

Section 2 presents the infrastructure that facilitates path knowledge discovery, including the multilevel lexicon and the corpus data set, and the index for association analysis and content retrieval. In Section 3 and 4 we introduce our approach for path knowledge discovery, which consists of path mining with multiple associations and relevant content retrieval. We demonstrate an application of path knowledge discovery in examination of the heritability of cognitive control in Section 5. In Section 6, we present PhenoWiki+, a knowledge repository that integrates mining results with the hierarchical multilevel framework. We then review related work in Section 7 and conclude our discussion in Section 8.

2 Infrastructure for Path Knowledge Discovery

Our path knowledge discovery tools are based on a lexicon of concepts at multiple levels of scientific inquiry and a corpus of scientific papers. The multilevel lexicon is a controlled vocabulary of concepts at different levels; this provides the knowledge of synonyms and the concept hierarchy. The corpus is stored in a semi-structured format so that information content with different granularity can be retrieved. Two indexes, the association index and document index, answer path queries and content retrieval queries, respectively.

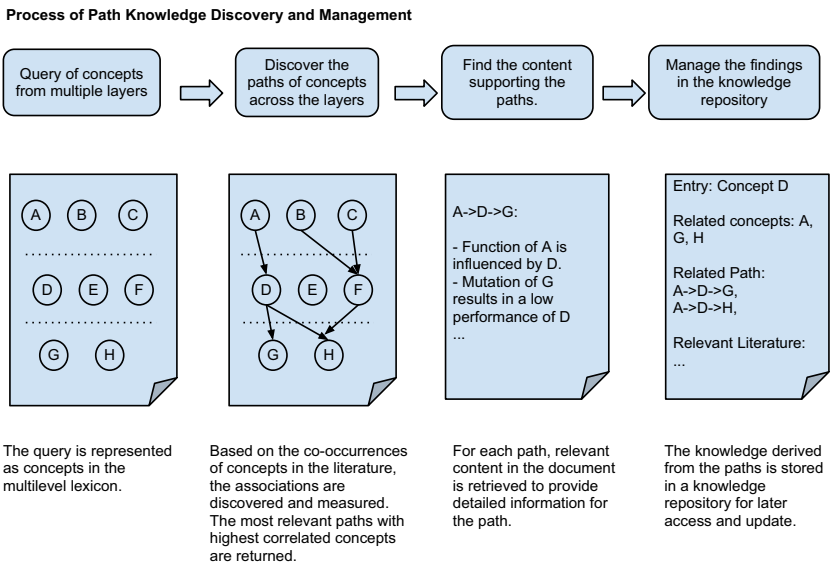


Fig. 2 Process flow of Path Knowledge Discovery and Management

2.1 Multilevel Lexicon

The multilevel lexicon provides a controlled vocabulary of concepts for the system. The lexicon is developed by domain experts according to a multilevel schema, such as the Consortium of Neuropsychiatric Phenomics (CNP) schema, which includes levels for syndromes, symptoms, cognitive concepts, neuroanatomy, cognitive tasks, brain signaling, and genes (Figure 1). In addition to these schema categories, other types of concepts are included, such as indicators, sample characteristics and subject species. These categories are used for classification of content to answer specific queries (see Section 4.3).

In the lexicon, each concept has a list of synonyms that specifies the common terms used in the literature. Because of the ongoing development of behavioral neuroscience research, some behavioral elements of the vocabulary are not as systematically defined as the neural or genetic elements, and different acronyms and synonyms are used. The synonyms listed in the lexicon reduce such variances. In addition, the lexicon also defines a hierarchy of relations among concepts. Concepts are organized in a tree structure so that more specific concepts can be defined as subconcepts of more general ones. For example, Figure 3 presents a sample hierarchy in the neuroanatomy category. Prefrontal cortex (PFC) and orbitofrontal cortex (OFC) are sub-concepts of frontal lobe, and have four concepts above them in the hierarchy. In the system the lexicon is stored in a tree structure in which sub-concepts

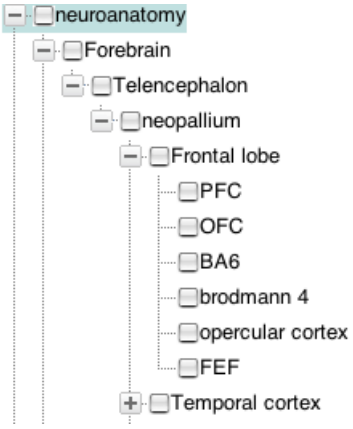


Fig. 3 Hierarchy of neuroanatomical concepts

and super-concepts of every concept can be accessed. The hierarchical structure of the lexicon is useful for query preprocessing, such as query expansion (Section 4.1). The lexicon defines 463 concepts and a hierarchy of relations among them. A complete lexicon can be found in [1].

The field of phenomics currently lacks a comprehensive ontology (like the Gene Ontology [5] for genomics) from which this multilevel hierarchy might be constructed, although existing hierarchies can certainly help, as discussed below. On the other hand, unlike comprehensive ontologies, a domain-specific lexicon can be very flexible and easy to modify. Our multilevel lexicon only includes the hierarchical structure describing the "is-a" relations, thus avoiding complexity and controversy in more complex interactions between concepts. Synonyms of concepts do not depend on one another and thus can be easier to maintain. Other relations between concepts are revealed over time by text mining in the corpus. With the advantage of domain knowledge, more features can be included in the lexicon, thus increasing the power of text mining.

Multilevel vocabularies are common in biomedical research, such as sets of terms used in scientific publications. For instance, Medical Subject Headings (MeSH) is a controlled vocabulary thesaurus used for indexing articles in the PubMed/MEDLINE database. MeSH is a tree structure containing sixteen root categories and includes in total about 25,000 descriptors [18]. These descriptors define topics covered in the literature, and are similar to the concepts in our lexicon. MeSH also includes about 200,000 entry terms that are related to the descriptors, and are similar in function to our synonym terms list. By extending the multilevel lexicon used here to a larger multilevel vocabulary such as MeSH, this work can potentially be applied to a broader scope. Indeed, MeSH is sparse at the behavioral/mental processes level, where we have invested much development.

2.2 A Corpus of Scientific Papers

Our corpus consists of a large number of full-text peer-reviewed publications. In the past decade, the availability of digitized scientific publications has significantly increased, and many scientific journals have opened access to the public. Moreover, PubMed Central (PMC)[2] is an open-access database of full-text papers and researchers with NIH funding are now required to submit publications to this database. PMC provides a collection of papers in Extensible Markup Language (XML) format, and these are downloadable for researchers. Compared to traditional pure text documents, XML formatted papers provide each of the different elements in the paper with different markup tags. Therefore, we are able to preserve the structural information of the document, such as paragraphs and sections. Moreover, we can extend from the text content to pictorial content, such as figures and tables. There are also papers available online in the form of web pages in HyperText Markup Language (HTML). Similar to an XML file, HTML is semi-structured. Content in the webpage can be extracted by specifying a pattern of tags. In the PhenoMining system, we convert different formats to a uniform XML format file called PhenoMining Document (PMDoc).

In the PMDoc file, tags are used to represent and organize general elements in a paper. We define the most basic element as a sentence, which represents a natural sentence in the text of the paper. Each sentence is assigned a sentence id, as well as a set of attributes to specify the location of the sentence with different granularity, such as section, subsection and paragraph. There are other types of elements, including figures and tables. Figure and table elements include a link to the graphic file via the publisher when available, the title of the figure or table, and the caption text from the paper. In addition, each figure and table is assigned an id, and its id will be included in the reference attribute of a sentence element where it is mentioned in the main text. Figure 4 presents an example of a sentence element in a PMDoc file. From the structure of the XML file, we can reconstruct the hierarchy of a paper (as shown in Figure 5) with the PMDoc file. Since the PMDoc file presents the text content, but also the position information in different granularity, we are able to locate the context of the sentence in different granularity; this is useful in finding the most relevant content (See Section 4.2).

The corpus is collected according to the domain of the lexicon. In PhenoMining we focus on the neuroscience domain. The initial corpus included 9000 papers retrieved from PubMed Central using the search query *((Schizophrenia OR Bipolar Disorder OR Attention Deficit Disorder) OR (Working Memory or Response Inhibition)) AND (Stop-Signal Task OR Go NoGo Task OR Spatial Capacity Task OR Digit Span Task OR Probabilistic Reversal Learning Task OR Spatial Manipulation Task OR Stroop Task)*. This query, designed by domain experts, includes important concepts at different levels so as to cover interactions among concepts and facilitate path knowledge discovery.

```
<article
 authors="Robert M. Bilder, Fred w. Sabb, D. Stott Parker,
     Donald Kalar
      Wesley W. Chu, Jared Fox, Nelson B. Freimer, and Russell
          A. Poldrack"
 date="July 2009" journal="Cognitive neuropsychiatry"
 pmcid="2752634" pmid="19634038"
 title="Cognitive Ontologies for Neuropsychiatric Phenomics
     Research">
...
<section id="3" title="Managing complexity in the Human
    Phenome Project">
...
<paragraph id="6">
 ...
<sentence id="33" paragraphId="6" refs="F1">
Within the Consortium for Neuropsychiatric Phenomics at UCLA
 (www.phenomics.ucla.edu), we have used a simple schematic
 scaffold for translational neuropsychiatric research from
     genome
 to syndrome, using seven levels (see Figure 1).
</sentence>
</paragraph>
...
</section>
...
<figure id="F1" refs="33" url="/pmc/articles/PMC2752634/
    figure/F1/"
smallThumb="/pmc/articles/PMC2752634/bin/nihms-134130-f0001.
    gif"
title="Figure 1">
  Simplified schematic of multilveled -nomics domains for
      cognitive neuropsychiatry.
</figure>
...
</article>
```

Fig. 4 The PMDoc presentation of document elements for the paper "Cognitive Ontologies for Neuropsychiatric Phenomics Research" [9]

2.3 Index for Path Knowledge Discovery

Indexing is an essential infrastructure component for efficient retrieval of content and query answering. We developed two types of indexing to facilitate path knowledge discovery. The document element index has been developed for content retrieval, and the association index for discovery of relations across concepts at different levels.

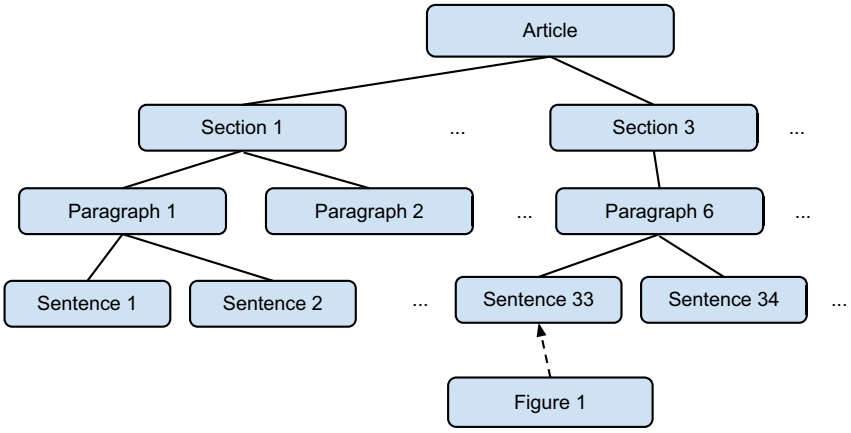


Fig. 5 The structure of the PMDoc file corresponding to the XML presented in Figure 4

2.3.1 Document Element Index

Document elements represent different components of papers that contain information describing the concepts. We store document elements at different granularity — sentence, paragraph, section and entire paper. By capturing the occurrence of synonyms of concepts in the text, we obtain an index for document elements that can track occurrences of concepts. The document index includes three fields: the document element id (e.g., article id, section id, paragraph id, sentence id, figure id, etc.), concept id, and occurrence frequency of the concept appearing in the document element. Both the document element id and concept id are indexed so that we can retrieve the occurrence frequency of a concept in a document element efficiently. Indexing at multiple granularity levels facilitates search and retrieval of content at different levels of detail.

2.3.2 Association Index

The other type of index we build is the concept association index. It records co-occurrences of concepts in the same document elements, which naturally suggests certain relations between the concepts. If we envision a graph of concepts, where concepts are vertices and an edge exists between two vertices if and only if the corresponding concepts co-occur in some document element, then our association index is equivalent to the edge list of this graph. The association index describes relations between concepts. By combining such associations, we are able to answer path queries for interrelations across multiple concepts at different levels.

The association index for two concepts can be derived from the document element index by taking intersections of document elements for the concepts. Similar to the document element index, the association index is built at multiple

granularities. If two concepts co-occur in a fine-granularity document element such as a sentence, they will also co-occur in corresponding coarser-granularity document elements such as the paragraph containing the sentence.

3 Path Mining: Discovering Sequences of Associations

Identifying relations between concepts is important in research, whether for establishing the cause of a disease in biomedicine, identifying connections between purchases in marketing/sales, or uncovering factors affecting the outcome of elections. Such problems involve concepts from multiple layers and the examination of their interactions. For example, in neuropsychiatric phenomics, instead of asking for simple and direct associations between gene and syndrome, researchers and psychiatrists would like to ask a series of questions involving intermediate factors such as the following: Which forms of behavior can be used for diagnosing syndromes? Which regions of the brain are linked to this behavior? Which genes have significant expression in these regions? These intermediate factors cut across different levels of biomedical research. Answering questions like these requires connecting relations among concepts from different levels that will form a "path". For example, the path "schizophrenia $\rightarrow$ working memory $\rightarrow$ prefrontal cortex $\rightarrow$ DRD1" describes a chain of dependencies linking a specific gene (DRD1) to effects in a region of the brain (prefrontal cortex), changes in cognitive ability (working memory), and ultimately a syndrome (schizophrenia).

A path should satisfy two types of constraints. First, the concepts in the path should follow a certain pattern according to the specific research problem. For example, in our previous example of schizophrenia, the path follows the pattern "syndrome $\rightarrow$ cognitive concepts $\rightarrow$ neuroanatomy $\rightarrow$ genes". The other constraint is that associations in the path should be strong enough to be considered relevant. The association strength of a path indicates the "interestingness" of the knowledge as well as confidence in its significance (e.g., unlikeliness of the association arising at random). It is possible that more than one path satisfies the pattern, and the strength constraints try to guarantee that only highly relevant paths are selected. Therefore, the objective of path mining is to find those paths that satisfy the pattern and strength constraints.

In the sections that follow, we present our approach to solving the *path mining* problem — including posing queries so as to specify the pattern constraint, measuring associations towards meeting the strength constraint, and discovering paths in a corpus with the use of a multilevel lexicon.

3.1 Path Mining Query

A path mining query indicates a pattern of interaction among concepts at different levels, and thus specifies a pattern of concepts in a path. Formally, a query specifies k sets of concepts $C_1, C_2, \ldots, C_k$, and denotes the set of all possible patterns $c_1 \rightarrow c_2 \rightarrow \ldots \rightarrow c_k$ satisfying the patterns where concept $c_i \in C_i$ and $i = 1, 2, \ldots, k$.

Posing a path mining query requires some prior knowledge of the field; i.e., which concepts should be specified in which levels. More importantly, the results are limited by the number of levels in the query. If the query misses a specific level of concepts, then the information in that level will not be discovered.

To address this problem, we introduced the idea of "wildcard queries" in path mining, where queries can leave multiple intermediate concept levels unspecified. If wildcard levels are used, all concept elements from the corresponding levels of the lexicon are considered. When specifying the query, the user may put the wildcard connectors between concept levels. We support three types of wildcard connectors in the query: "-" specifies no wildcard levels. "?" specifies zero or one wildcard levels, and "*" specifies arbitrarily many wildcard levels, respectively.

The query interface for the PathMining tool is presented in Figure 6. Three levels are specified in this example, where the first level includes syndromes, the second level includes neuroanatomical regions, and the third level includes genes. Users can use the drop-down menu to the right of the query concept for each level to indicate whether or not to include the subconcepts in the level. In this example, all three levels include all subconcepts of the query, and the query also includes any number of wildcard levels between level 2 (neuroanatomy) and level 3 (genes).

Fig. 6 Using the PathMining query interface to specify a path query. The radio buttons on the right are options for wildcard levels, where "-", "?" and "*" indicate no wildcard levels, zero or one wildcard levels, and any number of wildcard levels, respectively. This query searches for paths that match the pattern "syndrome $\rightarrow$ * $\rightarrow$ neuroanatomy $\rightarrow$ * $\rightarrow$ genes".

3.2 Measures for Path Strength

A path reveals the knowledge of relations among concepts in different levels. It is important to understand whether such relations exist with a high probability. On the other hand, if there are multiple paths that satisfy the query, we seek the one that is most relevant. Based on co-occurrence frequencies, we evaluate the strength of pairwise associations and evaluate relevance of the whole path.

3.2.1 Measuring the Strengths of Pairwise Associations

An association between two concepts is the simplest path. Measuring the strength of pairwise associations is the first step towards measuring the strength of a more complex path. In the context of text mining, the co-occurrence of concepts is an indicator of association. If two concepts tend to appear in the same paper, the probability is higher that these two concepts are related to each other. The association

index records the co-occurrences of pairs of concepts in the document elements. The association strength can be further computed from co-occurrence frequencies. The data mining community uses support and confidence to measure the strength of an association $A \rightarrow B$ between concepts A and B [3]:

$$support(A \rightarrow B) = \sigma(A \cap B) \tag{1}$$

$$confidence(A \rightarrow B) = \frac{\sigma(A \cap B)}{\sigma(A)} \tag{2}$$

where $\sigma(A)$ stands for the proportion of the documents in the corpus containing the concept A, and $\sigma(A \cap B)$ stands for the proportion of the documents in the corpus containing both concepts A and B.

Support measures the proportion of documents in which two concepts co-occur, and represents the probability of co-occurrence across the whole corpus. Confidence estimates the conditional probability of occurrence of B given A's occurrence. If we consider the occurrence of A and B as random events, we can also measure the strength of the association using the Pearson correlation $\rho_{A,B}$ between the two events

$$\rho_{A,B} = \frac{E(A,B) - E(A)E(B)}{\sqrt{E(A)(1-E(A))}\sqrt{E(B)(1-E(B))}} \tag{3}$$

where E(A) is the expectation of the probability of occurrence for the concept A (i.e. $\sigma(A)$). Tan et al [32] pointed out that $\rho(A,B)$ can be approximated by IS(A, B)

$$\rho_{A,B} \approx IS(A,B) = \sqrt{I(A,B) \times \sigma(A,B)} \tag{4}$$

where $I(A,B) = \frac{p(A,B)}{p(A)p(B)}$ is the interest factor [29]. The interest factor computes the ratio of the probability of co-occurrence and the expected probability of co-occurrence given that X and Y are independent of one another. The above approximation holds when $I(A,B)$ is high, and both $p(A)$ and $p(B)$ are very small, which in general fits the case of occurrences of concepts in a large text corpus. We can regard IS as an alternative interpretation of the association rule that does not indicate an inference from antecedents to consequents, but rather a measure of closeness between two concepts.

The conventional association rule mining problem is to find all associations whose strength indicators, such as support, confidence, and IS measure, are above given thresholds. Algorithms such as Apriori [3] solve the problem by generating the frequent item sets and then counting the support for the candidates in a bottom-up fashion. The FP-growth algorithm [12] solves the problem with the efficient data structure, frequent pattern tree (FP-Tree). In order to address the path mining problem, instead of finding individual associations, we need to measure the strength of a sequence of associations among the concepts in a path.

3.2.2 Measuring Strength of Associations in the Path

A path consists of a sequence of associations. In order to find paths with high association strength, we can impose a strength threshold on all the associations in the path. As with pairwise associations, each association in the path connects two concepts. However, since there are multiple associations in the path, measuring the strength of associations is more complicated. We use two approaches to measure the strength of associations: local strength and global strength.

Local Strength

The local strength measure considers the strength of individual associations as a "local" property. Each association in the path is independent of other associations and thus is only related to its direct antecedents and consequents. Therefore, the computation of association strength as a local strength measure is identical to the computation for pairwise relations (Equations (1), (2), and (4)).

Global Strength

The global strength measure considers the strength of individual associations as a "global" property of the entire path. In this setting, each association is related to the preceding associations. To compute association strength, we group all the concepts involved in previous associations in a path as the antecedent. For example, the second link of $A \rightarrow B \rightarrow C \rightarrow D$ would be regarded as $AB \rightarrow C$.

In this case, the measurement of support and confidence differs from simple pairwise association mining. Specifically, support of the second link of $A \rightarrow B \rightarrow C \rightarrow D$ is $\sigma(A \cap B \cap C)$, and the confidence can be computed as

$$Confidence(AB \rightarrow C) = \frac{\sigma(A \cap B \cap C)}{\sigma(A \cap B)} \tag{5}$$

With this definition, the confidence is the conditional probability that C is part of the path given that $A \rightarrow B$ is part of the path.

The correlation measure of the link can be derived by computing the correlation between two random events: the co-occurrence of all previous antecedents as one random event and the occurrence of the consequent as the other. According to this definition, the correlation score of the second link of $A \rightarrow B \rightarrow C \rightarrow D$ can be computed as $IS(AB, C)$.

Comparison of Local and Global Strength

Figure 7 presents an example path measured by the two different approaches. The support, confidence and IS measure are computed using local strength measure and global strength measure in Figure 7(a) and Figure 7(b), respectively. For the global strength measure, since the association takes all preceding concepts as an antecedent, the support value of the association decreases when the path length increases, and confidence and IS scores change correspondingly. This property makes

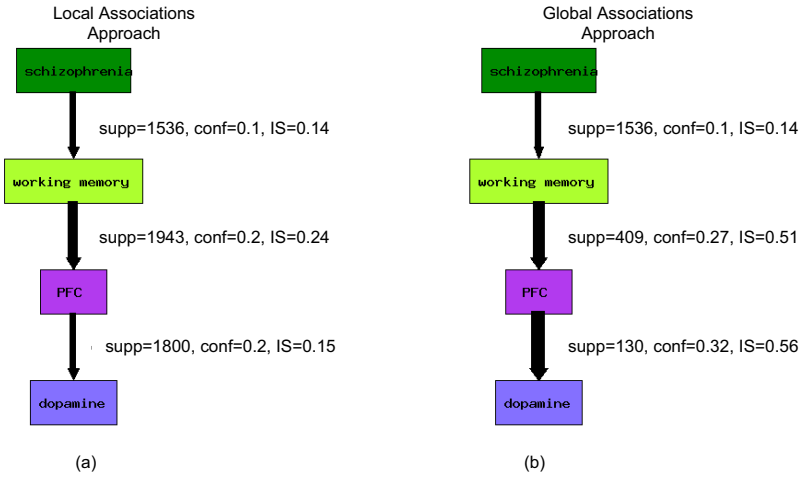


Fig. 7 Two different approaches for measuring the strength of associations for the path "schizophrenia → working memory → PFC → dopamine": (a) the local strength measure, (b) the global strength measure. The thickness of the links in the path is proportional to the IS score of the corresponding associations.

it more difficult to find a high-support path when more concepts are included in the path.

The major difference between the two approaches lies in the different requirements for co-occurrence of concepts in the paths. In the global approach, all the concepts are required to appear at least once in the same document element in order to ensure a non-zero confidence. On the other hand, the local approach only requires adjacent concepts in the path to appear in the same document elements. This difference leads to two different types of applications for path discovery. For the global approach, since there is at least one item of literature that explicitly discusses all the concepts in the paths, path mining reveals existing investigations in the literature covering the path. The local approach forms a path by stitching high-strength pairwise associations together. The paths discovered with the local approach have not necessarily been studied previously in literature, but may have a good potential for future study since each pair of concepts in the paths is well related. Therefore, the local approach can be applied to scenarios focusing on discovery of new paths and generating new hypotheses.

3.3 Path Mining Algorithms

Based on the choice of the association strength measurement, the path mining problem can be transformed into two different problems and solved by corresponding algorithms. When using the local strength measure, the path mining problem is

equivalent to a graph search problem. For the global approach, the path mining problem can be viewed as an extension of traditional association rule mining.

3.3.1 Path Mining as a Graph Search Problem

For the local strength measure, the path discovery process is that of finding strongly connected pairwise associations across the levels specified in the path query. We can construct a graph of concepts whose edges are these associations. Then the path mining problem is equivalent to a graph search problem which finds paths in the graph that satisfy the path query, and the strengths of associations meet the desired threshold.

We can use the graph traversal algorithms such as depth-first search or breadth-first search to examine the candidate associations. For example, assume concepts $a_1, a_2, \ldots, a_m$ are in level 1, concepts $b_1, b_2, \ldots, b_n$ are in level 2, and concepts $c_1, c_2, \ldots, c_l$ are in level 3. Then we can first draw an edge $a_1 \rightarrow b_1 \rightarrow c_1$, then draw an edge $a_1 \rightarrow b_1 \rightarrow c_2$, and so on. We can then pick the paths that meet the thresholds of association strength for measures such as support, confidence, and *IS*. In the case of answering wildcard queries, we will add wildcard levels between each level. The complexity of the computation can be $O(b^k)$ where b is the number of concepts in the level and k is the number of levels involved (including wildcard levels). This process can be computationally expensive when b or k is large. However, this high computational cost can be reduced significantly by introducing pruning steps when traversing the graph. According to our definition of association strength measures, computation of the strength of an association in a path is only affected by preceding associations (e.g., in $A \rightarrow B \rightarrow C \rightarrow D$, computation for strength of $B \rightarrow C$ would be affected by the strength of $A \rightarrow B$, but not by $C \rightarrow D$). For the local strength measure, since all the associations are independent of each other, the measurements of path strengths are also independent. For the global strength measure, the computation of strength of association only considers all the preceding concepts and the concepts that directly following. When a new association is added to the path, the strength of existing associations is fixed. Therefore, in either of the strength-measuring approaches, if a link fails to meet the strength constraint, we can drop the link and all the possible paths containing the link. Although the worst-case time complexity is not reduced by this pruning process, in practice the computation time is largely reduced.

3.3.2 Path Mining as Extension of Association Rule Mining

When using the global approach to measure the strength of association, we can also use existing association rule mining algorithms to solve the problem. The Apriori algorithm [3] generates the frequent patterns in a bottom-up fashion. As we discussed in Section 3.2.2, when taking the global approach, all the concepts in the path must appear in at least one document element, which is equivalent to the frequent pattern in the Apriori algorithm. The support measure for the association path monotonically decreases as path length increases. Thus the path will not be extended if one of

its associations fails to meet the threshold. This property makes the path equivalent to the frequent item sets in the Apriori algorithm.

The difference between path mining and traditional association rule mining is that a path has more than one association involved, and we need to check and maintain the strengths of all the associations in the path (such as confidence and *IS*). Although path mining provides more information, the computation cost is the same as traditional association rule mining. According to the definition, the computation of confidence and correlation is only affected by preceding links in the paths. Therefore, as the path grows, we only need to compute the strength of newly added links, which makes the complexity equivalent to conventional association rule mining using the Apriori algorithm.

3.3.3 Ranking Path Relevance via Association Strength

When multiple paths exist, one goal is to determine which path is most relevant to the query. The association strength can be a good indicator of path relevance. So far, however, all the measurements focus on individual associations in the path. We still lack a uniform measure that we can use to evaluate the relevance of the entire path.

In our algorithms we take a heuristic approach to rank the paths by comparing the "weakest link" of each path. For example, a path $A \rightarrow (0.4)B \rightarrow (0.4)C$ would be ranked higher than a path $A' \rightarrow (0.6)B' \rightarrow (0.3)C'$ (numbers in the parentheses indicate the strength of the link) because the weakest link in the first path (0.4) is stronger than the weakest link in the second one (0.3). This approach guarantees that higher ranked paths have reasonably high strength in all the links. In addition, this ranking approach can be exploited for pruning in the path discovery process. Consider the case for finding the top K paths satisfying a query. For any path containing a link whose strength is weaker than the weakest links of K existing paths, then no paths involving the link could be included in the result, and thus the path can be pruned. If we sort the associations by their strength before performing the search, then we can prune all the links with lower strength as well.

3.3.4 Interpreting Path Mining Results

After a path query is issued, a list of relevant paths are returned. Figure 8 presents an example of the results. By clicking the "detail" link, we are able to examine the strength of the associations in the paths. Figure 7 shows one of the paths returned with a four-layer query involving schizophrenia, working memory, neuroanatomical concepts, and dopamine. In this example, PFC (prefrontal cortex) is a neuroanatomical concept.

Path mining results return all paths that satisfy the pattern and strength constraints. The aggregation of the paths form a graph called a PhenoGraph, which visually summarizes the associations among concepts in different levels. The PhenoGraph is useful for further analysis involving multiple paths. In real applications, a single path is often insufficient to provide a perfect solution to the problem. Instead,

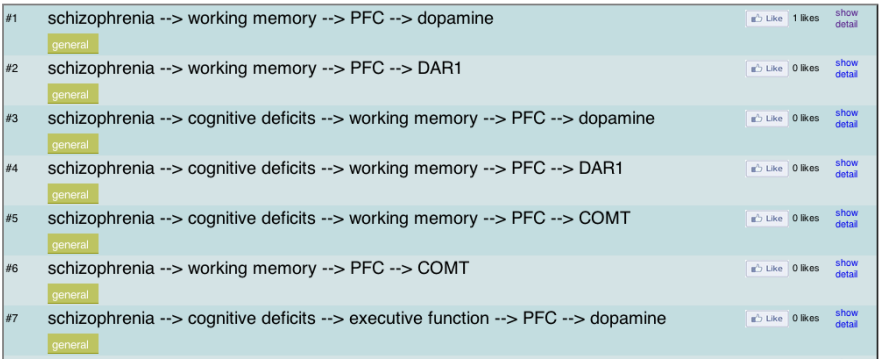


Fig. 8 The seven most relevant paths for path query "syndrome → * → cognitive concept → * → genes" (as specified in Figure 6). Based on the lexicon, schizophrenia is a syndrome-level concept, working memory and executive function are cognitive-level concepts, dopamine, and DAR1 and COMT are gene-level concepts. These concepts match the pattern specified explicitly in the path query. Meanwhile, Cognitive deficits and PFC are symptom-level and neuroanatomy-level concepts, which are introduced as wildcard levels in the paths. As a result, the returned paths have four or five concepts where one or two among them are wildcard concepts.

a consensus of multiple paths can provide a more complete picture. Figure 9 presents a graph structure aggregated by top paths returned in Figure 8.

4 Path Content Retrieval

In the previous section we discussed our approach to identifying the paths and measuring path relevancy via association strengths. After obtaining a list of paths, the next challenge is to study the path in order to obtain more detailed knowledge about the interactions among the concepts. Since our path knowledge discovery is derived from text mining over a corpus of scientific literature, the relevant content from the literature would be useful in presenting the details of path knowledge. We refer to relevant document elements with knowledge about a path as "path content" and the process of searching for path content as "path content retrieval."

Compared to traditional information retrieval, path content retrieval has many new challenges. First, we need to translate a path to a query so that it is digestible for an information retrieval system to find the relevant content describing relations between concepts. Second, the retrieved content should be in fine granularity so that specific information about the relations can be revealed. Third, according to the different demands of the biomedical field, specific types of results are demanded in path content retrieval — such as quantitative experimental results or experiment sample characteristics.

We developed a content retrieval tool called Document Content Explorer to handle these problems. Figure 10 presents the process of path content retrieval. First,

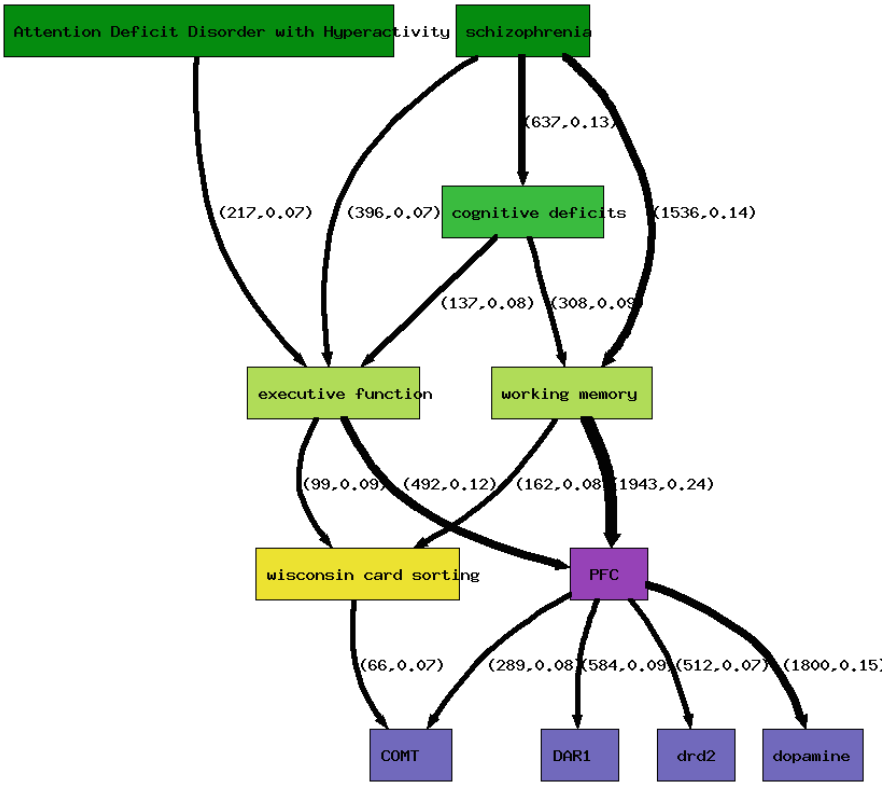


Fig. 9 A PhenoGraph can be generated by combining the paths returned from a path query. The above PhenoGraph is generated for the path query "syndrome * → cognitive concept * → genes", as shown in Figure 6. In the PhenoGraph, Attention Deficit Disorder with Hyperactivity and Schizophrenia are syndrome concepts, executive function and working memory are cognitive concepts, and COMT, DAR1, DRD2 and Dopamine are gene/signaling pathways concepts. In addition, cognitive deficits, Wisconsin card sorting and PFC are symptom-level, task-level and neuroanatomy-level concepts respectively. These three concepts were inserted into the paths as wildcard intermediate levels.

the path is converted to a query for document retrieval. Then, based on the document element index, contents at various granularities are matched to the query and the most relevant ones are returned to users. Finally, the results are classified and presented to the user for further analysis. The Document Content Explorer works in concert with path mining to conduct path knowledge discovery. For each path result returned from the PathMining tool, a user can use the "Retrieve Relevant Content" link to connect to the Document Content Explorer and retrieve path content. In this section we will present our approach to completing these tasks. The preprocessing, content matching and post-processing of retrieved content for paths are discussed in Sections 4.1, 4.2 and 4.3, respectively.

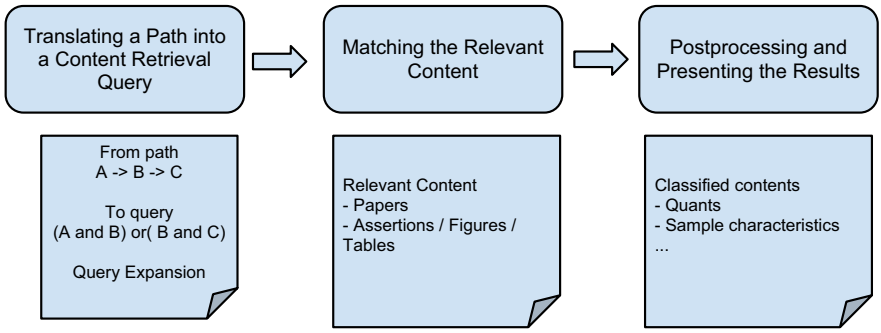


Fig. 10 The process flow of path content retrieval

4.1 Query Processing for Path Content Retrieval

The first step of path content retrieval is to create a proper query. Since path knowledge presents relations between concepts at different levels, the goal of path content query is to retrieve the content describing these relations. According to this goal, the path is translated into a content retrieval query by applying Boolean operators (e.g., AND, OR) among concepts to reveal the relations. For example, a path "schizophrenia → working memory → PFC → dopamine" will be translated to a query "(schizophrenia AND working memory) OR (working memory AND PFC) OR (PFC AND dopamine)." With the translation, the query will retrieve all the documents relevant to at least one of the associations in the path.

The query can be further refined with the query interface of the Document Content Explorer tool. If the "match all concepts" option is selected, the query will be transferred into "schizophrenia AND working memory AND PFC AND dopamine," which then retrieves the documents covering all concepts in the path.

Lexicon-Based Query Expansion

Queries in the Document Content Explorer are concept-based. Each query word is translated to a concept and matches all concept synonyms. This approach helps match more relevant content. However, in some situations, simple synonym-based expansion is inadequate. Fortunately, using the lexicon hierarchy, we may be able to further expand the query.

As described in Section 2.1, concepts are organized in a hierarchical structure. From the structure, we will be able to obtain subconcepts for a given concept. For example, "DRD1", "DRD2" and "D5-like" are sub-concepts of "dopamine receptors". The hierarchical structure is strictly defined by an "is-a" relation. Users searching for content about dopamine receptors could also be interested in content about specific types of receptors. We can use the hierarchy to rewrite the query to search this larger scope, and this may obtain better results. For general queries, we can expand the original query by including subconcepts. For instance, the query "dopamine

receptors" might be expanded into "dopamine receptors OR DRD1 OR DRD2 OR D5-like" so that more content can be retrieved. The expansion may continue to still deeper levels if the expanded concepts also have subconcepts. The Document Content Explorer provides an option for specifying the depth of expansion.

4.2 Finding Relevant Path Content

After translating a path to a query for content retrieval, standard information retrieval methods can be employed to return the most relevant content. In path content retrieval, we want to retrieve document elements that include relations between concepts, such as sentences describing such a relationship, tables presenting experimental results explaining the correlation, or figures illustrating the interactions between concepts.

As described in Section 2.3, document elements and queries can be represented as vectors, and we can apply the vector space model (VSM) [26] to match relevant content. There are different ranking functions available for measuring relevancy of the content. In the PhenoMining system, we implemented the BM25 function based on the probabilistic relevance framework [23], which can be viewed as a variant of the standard TF-IDF weighting scheme [27].

Path content is more likely to describe relations between concepts when it is specific, with fine-granularity elements such as sentences, figures and tables. As in Figure 5, the document elements for each paper are organized in a hierarchical structure, which enables retrieval of contents in different granularity. However, due to the short length of fine-granularity content, the number of hits of concepts in fine-granularity document elements is usually very low (most fine-granularity concepts hit the query only once or twice) and these elements tend to be similar (many document elements have the same number of hits). Therefore, it is difficult to rank fine-granularity elements. To remedy the problem, we take a two-step approach to finding relevant path content. We first rank coarse-granularity content such as papers, then for each coarse-granularity element, we match fine-granularity content such as sentences and figures and display these to users. Since fine-granularity content is included within coarse-granularity content (e.g., a sentence is part of a paragraph, section and so on), highly ranked coarse-granularity content most likely also contains relevant fine-granularity content. Such a two-step ranking schema enables users to find the most relevant content.

Figure 11 presents the interface for the first step in content matching. Given a query, a list of relevant papers is returned. For each paper, basic information such as title, authors, journal, and published date are displayed, along with basic occurrence statistics. By selecting each paper, its fine-granularity content is displayed in the document details panel as shown in Figure 12. Based on automated results classification (Section 4.3), extracted content is also displayed in separate tabs. In each tab, results are broken down by sections and kept in the same order as in original document, so that users can read them just as when reading the original paper. The

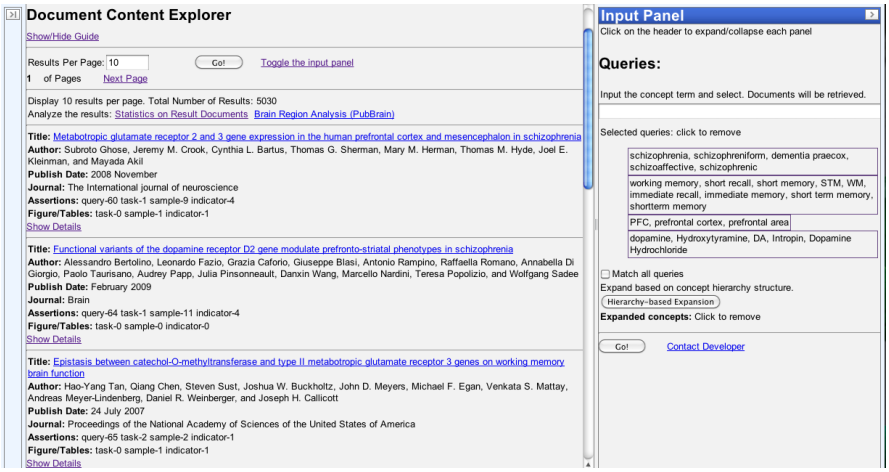


Fig. 11 User interface of the Document Content Explorer. User specifies the query in the "input panel" on the right and the relevant papers are displayed in the "results panel" on the left. The query shown includes four concepts: schizophrenia, working memory, prefrontal cortex and dopamine, translated from the path "schizophrenia → working memory → PFC → dopamine".

detailed view of a paper provides a quick summary that permits users to quickly grasp the relevance of the results.

4.3 Results Classification for Specific Research Goals

Based on different research goals, users may be interested in different types of content. For example, in phenomics, researchers are typically interested in quantitative experimental results (phenotype measurements) and experiment descriptions such as sample characteristics. To satisfy the demand of different users, we further classify our results and filter them according to different research goals. The results are classified using the category information from the concept lexicon. For each document element, we create a histogram vector by aggregating the count of concepts for different concept categories. The histogram of concept categories can be viewed as a feature that indicates the focus of the document element. In our lexicon, there are several special categories introduced for content classification, such as sample characteristics, indicators, and sample species. We classify the content based on the majority category of its concepts. For example, in the sentence, "We tested WM [working memory] in infants at 6.5 and 9 months of age in a task that challenged them to remember the location of social and non-social targets." WM is a cognitive concept, and infants and months of age are concepts related to sample characteristics. In this case, the majority of the concepts are in the sample characteristics category, so the content is classified as sample characteristics.

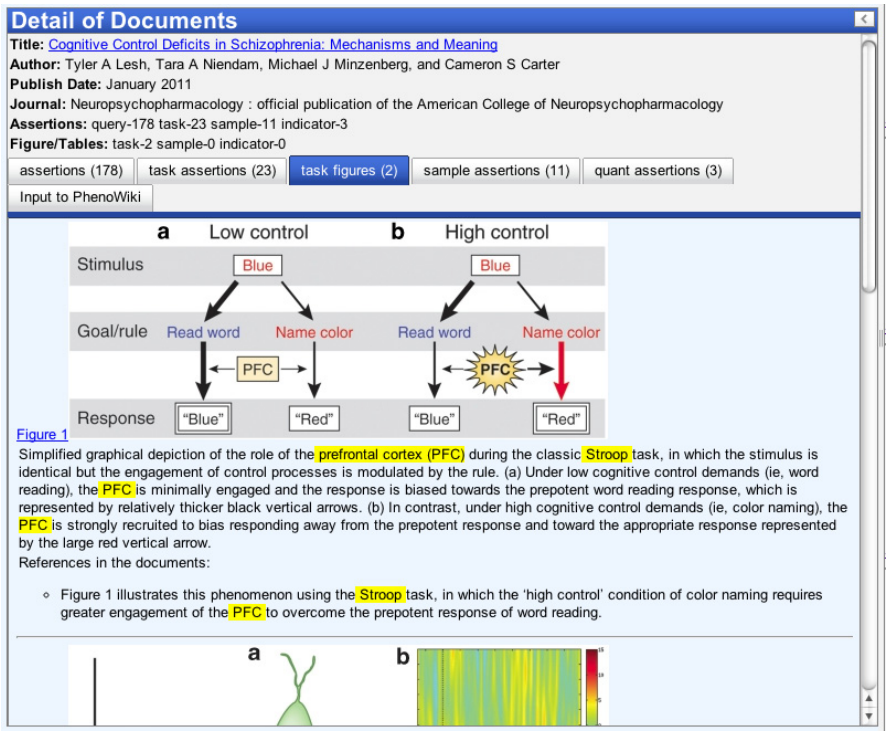


Fig. 12 Detailed view of Document Content Explorer. The paper "Cognitive control deficits in schizophrenia: mechanisms and meaning" [16] has been retrieved with the path query "schizophrenia → working memory → PFC → dopamine". Fine-granularity content in the paper is classified into different categories such as task description, sample characteristics and quantitative results. Content of different categories is presented in different tabs. The selected tab (task figures) shows figures in the paper that include task descriptions. Both the task descriptor keywords ("Stroop") and the query keywords ("prefrontal cortex, PFC") are highlighted.

In our PhenoMining tool, we classify results into three categories, "task description," "sample characteristics" and "quantitative indicators" to facilitate different research purposes. If no concepts corresponding to these categories are found in a document element, it is classified as "general content." With proper training data, it is possible to extend such simple rule-based classifications to machine-learning-based classifiers to improve the accuracy of classification.

In the document detailed view of Document Content Explorer (Figure 12), we can observe that the results are classified into different categories and are displayed by the corresponding tabs to permit users to choose content of interest. In the list of papers returned for a query, the numbers of results classified into different categories in the paper are also presented; this helps users select the papers relevant to their interests.

Table 1 Path content for the path "working memory → PFC → D1 Receptors" discovered by the Document Content Explorer. The discovered content describes relationships among the concepts working memory (WM), D1 receptors, and prefrontal cortex (PFC), which are highlighted in the extracted assertions.

Paper Title	Assertions
A role for prefrontal calcium-sensitive protein phosphatase and kinase activities in working memory [24]	· Within the **PFC**, depletion of dopamine or inhibition of dopamine **D1**/D5 receptor activity results in an impairment in **working memory** tasks (Brozoski et al. 1979; Kozlov et al. 2001) · local infusion of a dopamine **D1**/D5 receptor antagonist into the **PFC** interferes with delay period activity and **working memory** (Sawaguchi and Goldman-Rakic 1991; Sawaguchi 2001) · This dose was based on previous studies demonstrating that intra-mPFC infusion impairs working memory and disrupts **D1 receptor** activity (Seamans et al. 1998)
Dopamine D1 and D5 Receptors Are Localized to Discrete Populations of Interneurons in Primate Prefrontal Cortex [11]	· **Working memory (WM)** is a core cognitive process that depends upon activation of **D1 family receptors** and inhibitory interneurons in the prefrontal cortex. · Dopamine activation of **D1 family receptors** in the **prefrontal cortex (PFC)** regulates **PFC** functions, especially **working memory (WM)** (Brozoski et al. 1979; Sawaguchi and Goldman-Rakic 1991; Muller et al. 1998).
Differential Contributions of Dopaminergic D1-like and D2-like Receptors to Cognitive Function in Rhesus Monkeys[34]	· Spatial **working memory** accuracy was reduced to a greater extent by raclopride than by SCH which was unexpected, given prior reports on the involvement of **D1** signaling for spatial working memory in monkeys. · The oculo-motor version of the spatial delayed response task has provided evidence for an optimal range of **D1** signaling in primate **prefrontal cortex** for spatial **working memory** (Williams and Goldman-Rakic 1995).
Dopamine D1/D5 receptor modulation of excitatory synaptic inputs to layer V prefrontal cortex neurons [28]	· Dopamine acts mainly through the **D1/D5 receptor** in the **prefrontal cortex (PFC)** to modulate neural activity and behaviors associated with **working memory**. · Dopamine levels are elevated in the PFC during performance of **working memory** tasks (1), and task performance is generally modulated by the **D1**, but not D2, class of dopamine receptors (2–5). · Therefore, understanding how **D1 receptor** activation affects synaptic responses in **PFC** neurons is critical for understanding the functional neuromodulation of sustained activity patterns underlying **working memory** processes within the PFC.

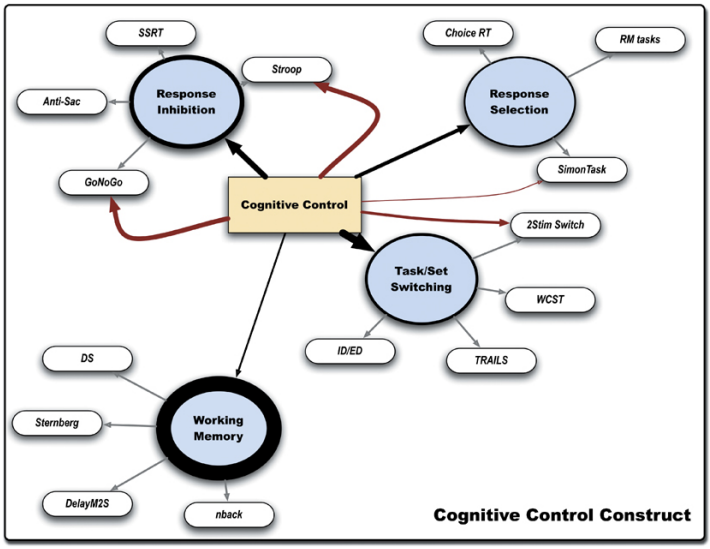


Fig. 13 Components of the construct "cognitive control". This figure from [25] displays a graphical representation of the construct "cognitive control" as defined by the literature and expert review of behavioral tasks.

4.4 An Example of Path Content Retrieval

By compiling results from Document Content Explorer, a sample of path content derived for the path "Working Memory → PFC → Dopamine Receptors" is presented in Table 1. Coarse-granularity content (papers) is first selected from search results (Figure 11) and fine-granularity content (assertions) is extracted from the selected paper (Figure 12). In this example, the retrieved assertions describe relations among working memory (WM), D1 receptors (gene DRD1), and prefrontal cortex (PFC).

5 A Sample Application of Path Knowledge Discovery

In this section we will present an example of using PhenoMining tools for knowledge discovery in phenomics. More specifically, we plan to answer the question of heritability for cognitive control phenotypes which was previously presented in [25].

"Cognitive control" is a complex process that involves different phenotype components. Deficits in cognitive control are apparent in many neuropsychiatric disorders with strong genetic components. Different behavioral tasks are used for measuring the performance of those components with specific indicators. Knowing whether these components are also under strong genetic control is important for neuropsychiatric research. As an example, "working memory" is a latent component of cognitive control associated with schizophrenia and bipolar disorder [25]. "N-back test" is a behavioral test measuring a person's working memory

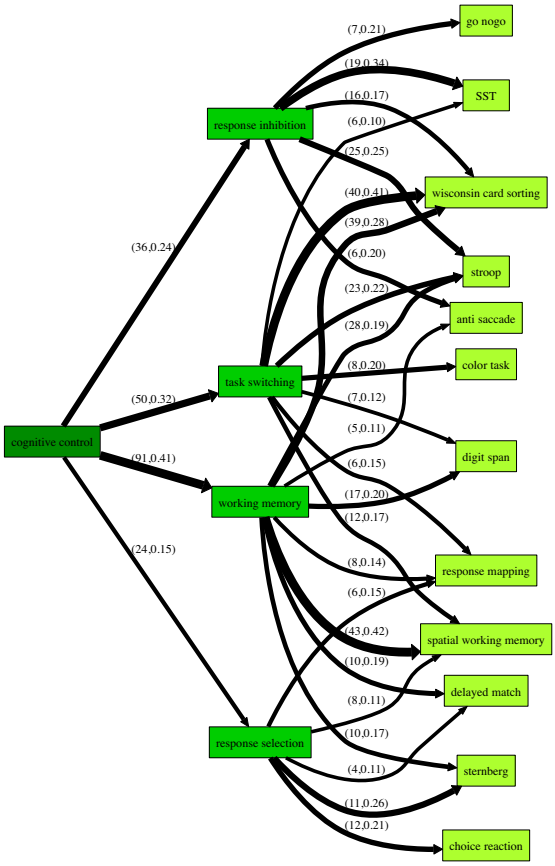


Fig. 14 A PhenoGraph generated from path query "cognitive control → subprocesses → cognitive tests". The strength of associations are computed based on the local strength measure. The numbers next to the links in the graph show the support (in absolute value of co-occurrence) and correlation scores of associations represented by the corresponding link. The thickness of links is proportional to their correlation score.

performance. One important indicator for the n-back test is accuracy. The heritability of cognitive control is associated with the heritability of the indicators of behavioral tasks (e.g., the heritability for accuracy in the n-back test). Formalizing the nature of cognitive control requires studying relations among cognitive control, its subprocesses, and phenotypes such as heritability scores and indicators of behavioral tasks. This can be viewed as a path knowledge discovery problem. With the pattern "cognitive control → subprocess → task → indicator" we can gather known results about cognitive control. The results of path knowledge discovery provide a basis for interdisciplinary analysis of the heritability of cognitive control.

Table 2 Subprocesses and their corresponding cognitive tests. The matching is based on the correlation score of the associations between subprocesses and cognitive tasks. The association with the highest correlation score for each task is selected. The name in the parenthesis are the names of the tasks appeared in [25].

Latent Subprocess	Cognitive Tests
Response inhibition	GoNoGo, SST (SSRT), Stroop, Anti Saccade (Anti-Sac)
Task switching	Wisconsin Card Sorting (WCST), Color Task, ID/ED
Working Memory	digit span (DS), spatial working memory, Delayed Match (delayed M2S)
Response Selection	Response Mapping (RM Tasks), Sternberg, Choice Reaction (Choice RT)

Therefore, we can view the problem of path knowledge discovery as a three-step process. First, we complete a query schema to operationally define the construct of cognitive control by identifying candidate components, tasks, and indicators that exist in the literature (such as those in Figure 13). Then, we use path queries to obtain quantitative heritability results for the task indicators in the corpus. Finally, we explore this content and extract discoveries about the heritability of cognitive control.

Such an analysis requires effort, even from domain experts. For instance, as reported in [25], although papers can be searched and retrieved using PubMed, domain experts still have to perform a deeper analysis, such as operationally defining the construct of cognitive control, digesting the content of papers to find the heritability scores, and conducting further analysis — which can be exceedingly time-prohibitive, especially as the literature base grows. Utilizing our path knowledge discovery tools, findings that relate cognitive control, its subprocesses, behavioral tasks, and indicators can be compiled automatically. Further, the path content, including the heritability score, can be retrieved for users. The actual derivation of results concerning the heritability of cognitive control cannot be automated, as the selection of which evidence to include/exclude requires domain expertise. By employing our tools, however, the whole process can be greatly accelerated.

Prior to performing path knowledge discovery, we first created a lexicon from the concepts covered in PhenoWiki [25]. The lexicon consisted of concepts from four levels — latent complex constructs, latent processes, cognitive tasks and indicators.

Figure 14 shows the result of top paths returned by a path query of three layers that included cognitive control as the first layer, concepts in latent processes as the second layer, and cognitive tasks as the third layer. This PhenoGraph demonstrates the matching between tasks and the subprocesses of cognitive control that the tasks measure. In the figure, each task has relations with multiple subprocesses. For clarity in the presentation, we choose a higher threshold to generate the PhenoGraph. Table 2 presents the results obtained by choosing the most highly correlated subprocesses for each task, which are derived using a lower threshold and thus match more tasks (e.g., ID/ED task). Compared to Figure 13 in [25], which was created by domain experts, the mining tool achieved very promising results; 12 out of 15 tasks are correctly associated with their corresponding subprocesses. Compared to the results from domain experts (as shown in Figure 13), Figure 14 includes extra

links since some tasks match multiple subprocesses. False positives exist because the co-occurrence of tasks and subprocesses in a document element (using paragraph granularity in this example) does not necessarily indicate that the subprocesses are measured by the task. Also, it is entirely possible that two subprocesses are discussed in the same document element, and our system is unable to separate them. On the other hand, some tasks are not included in the top paths because the occurrences of those tasks in the corpus is so low that the correlation with subprocesses is too low to be included in the results. By setting the threshold lower, the missing tasks may appear but may also introduce more noise. Choosing the proper threshold to trade off precision and recall would be a decision for users to make. Overall, using our proposed tools, the time spent on collecting the relevant literature and deriving the knowledge structure is greatly reduced, and the results from the tools are comparable to human-derived results.

Figure 15 extends the query to indicator-level concepts, which gives us the complete structure of the cognitive control construct. From the figure, we can observe the highly correlated indicators for the tasks. For clarity of presentation, we only present the top paths in Figure 15. By lowering the association strength threshold, a more complete compilation of findings on cognitive control can be obtained.

After retrieving the paths, the next step is to find the heritability values for indicators. To retrieve information about heritability, we add a "heritability" concept to the query in the Document Content Explorer. Table 3 shows some sample results of relevant path content, including assertions, figures, and tables containing the heritability result. These results are analyzed by domain experts to extract discoveries concerning the heritability of cognitive control.

Compared to traditional approaches, which require a significant amount of manual work by domain experts, our approach provides a much more efficient way to find paths that match the tasks and indicators to subprocesses, as well as extract heritability scores from the relevant literature. Our experience is that the mining results are comparable to human-generated results. It takes seconds to retrieve the content and a few minutes for a user to browse and select the relevant content. The traditional manual approach may take several orders of magnitude longer to execute the same steps and becomes infeasible when the number of papers to examine becomes large. This typically results in severe reductionist approaches by domain experts when trying to identify a significant but manageable subset of the literature. Our tools eliminate the need for drastic a priori approaches to reduce the scope of literature for review. Thus, with the aid of mining tools, the scope of research can be enlarged into a corpus of thousands of papers instead of the 150 papers used in [25]. Our tool greatly improves the scalability of such a complex analysis. Meanwhile, human intelligence still plays an important role in the process; selecting the best paths and the best content are quite subjective and different users may use the results differently. It is unrealistic to automate the entire research process, but it is clearly beneficial to use text mining and information retrieval techniques to replace the mechanical aspects and speed up the process.

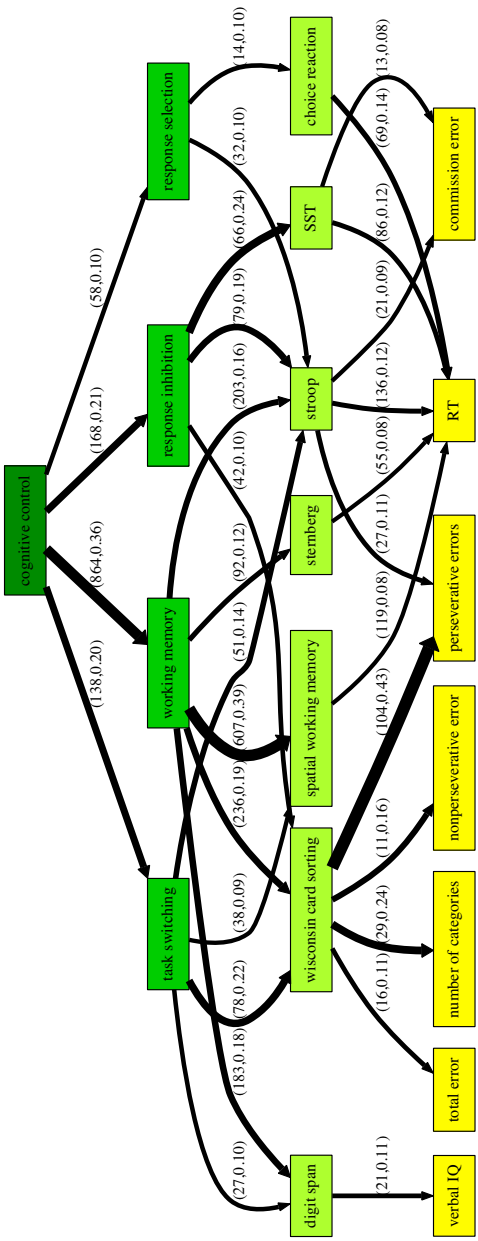


Fig. 15 The phenograph generated from path query "cognitive control → subprocesses → cognitive tests → indicators". The strengths of associations are measured with local strength measure. The numbers on the links in the graph indicates the support (in absolute value of co-occurrence) and correlation score of the association presented by the corresponding link. The thickness of the links are proportional to the correlation score.

Table 3 Samples of path content that contains heritability data extracted for paths with selected task/indicators

Task / Indicator	Heritability	Paper	Relevant Content
Stroop / RT	0.5	Heritability of Stroop and flanker performance in 12-year old children [31]	In the Stroop task we found high heritabilities of overall reaction time and - more important - Stroop inference (h2 = nearly 50%).
Forward Digit Span Backward Digit Span	0.542 ± (0.08) 0.475 ± (0.09)	A Multimodal Assessment of the Genetic Control over Working Memory [14]	Table 2, working memory, gray matter and white matter tract measures.
Forward Digit Span Backward Digit Span	0.22 CI(0.15 -0.65) 0.50 CI(0.41-0.58	Storage and Executive Components of Working Memory: Integerating Cognitive Psychology and Behavior Genetics in the Study of Aging [15]	In this model, heritabilities were 0.22 (CI = 0.150.65) for digits forward, 0.45 (CI = 0.130.54) for add-3 trials, and 0.50 (CI = 0.410.58) for add-4 trials (see Figure 2).
WCST/ perseverative errors		Developmental and genetic influences on prefrontal function in adolescents: a longitudinal twin study of WCST performance [4]	Table 2 heritability of WCST performance
Sternberg	0.38 (pos. trials) 0.32 (neg. trials)	Individual Differences in Processing Speed and Working Memory Speed as Assessed with the Sternberg Memory Scanning Task [33]	Heritability was 38% for positive and 32% for negative trials.

6 PhenoWiki+: Integrating Data Mining with Knowledge Base

6.1 PhenoWiki: A Collaborative Knowledge Base for Cognitive Phenomics

Compared to the rapidly developed genomic-level knowledge bases stimulated by the success of the human genome project, higher-level knowledge bases that include cognitive and behavioral data are relatively underdeveloped. In particular, in the realm of neuropsychiatric disease, few databases have been developed to maintain knowledge about phenotypes at the syndrome, symptom, cognitive, and neural systems level. PhenoWiki[25] is a collaborative annotation database that enables representation and sharing of empirical information about phenotypes for neuropsychiatric research.

PhenoWiki stores the empirical results extracted from papers by domain experts and provides references for cognitive tasks and results. Such a database is very useful in that it enables investigators to select and prioritize the endophenotypes.

However, constructing such a knowledge base can be difficult, and it faces two three roadblocks. First, populating the knowledge base requires a large amount of manual work by human experts, making it expensive and also introducing noise. Second, multiple levels of phenotypes make linking different knowledge in the

knowledge base very difficult. Third, even when each phenotype concept has its entry in the knowledge base, summarizing related knowledge for concepts is a manual process.

To resolve the shortcomings of the preexisting PhenoWiki system, we developed the PhenoWiki+ system for integrating our more advanced and automated mining techniques in order to more efficiently construct and manage a large repository of phenotype knowledge. Taking advantage of the knowledge discovery abilities of PhenoMining tools , we are able to build the knowledge base content faster and on a larger scale. The PhenoWiki+ system is implemented using the Resource Description Framework (RDF) data model, which enables the storage and retrieval of knowledge with the relationship information preserved, connects knowledge of different concepts to complete the knowledge structure, and integrates with external knowledge sources. Furthermore, by incorporating the annotations generated by users, the knowledge quality can be further improved and will ease the management of the knowledge base.

6.2 Populating the Knowledge Base with Mining Results

Populating a knowledge base like PhenoWiki requires enormous effort by domain experts, making the process of constructing the database expensive and time-consuming. Using our path knowledge discovery tools, we can accelerate the process and improve the usefulness of such a knowledge base. Path knowledge includes two integral parts: the associations among concepts at different levels and the content describing such interrelationships. Both parts of the path knowledge are useful in the construction of the database. The relevant content can be populated to the knowledge base (e.g., the heritability of certain phenotypes).

By using the automatic knowledge discovery with PhenoMining tools, the knowledge acquisition process can be accelerated and can be scaled to a large corpus. In PhenoWiki+, users input the data into a knowledge base with assistance from automatic mining tools. For example, Figure 16 shows the interface for adding quantitative findings from an empirical study. When the user is editing the quantitative data (quants), relevant assertions, figures or tables are extracted by the Document Content Explorer and are provided as suggested candidates. Using the content retrieval technique provided by Document Content Explorer, users are able to find the information faster and more easily, and can populate the knowledge base more conveniently.

The quantitative data discovered by the mining tools are stored with a specific data model (Figure 17). Even complex data models can be represented by Resource Description Framework (RDF). The RDF data adopts a subject::predicates::object model, known as “triples,” to represent different entities and relationship among them. In our implementation, each entity has its own identifier (concept ids for concepts, document element ids for documents, and identifiers for quant data and sample groups). The relations between entities are defined using triples that include the identifiers. For instance, a quantitative finding (id: quant1) about an experimental

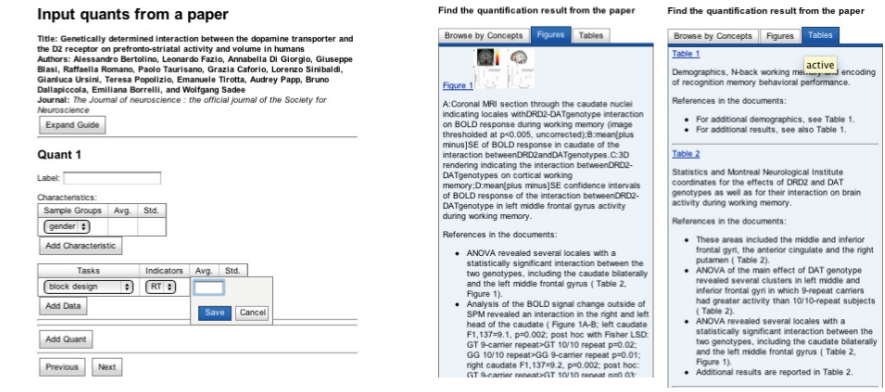


Fig. 16 User interface for inputting quantitative experimental results in PhenoWiki+. On the left, users can specify the characteristics of the sample group in the experiment, the task and indicators used and the quantitative results. Using our mining results, we are able to make suggestions to the user with content extracted from the paper. The two screen shots on the right present the content extracted from the paper.

2sec reaction time in an n-back test can be specified as quant1 :: has-task :: ⟨ nback ⟩, quant1:: measured-by :: ⟨ reaction time ⟩ and quant1 :: data :: "2 seconds," where nback and reaction time are concepts in the lexicon, ⟨ nback ⟩ and ⟨ reaction time ⟩ are the concept ids of concepts "nback" and "reaction time" respectively. Each piece of quantitative data is linked with a sample group. One sample group can be shared by multiple quantitative data when there are multiple experiments performed on the sample. A sample group can have multiple sample characteristics such as age, gender, etc., which can also be conveniently represented with RDF. Each piece of quantitative data is also linked with the document elements from which it was obtained.

Query languages such as SPARQL [22] enable users to query an RDF data store with patterns of triples. Quantitative data can be queried with different criteria, such as the tasks or indicators used in the experiments, range limits on experimental results, or certain sample characteristics. In our implementation of PhenoWiki+, the quant search functionality enables users to search quantitative results by tasks, indicators, and sample characteristics.

6.3 Connecting Knowledge with Paths Using the RDF Data Model

Path knowledge reveals relations among concepts, which is useful for connecting different knowledge in the knowledge base. For example, when constructing a Wiki page for a concept such as "working memory," knowledge about which genes have been linked with working memory ability is relevant. The paths generated by PhenoMining tools provide this knowledge. In the implementation of the PhenoWiki+

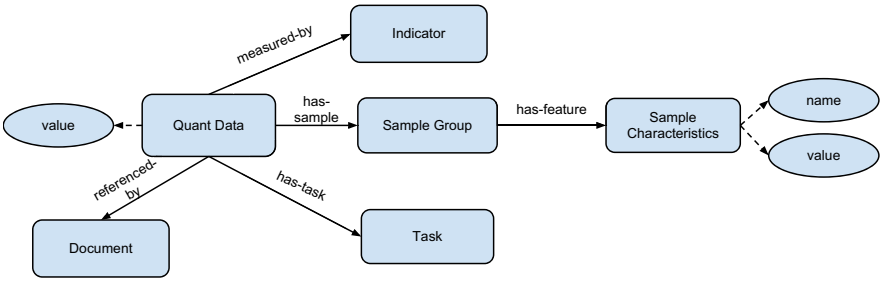


Fig. 17 Data model for quantitative data. The squares represent the entities in the knowledge base. (each entity is specified with identifiers) The arrows represent the relationships defined among different types of entities. The ellipses linked by dashed arrows represent the attributes of specified entities. All the data are represented with RDF triples with the pattern "entity1 :: relation :: entity2". Using an RDF database supporting SPARQL queries can efficiently retrieve quantitative data by various criteria such as indicators, tasks or sample characteristics.

system, we record the top paths for each concept entry in the knowledge base. Concepts appearing in the same paths are defined as "related concepts" in the knowledge base. Users can conveniently navigate the knowledge from one concept to related concepts in the knowledge base.

The Resource Description Framework (RDF) is a data format widely used in representing knowledge involving relations among entities. In PhenoWiki+, we store the relations of concepts in triples, e.g., "working memory :: is related to :: prefrontal cortex." Some relations are defined in the lexicon. For example, the concept hierarchy presented in Figure 3 can be represented as "PFC :: is-subconcept :: Frontal Lobe." Moreover, by using path knowledge discovery tools, association paths among concepts can be identified. Such relationships can also be stored in the knowledge base and represented as triples in RDF. In our implementation, each path is defined as an entity in the RDF data model and has a sequence of associations. For each association, the antecedent and consequent concepts and strength measures are defined as properties. Furthermore, the path content, such as documents and quantitative data, can be also linked to the associations and the path. Figure 18 presents the data model of a path.

Furthermore, since RDF is a well-adopted standard in knowledge representation, we can link our knowledge with many external knowledge bases — including the Cognitive Atlas [21] and GO [5]. This capability can greatly enlarge our scope of knowledge acquisition. For example, in the implementation of PhenoWiki+, by including the knowledge graph of the Cognitive Atlas, we were able to include the "is-part-of" relations and the "is-a-kind-of" relations among concepts, as well as the concepts defined by domain experts in the Cognitive Atlas.

PhenoWiki+ summarizes various relations about a concept in its concept summarization page (Figure 19). From this page, users can navigate to other related concepts, look for related literature, and find related paths.

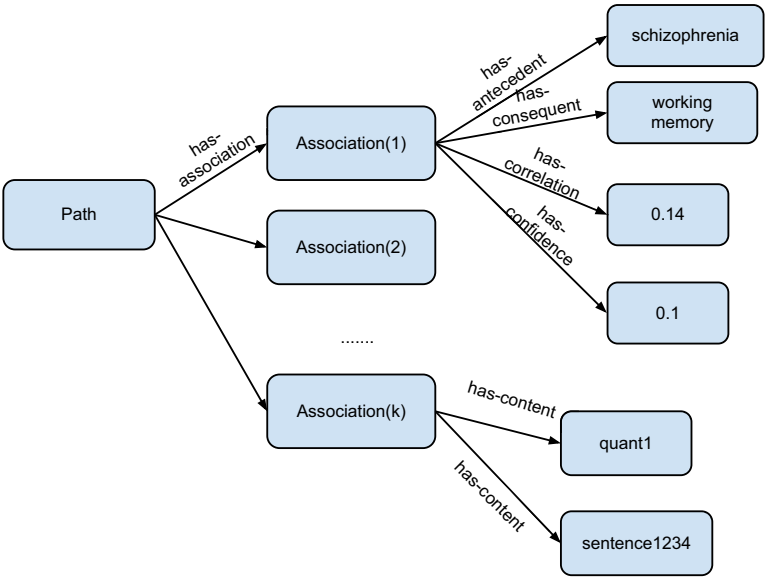


Fig. 18 Path data model. A path consists of an ordered sequence of associations. For each association, the antecedent and consequent concepts, and the strengths measures such as support, confidence and correlations, are defined as properties. As an example, the figure shows the properties for the association schizophrenia → working memory in the path presented in Figure 7. Path content is linked with a path via the "has-content" relationship referenced by the identifiers of the corresponding content.

6.4 Annotations: Complementing Mining Results with Human Knowledge

In the PhenoWiki+ system, along with the content generated by the automatic mining results, user-added annotations are also supported. These annotations are pieces of free text attached to information in the knowledge base, such as a concept, a paper, or a quantitative result. The annotation text is indexed and searchable in the knowledge base. Annotation has two major functions in PhenoWiki+ — providing additional knowledge and helping index data in the knowledge base.

Annotations provide additional domain expert knowledge to the mining results. Not all knowledge can be extracted by automated mining. For example, a paper may discuss the gene COMT but do so in the absence of any specific discussion about disease-relevance. In this case, an annotation can be added to the paper to suggest related syndromes. In addition, quantitative results may depend on specific experimental conditions, which can be documented by annotations. Since all annotations are indexed in the knowledge base, the augmented information can improve the search.

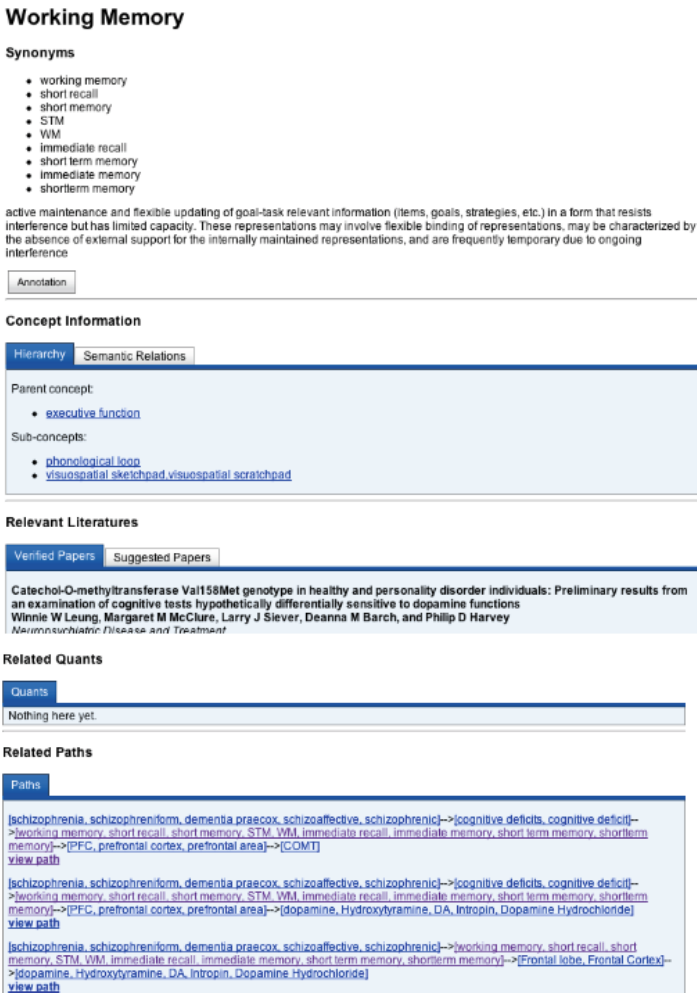


Fig. 19 Concept summarization page for working memory. Various types of data related to the concept are presented, including concepts, content (documents and quants) and related paths.

Annotations can also be used as an organizational tool, i.e., as a "user-generated index." For example, annotations on neuroanatomical concepts can link them with putatively related syndrome concepts. In this way, annotations can help the system understand the connections between concepts which are not identified by automatic text mining techniques. The PhenoWiki+ system supports two types of annotations — personal and public. Personal annotations are only visible and searchable by the current user; these can be used as a "personal index". Public annotations are visible to all users for collaborative development of the knowledge base. Figure 20 presents examples of two types of annotations.

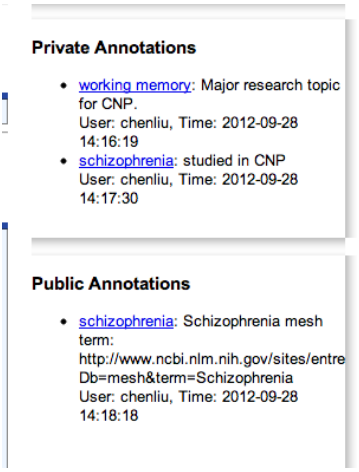


Fig. 20 Private and public annotation examples for different concepts. Private annotations are like personal bookmarks on concepts. In this figure, a concept is privately annotated with a project name. The public associations are used for collaborative development of knowledge bases. In this example, the user contributes the MeSH term link of the concept as a public annotation.

7 Related Work

The path knowledge discovery problem comprises two integral parts — discovery of the relations among concepts in different levels and retrieval of the path content describing such relationships. To the best of our knowledge, there is no existing published work covering both aspects of the problem. In the following, we will discuss the related work in association rule mining and relation discovery, knowledge-based content retrieval, and related studies from the Consortium for Neuropsychiatric Phenomics.

7.1 Association Rule Mining and Relation Discovery

Traditional association rule mining studies [3, 12] have focused on finding recurring patterns. As described in [29], the association rules discovered are primarily intended to identify rules such as, "a customer purchasing item A is likely to also purchase item B." Extending this approach to the bioinformatics field, association rule mining has been used to profile gene expression [10] and study protein-protein interaction [19]. These studies focus on the discovery of individual associations.

In [32], Tan et al. studied indirect associations, which are a special type of association rule describing associations $A \rightarrow B \rightarrow C$: "A customer purchasing item A is likely to also purchase item $B_i \in B$, and a customer purchasing item B_i is likely to

also purchase item C," where $i = 1, 2, \ldots, n$. By introducing the intermediate item sets B, the rules reveal a "higher-order" (indirect) data dependency between A and C. This higher-order dependency is similar to the idea of path in our work. However, there are some major differences in path knowledge discovery. First, the goal of mining is different. The high-order dependency focuses on identifying pairs of indirectly related item sets connected by an intermediate item set. Our path mining not only identifies such indirect relations, but also requires that the intermediate relations satisfy a certain pattern specified in the path query. Second, our path mining is closely integrated with content retrieval. Instead of only identifying relations, our path knowledge discovery process also provides relevant content describing such relations.

Association analysis involving intermediate concepts has been applied in bioinformatics. Baker et al. [6] developed a method for mining connections between chemicals, proteins and diseases using the biomedical literature as a knowledge source. Voytek et al. [35] developed a semi-automatic way to extract the "cognome" — relationships between brain structure, function and disease. Both works essentially followed the model that "if A is related to B, and B is related to C, then A is likely to be related with C". These authors empirically evaluated their results by comparing them with human-generated ones. However they did not employ quantitative measurements in these relations, or extend their methods to an association with more than three concepts. Our work presents a methodology to evaluate sequences of associations and discover path associations with a multilevel lexicon from a large text corpus. Moreover, the introduction of wildcard concept levels greatly increases the path discovery scope and can lead to new hypotheses for further research.

There are also other literature-based discovery tools based on association rule mining. BITOLOA [13] is a tool mining the association pattern $X \rightarrow Y \rightarrow Z$ when two of the three concepts are specified by the user (e.g., the user may specify X and Y or X and Z). Arrowsmith [30] is a tool that finds the links between two separate sets of documents via common title words and phrases. Both of these tools are based on patterns of pairwise associations between three concept sets. By contrast, our tools provide not only the ability to mine more complex path patterns, but also the ability to retrieve relevant path content.

7.2 Knowledge Based Information Retrieval and Path Content Discovery

In comparison to existing work that focuses on revealing the relations among concepts, our work defines path knowledge in a broader scope. Not only does it include relationships among concepts, but it also includes content mining to describe such relationships and facilitate deeper analysis; this naturally leads to connections with knowledge-based information retrieval systems.

Our approach for content retrieval is an extension of traditional information retrieval vector-space models [26] and term indexing and ranking methods [27, 23].

Using the concept hierarchy provided in the lexicon, we are able to achieve better information retrieval performance and preprocessing and post-processing of the query to facilitate various research goals. The utilization of domain knowledge in information retrieval systems has been studied by e.g., Liu et al. [17], who performed scenario-based knowledge expansion using the Unified Medical Language System (UMLS) — a well-defined medical ontology. Since our system relies on a lightweight multilevel hierarchy lexicon, we use the knowledge hierarchy to perform knowledge expansion, which is similar to approaches in ontology-based knowledge expansion [7].

The novel contributions of our approach to content retrieval are: 1) The query is translated based on a path and focuses on the content describing the relations among concepts; 2) Our content retrieval focuses on finer-granularity content such as sentences and figures, and the classification of the content provides a deeper understanding of the content based on knowledge from the lexicon. As a result, such mining greatly reduces the human labor involved in reading papers and digesting content, and improves the scalability and quality of the findings.

7.3 Relations to Other CNP Projects

Motivated by the multilevel schema proposed by the Consortium for Neuropsychiatric Phenomics [8, 9], information systems for the study of multilevel phenomics can be constructed. PubAtlas [20] is a web service and standalone program extending PubMed by exploring the associations between concepts with temporal analysis. PhenoWiki [25] and Cognitive Atlas [21] are open collaborative projects that provide knowledge bases for cognitive neuroscience. Unlike these existing tools, PhenoMining focuses on efficient discovery of relations among concepts under a multilevel schema, and on providing finer-granularity knowledge. For example, PhenoMining tools can be used to facilitate research on relationships among multiple levels of concepts (e.g., the heritability of complex phenotypes such as cognitive control), but also can be used for populating knowledge bases such as PhenoWiki. Along with other applications developed for CNP, path knowledge discovery enables comprehensive and systematic study of neuropsychiatric phenomics with a multilevel schema.

8 Conclusion

Path knowledge discovery consists of two integral parts — path discovery and path content retrieval — and focuses on the study of relations among concepts at multiple levels. This is useful in many research fields where a vast number of concepts are involved, and establishing relations among concepts across levels is important.

Our path discovery identifies and measures a path of knowledge, i.e., a sequence of associations among concepts at different levels. We have proposed two

approaches for measuring the strength of these path associations — the local strength measure in which associations are considered independent, and the global strength measure in which the preceding path associations are considered preconditions of following associations. We have also extended support, confidence, and correlation measures from traditional association rule mining to the context of path discovery.

Path content retrieval is a process of searching for relevant content describing the relations specified in the paths from a particular corpus. Path content reveals the semantics of relations represented in the paths and provides a basis for deeper analysis and further study. With the knowledge from the multilevel lexicon, we are able to preprocess the query by expanding queries using the synonyms list and the concept hierarchy, and post-process the query by classifying content according to different research goals.

We presented an example of using path knowledge discovery to examine a relevant research question in neuropsychiatric phenomics: What is the heritability of the complex phenotype *cognitive control*? Compared to manual marathons by human domain experts, path knowledge discovery can greatly reduce labor and achieve results of comparable quality. Preliminary results show the benefit of using data mining for path knowledge discovery in order to study complex problems. We also applied our mining results to the construction of a knowledge base. By extending the PhenoWiki system, the PhenoWiki+ system overcomes the difficulties of knowledge acquisition for traditional knowledge base systems by accelerating the data populating process and connecting scattered knowledge with paths.

Although our work on path knowledge discovery represents a significant step forward, further work is needed to improve the accuracy of our methodology. First, path discovery is currently based on association strengths used to satisfy strength constraints for the paths. The threshold setting can be difficult for users. If training data with labeled paths is available, machine learning techniques may be used to automatically set the thresholds. Second, currently the associations between concepts are based on statistical co-occurrence, but where more complete ontologies are available, more sophisticated computations based on the ontological structure can expose relations between concepts. Third, advanced information retrieval techniques, such as relevance feedback, may be used to improve search quality. We believe our approach to path knowledge discovery provides the framework for building sophisticated discovery tools for complex knowledge areas such as phenomics.

Acknowledgements. This work was supported by USPHS grants, including the NIH Roadmap Initiative Consortium for Neuropsychiatric Phenomics, including linked awards UL1DE019580, RL1LM009833. We thank Jianming He, Ying Wang, Andrew Howe, Jiacheng Yang, Jianwen Zhou, Xiuming Chen and Jiajun Lu of the CoBase research group in the UCLA Computer Science Department for their work on the PhenoMining tools and PhenoWiki+ implementation. We would also like to thank Professors Carrie Bearden and Joseph Ventura from the Consortium for Neuropsychiatric Phenomics for the initial testing of the tools and for their stimulating discussions during the development of this work.

References

1. Phenomining lexicon, http://phenominingbeta.cs.ucla.edu/static/new_lexicon.txt
2. Pubmed central web site, http://www.ncbi.nlm.nih.gov/pmc/
3. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB 1994, pp. 487–499. Morgan Kaufmann Publishers Inc., San Francisco (1994)
4. Anokhin, A.P., Golosheykin, S., Grant, J.D., Heath, A.C.: Developmental and genetic influences on prefrontal function in adolescents: a longitudinal twin study of wcst performance. Neuroscience Letters 472(2), 119–122 (2010)
5. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al.: Gene ontology: tool for the unification of biology. Nature Genetics 25(1), 25 (2000)
6. Baker, N.C., Hemminger, B.M.: Mining connections between chemicals, proteins, and diseases extracted from medline annotations. Journal of Biomedical Informatics 43(4), 510 (2010)
7. Bhogal, J., Macfarlane, A., Smith, P.: A review of ontology based query expansion. Information Processing & Management 43(4), 866–886 (2007)
8. Bilder, R.M., Sabb, F.W., Cannon, T.D., London, E.D., Jentsch, J.D., Parker, D.S., Poldrack, R.A., Evans, C., Freimer, N.B.: Phenomics: the systematic study of phenotypes on a genome-wide scale. Neuroscience 164(1), 30–42 (2009)
9. Bilder, R.M., Sabb, F.W., Parker, D.S., Kalar, D., Chu, W.W., Fox, J., Freimer, N.B., Poldrack, R.A.: Cognitive ontologies for neuropsychiatric phenomics research. Cognitive Neuropsychiatry 14(4-5), 419–450 (2009)
10. Creighton, C., Hanash, S.: Mining gene expression databases for association rules. Bioinformatics 19(1), 79–86 (2003)
11. Glausier, J.R., Khan, Z.U., Muly, E.C.: Dopamine D1 and D5 receptors are localized to discrete populations of interneurons in primate prefrontal cortex. Cerebral Cortex 19(8), 1820–1834 (2009)
12. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD 2000, pp. 1–12. ACM, New York (2000)
13. Hristovski, D., Friedman, C., Rindflesch, T.C., Peterlin, B.: Exploiting semantic relations for literature-based discovery. In: AMIA Annual Symposium Proceedings, vol. 2006, p. 349. American Medical Informatics Association (2006)
14. Karlsgodt, K.H., Kochunov, P., Winkler, A.M., Laird, A.R., Almasy, L., Duggirala, R., Olvera, R.L., Fox, P.T., Blangero, J., Glahn, D.C.: A multimodal assessment of the genetic control over working memory. The Journal of Neuroscience 30(24), 8197–8202 (2010)
15. Kremen, W.S., Xian, H., Jacobson, K.C., Eaves, L.J., Franz, C.E., Panizzon, M.S., Eisen, S.A., Crider, A., Lyons, M.J.: Storage and executive components of working memory: integrating cognitive psychology and behavior genetics in the study of aging. The Journals of Gerontology Series B: Psychological Sciences and Social Sciences 63(2), P84–P91 (2008)
16. Lesh, T.A., Niendam, T.A., Minzenberg, M.J., Carter, C.S.: Cognitive control deficits in schizophrenia: mechanisms and meaning. Neuropsychopharmacology 36(1), 316–338 (2010)
17. Liu, Z., Chu, W.W.: Knowledge-based query expansion to support scenario-specific retrieval of medical free text. Information Retrieval 10(2), 173–202 (2007)

18. U.S. National Library of Medicine. Fact sheet. medical subject headings, http://www.nlm.nih.gov/pubs/factsheets/mesh.html
19. Oyama, T., Kitano, K., Satou, K., Ito, T.: Extraction of knowledge on protein–protein interaction by association rule discovery. Bioinformatics 18(5), 705–714 (2002)
20. Parker, D.S., Chu, W.W., Sabb, F.W., Toga, A.W., Bilder, R.M.: Literature mapping with pubatlas extending pubmed with a blasting interface. Summit on Translational Bioinformatics 2009, 90 (2009)
21. Poldrack, R.A., Kittur, A., Kalar, D., Miller, E., Seppa, C., Gil, Y., Parker, D.S., Sabb, F.W., Bilder, R.M.: The cognitive atlas: toward a knowledge foundation for cognitive neuroscience. Frontiers in Neuroinformatics 5 (2011)
22. Prud'hommeaux, E., Seaborne, A.: Sparql query language for rdf, http://www.w3.org/TR/rdf-sparql-query/
23. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Found. Trends Inf. Retr. 3(4), 333–389 (2009)
24. Runyan, J.D., Moore, A.N., Dash, P.K.: A role for prefrontal calcium-sensitive protein phosphatase and kinase activities in working memory. Learning & Memory 12(2), 103–110 (2005)
25. Sabb, F.W., Bearden, C.E., Glahn, D.C., Parker, D.S., Freimer, N., Bilder, R.M.: A collaborative knowledge base for cognitive phenomics. Molecular Psychiatry 13(4), 350–360 (2008)
26. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM 18(11), 613–620 (1975)
27. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. 24(5), 513–523 (1988)
28. Seamans, J.K., Durstewitz, D., Christie, B.R., Stevens, C.F., Sejnowski, T.J.: Dopamine D1/D5 receptor modulation of excitatory synaptic inputs to layer V prefrontal cortex neurons. Proceedings of the National Academy of Sciences 98(1), 301–306 (2001)
29. Silverstein, C., Brin, S., Motwani, R.: Beyond market baskets: Generalizing association rules to dependence rules. Data Min. Knowl. Discov. 2(1), 39–68 (1998)
30. Smalheiser, N.R., Torvik, V.I., Zhou, W.: Arrowsmith two-node search interface: A tutorial on finding meaningful links between two disparate sets of articles in medline. Computer Methods and Programs in Biomedicine 94(2), 190 (2009)
31. Stins, J.F., van Baal, G.C.M., Polderman, T.J.C., Verhulst, F.C., Boomsma, D.I.: Heritability of stroop and flanker performance in 12-year old children. BMC Neuroscience 5(1), 49 (2004)
32. Tan, P.-N., Kumar, V., Srivastava, J.: Indirect association: Mining higher order dependencies in data. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 632–637. Springer, Heidelberg (2000)
33. Vinkhuyzen, A.A.E., Van Der Sluis, S., Boomsma, D.I., de Geus, E.J.C., Posthuma, D.: Individual differences in processing speed and working memory speed as assessed with the sternberg memory scanning task. Behavior Genetics 40(3), 315–326 (2010)
34. Von Huben, S.N., Davis, S.A., Lay, C.C., Katner, S.N., Crean, R.D., Taffe, M.A.: Differential contributions of dopaminergic D1-and D2-like receptors to cognitive function in rhesus monkeys. Psychopharmacology 188(4), 586–596 (2006)
35. Voytek, J.B., Voytek, B.: Automated cognome construction and semi-automated hypothesis generation. Journal of Neuroscience Methods (2012)

InfoSearch: A Social Search Engine

Prantik Bhattacharyya and Shyhtsun Felix Wu

Abstract. The staggering growth of online social networking platforms has also propelled information sharing among users in the network. This has helped develop the user-to-content link structure in addition to the already present user-to-user link structure. These two data structures has provided us with a wealth of dataset that can be exploited to develop a social search engine and significantly improve our search for relevant information. Every user in a social networking platform has their own unique view of the network. Given this, the aim of a social search engine is to analyze the relationship shared between friends of an individual user and the information shared to compute the most *socially relevant* result set for a search query.

In this work, we present *InfoSearch*: a social search engine. We focus on how we can retrieve and rank information shared by the direct friend of a user in a social search engine. We ask the question, within the boundary of only one hop in a social network topology, how can we rank the results shared by friends. We develop *InfoSearch* over the Facebook platform to leverage information shared by users in Facebook. We provide a comprehensive study of factors that may have a potential impact on social search engine results. We identify six different ranking factors and invite users to carry out search queries through *InfoSearch*. The ranking factors are: 'diversity', 'degree', 'betweenness centrality', 'closeness centrality', 'clustering coefficient' and 'time'. In addition to the *InfoSearch* interface, we also conduct user studies to analyze the impact of ranking factors on the social value of result sets.

Keywords: Online Social Network, Social Search.

1 Introduction

Users in online social networks have surpassed hundreds of millions in number. With this staggering growth in the network size, social network platforms like Facebook and Twitter have introduced various software tools to engage users. In

Prantik Bhattacharyya · Shyhtsun Felix Wu
University of California, Davis, 1 Shields Ave, Davis, CA
e-mail: {pbhattacharyya,sfwu}@ucdavis.edu

W.W. Chu (ed.), *Data Mining and Knowledge Discovery for Big Data,*
Studies in Big Data 1,
DOI: 10.1007/978-3-642-40837-3_6, © Springer-Verlag Berlin Heidelberg 2014

addition to connecting and exchanging messages with friends on a regular basis, social network platforms also provide a great place to share useful information. Consequently, people have become very good at sharing the information that they value, support, endorse and think their friends might benefit from. Users share their favorite web-page(s) on current affairs, news, technology updates, programming, cooking, music and so on by sharing Internet URLs with their friends through the social network platform. Facebook has introduced 'Like', 'Share' and 'Recommend' buttons that content providers of any website can include on their website to help visitors share the URLs with their friends in a fast and easy way. Twitter has also introduced similar technologies to let users 'Tweet' the URL in addition to their personal comment about the URL.

Fig. 1 Example of Information Sharing over Online Social Network (Facebook in this example)

The simplicity and ubiquitousness of this technology has propelled the integration of the web graph with the social graph. The additional information present in each individual's personal network can be utilized to develop search engines that include social context in information retrieval and ranking. In typical web search engines, users are restricted to search for information from the global web and retrieve results that are ranked relevant by a search engine's algorithm. For example, web search engines like Google, Yahoo! and Bing traditionally analyzes the information present in the form of hyper-link structures to rank results during a typical query. The intuitive justification for utilizing the hyperlink structure to rank web-pages is based on the idea that one web-page links to another web-page to indicate usefulness and relevance. During the process of crawling, indexing and ranking, each search engine formulates result set(s) for a set of keyword that are unique in nature and are identical to every user visiting the search engine. For example, when users search for queries related to 'programming' or 'cooking recipes', search results are similar in nature to every individual performing a query on the engine.

A search engine result set, however, can be significantly updated to incorporate social context as a factor during the ranking process. The social context in retrieving results will allow users to identify results based on the way their friends have shared and endorsed similar information. Each search query from a user will thus retrieve a unique set of result. The exclusive nature of each result set will thus be based on the large volume of information available in each individual user's personal network. The search process thus not only enables a user to access a set of information that has a distinct social component attached to it but also to gain from the collective

knowledge of their respective social network. In other words, a search process is no longer limited to retrieving a random piece of information from the Internet with no trust value attached to it but extends to a retrieval process that includes a trusted source, that is, their friends' personal attachment or endorsement of that piece of information. Providing search results exclusively from the personal network of users creates a scope of unique challenges. How do we understand the relative importance of one user to another user in the network? How do we rank individual users? What are the primary factors that exemplify social relationship semantics?

The growth in the volume of shared information has also altered the way major search engine providers like Google and Microsoft rank web-search results. In 2011, the search engine companies introduced signals in their ranking algorithms to reflect patterns of information share across the social graph [25, 40]. The primary efforts are concentrated to introduce signals from social sharing to explore popularly shared URLs and boost their corresponding rankings in a result set that continues to be identical for all users with respect to a specific query.

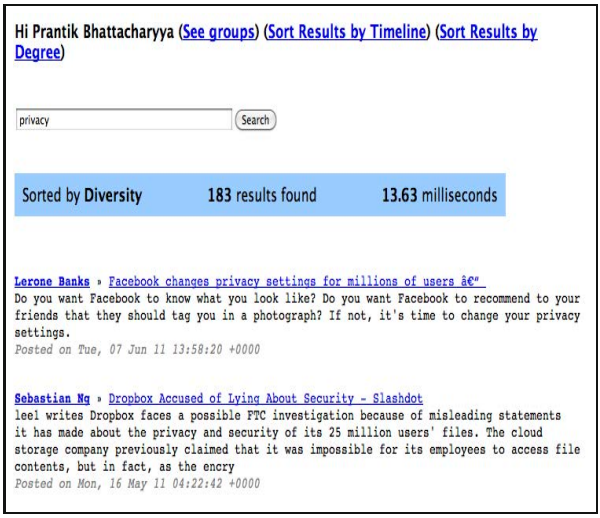


Fig. 2 Screenshot of InfoSearch Application on Facebook: Results for the query 'privacy' appear for one of the authors

In this work, we develop a search engine to demonstrate how user shared information can be exploited to deliver search results. Our work can be described in two parts. In the first part, we develop the social search engine system based on the Facebook platform that leverages the information shared by users in Facebook as an extension of our previous work [6]. The search engine is called *InfoSearch* and is available at `https://apps.facebook.com/infosearch`. In the second part of our work, we discuss key issues that influence result ranking. We explore questions on how we can define the best result in a social context. In the absence

of ground truth data about the relationship shared between two users (in real or online life), we investigate different ranking factors to analyze the social relationship between two users and rank search results. We provide a comprehensive study of factors that impact social search engine results. The ranking factors are based on an analysis of the structure of the social relationship between friends of a given user: social diversity, three different measures of centrality: degree, betweenness centrality and closeness centrality, a measure of clustering: clustering coefficient and finally a factor based on the time property of a shared information. We derive the social relationship between two users (friends) of a given user based on the social group structure shared between them in the user's individual social network. We present results based on the impact of the above ranking factors in retrieving information through user studies.

In section 2, we discuss related work. We formally describe the problem statement related to social search in section 3 and follow up with a discussion of social network relationship semantics in section 4. In section 5, we discuss the ranking factors and corresponding algorithms and section 6 describes the system development process. Section 7 presents statistics on usage. In section 8, we present our findings obtained through user studies and section 9 concludes with a discussion of future research directions.

2 Related Work

We discuss related work in this section. First, we discuss work in the area of search in social networks. Second, we discuss research related to the study of social relationship semantics. We primarily focus on research related to group and community formation in social networks.

Several projects have looked into the area of search in social networks. The research problems have broadly fallen into the following categories. First, the identity or profile search problem in which social network information is used to connect and subsequently search for users. Dodds et. al. [14] conducted a global social-search experiment to connect 60,000 users to 18 target persons in 13 countries and validated the claims of small-world theory. Adamic et. al. [1] conducted a similar project on the email network inside an organization. More recently, Facebook has introduced 'Graph Search' [15] that aims to help user search for content linked by their friends. Facebook defines a content as any object on the open graph api. Examples of object in the open graph api include facebook-pages (e.g. a facebook account created by a local business, musician, artist) , facebook-apps (e.g. social games), facebook groups (e.g. university course groups, athletic group), photos shared by its users and geographic locations shared by the users.

In the second category, social networks have been leveraged to search for experts in specific domains and find answer to user questions. Lappas et. al. [23] addressed the problem of searching a set of users suitable to perform a job based on the information available about user abilities and compatibility with other users. The work in [11] attempted at automated FAQ generation based on message routing in a

social network through users with knowledge in specific areas. Other works in similar directions have also been presented, e.g. [10, 33]. Query models [2] based on social network of users with different levels of expertize for the purpose of decentralized search have also been developed. Horowitz et. al. [21] presented *Aardvark*, a social network based system to route user questions into their extended network to users most likely knowledgeable in the context of the question.

In the third category, social networks are considered to improve search result relevancy. User connections are interpreted as a graph such that a user can be represented as a node and each friend connection can be treated as an edge between two nodes. Haynes et. al. [20] studied the impact of social distance between users to improve search result relevancy in a large social networking website, *LinkedIn*. The author defined the social distance between users based on the tie structure of the social graph and aims to provide improved relevance and order in profile identity entries. Link analysis algorithms, like PageRank [7, 9, 13], are also not suitable for application since during the search process of an individual user, results from members of their social circle should not be ranked based on a generalized analysis of the relative importance of those members in the larger network but rather on their local importance to the querying user [24, 38]. Mislove et. al. [26] considered the problem of information search through social network analysis. They compare the mechanisms for locating information through web and social networking platforms and discuss the possibility of integrating web search with social network through a HTTP proxy.

A primary way to understand social relationships is by analyzing social group formation in social networks. Work in group detection in graphs are primarily associated with community detection and graph partitioning problems. Past works [29] describe the motivation and technical differences between the two approaches. Detailed discussions can also be found in the recent survey [16]. Here, we discuss works related to community detection in social networks.

A common approach for finding sub-communities in networks uses a percolation method [12, 32, 31]. Here, k-clique percolation is used to detect communities in the graphs. Cliques in the graph are defined as complete and fully connected subgraphs of k vertices. Individual vertices can belong to multiple cliques provided that the overlapping subgroups don't also share a $(k-1)$ clique. The work in [17] uses centrality indices to find community boundaries in networks. The proposed algorithm uses betweenness between all edges in the network to detect groups inside the graph. The worst-case runtime of the algorithm is $O(m^2n)$ for a graph of m edges and n vertices and is unsuitable for large networks. Improvements in the runtime have been suggested in later works [34, 36]. Impact of network centrality on egocentric and socio-centric measures have also been studied [24].

The betweenness approach places nodes in such a way that they exist only in a single community, restricting the possibility of overlapping communities and detecting disjoint groups in the network. To overcome this shortfall, algorithms in [18, 19] have proposed the duplication of nodes and local betweenness as a factor in detection of communities. Other approaches to identify overlapping communities have also been proposed [5, 4]. The above works describe the community structure based

on relative comparison with the graph segment not included in the community [28] or based on comparisons with random graphs of similar number of nodes and vertices but different topological structures. For example, the definition of modularity [30] as an indicator of the community strength defines the measure as a fraction of the edges in the community minus the edges in a community created by the same algorithm on a random graph. Community definitions also include detection of groups within the network such that the interconnection between the different groups are sparse [18, 19]. In this work, we build the social search system on Facebook, utilizing the existing social graph as well as the information database being built by users. We discuss the details next.

3 Social Search – Problem Statement

In this section, we start with a discussion of the benefits of a social search engine and end by introducing the key information structures.

A user introduces an article to his/her friends by sharing the article URL on Facebook. It can be intuitively theorized that the user shared the article because he/she found the article to be relevant and beneficial in a particular context. Through the sharing process, the user extends the information database of his/her social network with the context of the shared article and consequently other friends in their network can benefit from this endorsement. In the example of Figure 1 the primary context of the article is 'privacy'. Users in the network benefit from this shared knowledge when they try to find information related to 'privacy'. Furthermore, the social context in this case i.e. the person who shared this information can help querying user(s) to disambiguate and choose from a large number of articles available on 'privacy' in general on the web.

It is important to understand that the subjectiveness of social relationships make it extremely difficult to correctly predict the value of each relationship. Furthermore, in the absence of ground truth data, it is also difficult to accurately postulate that one friend or user is more important compared to another user. In this direction, we focus on computing the most socially relevant "result set" rather than emphasizing on ranking individual results in a result set. Thus, in this work the relevance of a comprehensive result set is given a higher priority over the ranking of individual results during a search query and relevance values of each result sets are determined to select the best result set for a given user query. Next, we formally define the key information structures required to develop a social search engine and rank query results.

Definition 1. Social Network: A social network is a graph $G = (V,E)$, where V is a set of nodes and E is a set of edges among V. A node stands for a user in the social network, and an edge e stands for a connection between two users u and v. In our work, we consider undirected edges. The shortest geodesic distance between two nodes n_1 and n_2 in the network is defined as $d(n_1,n_2)$. Let $d(n_1,n_2) = \infty$ if no path exists between the nodes in the network.

Definition 2. Ego Network: For a user u, ego network is a graph $G(u) = (V(u), E(u))$, where $V(u)$ is a set of nodes that includes all friends of u, $F(u)$ and the node u itself. $E(u)$ is a set of edges among $(V(u) - u)$ such that $\forall v \in (V(u) - u)$, v and u are friends and share an edge in E. Additionally, all edges between nodes in $(V(u) - u)$ that existed in E are also included in $E(u)$.

Definition 3. Mutual Friend Network: A mutual friend network of an user u is defined as a subset of the ego network, represented as $MF(u) = (F(u), E'(u))$. $F(u)$ is the set of all friends of user u and $E'(u)$ is a subset of the edges from $E(u)$ with the edges between user u and nodes in $F(u)$ absent.

Definition 4. Shared Information: A shared information in a social network can be identified as an URL or a document. An URL or document shared by a user u is denoted by the tuple (u,d). Each shared URL or document is tagged by a set of keywords $K(d) = (k_1^d, k_2^d, ..., k_m^d)$. Additionally, each information is also tagged by a time-stamp, $T(d)$, based on the time the information was shared by the user in the social network platform.

Definition 5. Query: A query q by a user u is defined as $Q(u,q)$. The query q can be a single keyword or a set of keywords i.e. a key-phrase. We discuss details about how we distinguish keywords and key-phrases during the search process later in section 6.

Definition 6. Factor: The term 'factor' is used to define a ranking factor that orders and ranks results in the search process. The factors used in this work are defined in section 5.

Definition 7. Result Candidates: The result candidates, $RC(Q(u,q))$ for a query $Q(u,q)$ is defined as the set of shared document tuples (v_i, d_j) such that $v_i \in F(u)$ and $\forall d_j, q \in K(d_j)$.

Let the number of results in $RC(Q(u,q))$ be represented as λ such that $\lambda = |RC(Q(u,q))|$. Lets also denote the number of users in result candidates tuple list as λ_v and the number of documents by λ_d. Also, lets assume the number of unique users in the above list as λ'_v.

Definition 8. Result Set: A result set, $RS(Q(u,q))$, for a query $Q(u,q)$ is defined as a set of ρ document tuples (v_i, d_j) such that $v_i \in F(u)$ and $\forall d_j, q \in K(d_j)$. Thus, for a query $Q(u,q)$ with result candidates, $RC(Q(u,q))$, the number of result sets possible is given by $\alpha = \lceil \frac{|RC(Q(u,q))|}{\rho} \rceil$.

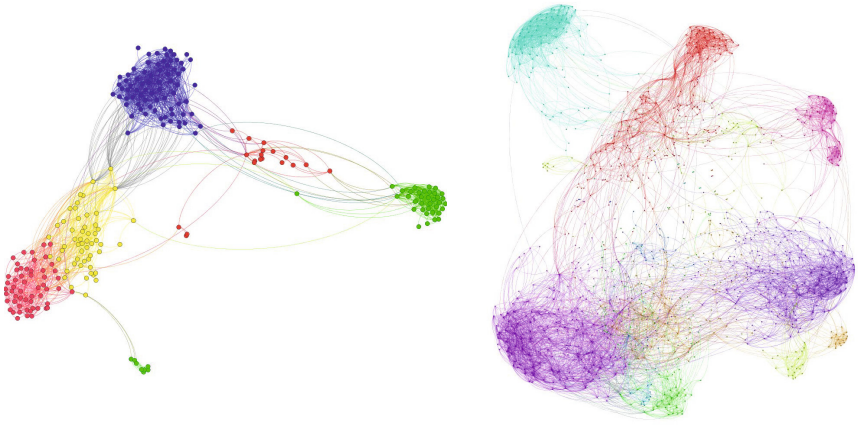
Definition 9. Result Value: The result value of a result set for a given *factor* is defined as $RV(RS(Q(u,q)), \text{Factor})$. The method to compute the result value of a result set will vary according to the factor and will be described in section 5 along with each factor.

Definition 10. Result Final: The result final is a collection of result sets, ordered by decreasing result value. Thus, the result final for query $Q(u,q)$ can be defined as $RF(Q(u,q)) = \{RS_1(Q(u,q)), RS_2(Q(u,q)), .., RS_\alpha(Q(u,q))\}$ such that $RV(RS_1(Q(u,q)), \text{Factor}) \geq RV(RS_2(Q(u,q)), \text{Factor}) \geq ..RV(RS_\alpha(Q(u,q)), \text{Factor})$.

In the next section, we will discuss and define the semantics of social relationship to formalize contribution of each user as they impart social context to formulate the final result set.

4 Semantics of Social Relationships

There are multiple ways to understand the relationship between two users, v and w, in a social network. The analysis can be based on the understanding of the social groups present in the graph or measures of centrality or the clustering properties of the social network graph. In the absence of ground truth data about the relationship shared between two users (in real or online life), in this work we explore multiple properties to analyze the social relationship between two users and provide a comprehensive study of factors that may have a potential impact on social search engine results.



(a) userA **(b)** userB

Fig. 3 Mutual friend network visualizations

We base our analysis of the relationship between two users from the point of view of the user, u, performing a search query through the search engine. Thus, we analyze the relationship shared between users, v and w, through the mutual friend network of the user u i.e. $MF(u)$. For different users, u_1 and u_2 with respective mutual friend networks, $MF(u_1)$ and $MF(u_2)$ such that the graphs are distinct either in terms of topology or based on the number of users present in the network, the relationship shared between two users v and w where both $v, w \in MF(u_1)$ and $MF(u_2)$ may vary accordingly. We present example mutual friend network visualizations in Figure 3. The visualizations represents the mutual friend networks of 'userA' and 'userB' (details about the users and the network properties are mentioned in

section 8.2), respectively, from their Facebook profile. The visualizations were created using the Gephi platform [3].

We empirically determine the social groups of a user's network by analyzing the mutual friend network of the user. In centrality based methods, we use the factors of degree, betweenness centrality and closeness centrality. We further explore clustering based methods, namely local clustering coefficient property, to determine the social relationship semantics between two users, v and w. We present an example in Figure 4. Ego e is connected to all the other nodes in the graph and shown using a broken line between the vertices and ego e. The mutual friend network of the ego e is shown by the connected lines between the other vertices of the figure. We introduce the formal definition of each relationship characteristic and compare and contrast the merits of each property next.

A social group in the ego-network of user u can be defined as a set of friends who are connected among themselves, share a common identity and represents a dimension in the social life of the user u. A social group can be defined in multiple ways. In this work, we base our definition on mutuality [38] and the formal definition is presented next.

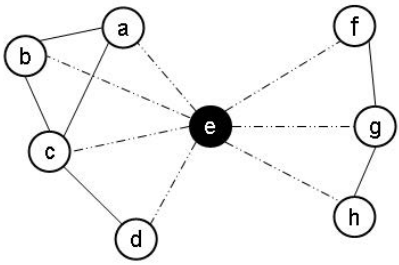


Fig. 4 Example ego-network of ego e

Definition 11. Social Group: A social group of a user u is defined as $sg(u) = (V'')$ where V'' is a set of vertices such that $V'' \subseteq F(u)$ and for two users v and w in V'', $d(v,w) \leq k$ in the mutual friend graph, $MF(u)$. The set of all such social groups formed from the mutual friend graph of a user u is represented as $SG(u)$.

4.1 Social Groups

The above definition allows for duplication of users across different social groups since a user can belong to multiple social groups as it satisfies the geodesic requirement with other users of each group.

Let user u's social circle be divided into a set of groups represented as $SG_u = \{sg_u^i\}$, where $1 \leq i \leq ng_u$, ng_u represents the number of social groups formed. Based on two different parameter values, examples of such groups are presented in Figure 5 and Figure 6. We observe that four social groups are discovered for $k = 1$. Nodes c and g overlap in both the two groups. Now, when we inspect the graph for $k = 2$,

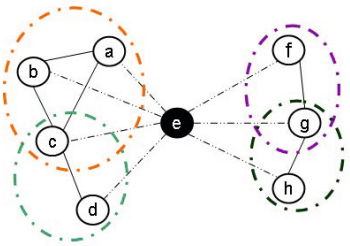


Fig. 5 Social Groups for an ego e at $k = 1$

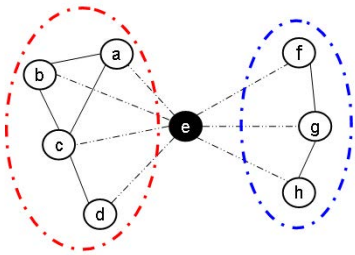


Fig. 6 Social Groups for an ego e at $k = 2$

we discover only 2 social groups with no overlapping vertices. It is also important to note here that further increase in the value of k has no effect in group generation. Thus, in a way the group formation gives a sense of separation or distance between the users based on the value of k for the group formation process.

We use the set of all social groups formed from the mutual friend graph of an user u to next define the social distance between two users present in the ego network of user u. Let user v belong to the set of social groups g_v such that $g_v \subset SG(u)$. Let η_v represent the cardinality of g_v and let each element of set of groups g_v be represented as g_v^i such that $1 \leq i \leq \eta_v$. We utilize the group member information to next define group distance and user distance.

Definition 12. Social Group Distance: The distance between two social groups is defined to be equal to the Jaccard distance between the groups. For two social groups, $sg(u)_i$ and $sg(u)_j$, from the set $SG(u)$ of user u, distance is defined as:

$$dist(sg(u)_i, sg(u)_j) = 1 - \left(\frac{|sg(u)_i \cap sg(u)_j|}{|sg(u)_i \cup sg(u)_j|} \right) \tag{1}$$

Definition 13. User Distance in Ego Network: User distance between two users, v and w, in the ego network of user u is defined as the mean distance between the two user's associated group(s). For users v and w associated with η_v and η_w number of social groups represented by g_v^i and g_w^j such that $1 \leq \eta_v$ and $1 \leq \eta_w$ respectively, user distance is defined as:

$$\omega(v,w) = \frac{\sum\limits_{\substack{1 \leq i \leq \eta_v \\ 1 \leq j \leq \eta_w}} dist(g_v^i, g_w^j)}{\eta_u \times \eta_w} \tag{2}$$

The social group distance and user distance formula as proposed above paves the way for us to understand the social relationship between two users based on mutuality and creates scope for us to distinguish how distant (or close) users are to each other from the point of view of a single user. A high value in the user distance thus empirically suggests a separation (possibly to an extent of unfamiliarity) and furthermore existence of multiple facets to an individual's social life. For example, a typical individual has friends from their place of employment (which can

be multiple and fairly distinct as individuals move through phases of professional career growth), place of education (with strong possibilities of multiple and distinct groups again as individuals go through high school, college, graduate school, etc.) and so on. The concepts related to social groups and multiple sections of a user's social network are analogous and the terms have been used interchangeably in rest of the paper. The 'diversity' factor as will be introduced in section 5.1 tries to capture the underlying hypothesis from the above discussion and helps build search engine results by exploiting the information present in a dormant format in social group information.

The semantics described next are more direct in this approach to capture social relationships and are used more explicitly to define respective factors and rank results in the social search engine.

4.2 Degree

In this factor, we consider the degree of user v in $MF(u)$ i.e. the factor that indicates the number of users in $F(u)$ connect to v. Let, this value be represented as $deg(v, MF(u))$, for all $v \in F(u)$. In the example of Figure 4, users a, b and g has a degree of 2, user c has a degree of 3 and users d, f and h has a value of 1. The number indicates the strength of connectivity of a particular vertex in the mutual friend network. A high value can be interpreted as a signal of support for the friend and reflects their relative importance in $MF(u)$ and thus stands as an important signal to represent the social relationship shared between users.

While the value of degree (indegree and outdegree values in directional graphs) have been a signal of significant importance in graph based methodology developments, e.g. HITS, PageRank, in the context of social relationships and the mutual friend network of a user, the degree property can often formulate results to indicate biasness towards a few social relationships. For example, friends from a particular group (say place of work) can all know each other and can form complete graph, thus leading towards every user in the said group to have high and similar degree values and constraining the result set to include results from only one group. Other properties described next, e.g. betweenness, closeness centrality and clustering coefficient also tends to address these issues and thus, we believe 'diversity' offers a certain level of contrast to other social relationship characteristics and hence has the potential to offer interesting results in a social search engine result set.

4.3 Betweenness Centrality

The betweenness centrality characteristic of a node in a graph is used to quantify the extent to which a node lies between other nodes in the network [38]. The betweenness of a user v is represented as $C_B(v, MF(u))$ for all $v \in F(u)$. The measure based on the connectivity of a node's neighbors, assigns a higher value for nodes that bridge clusters in the graph. The measure indicates the number of users that

an individual user connects through to connect to other users in the graph and is another important signal to understand the semantics of social relationships.

The betweenness centrality of a node v is computed as [8]: $C_B(v,MF(u)) = \sum_{s\neq v\neq t\in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$ where σ_{st} is total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of those paths that pass through v. In the example of Figure 4, users a,b,d,f and h has a betweenness centrality value of 0.0, user c has a value of 0.13 and g has a value of 0.067.

4.4 Closeness Centrality

The closeness centrality characteristic is a measure of how a node is central in a given network. The measure is defined as the sum of (geodesic) distance of a given node to all other nodes in the network [35, 8, 38]. Consequently, a user is termed as more central in the network if the total distance to all other users is lower relative to other user's respective value.

The closeness centrality for a user v in the mutual friend network of user u is defined as [8]: $C_C(v,MF(u)) = \sum_{t\in V\setminus v} 2^{-d_{MF(u)}(v,t)}$. In the example of Figure 4, users a and b has a closeness centrality value of 0.375, user c has a value of 0.50, user d has a value of 0.30, user g has a value of 0.33 and users f and h has a value of 0.22.

4.5 Clustering Coefficient

The clustering factor captures the tendency of nodes to form a clique [39]. We particularly focus on the local clustering coefficient property of each user in the graph. For a user v in the mutual friend network of user u, let the neighborhood of the user be defined as $N_{v,MF(u)} = w_i$ where w_i is a user directly connected to user v and $d(v,w_i) = 1$ in $MF(u)$. Lets define k_v as the number of users, $|N_{v,MF(u)}|$, in the neighborhood, $N_{v,MF(u)}$ of user v in $MF(u)$.

The local clustering coefficient of each user v in the mutual friend network $MF(u)$ is defined as: $C_L(v,MF(u)) = \frac{2\times|e_{w_i,w_j}|}{k_v(k_v-1)}$ such that $w_i,w_j \in N_v$ and $e_{w_i,w_j} \in E'(u)$. In the example of Figure 4, user a and b has a local clustering coefficient value of 1.0, user c has a value of 0.33 and users d,f,g and h has a value of 0.0.

The details of how the ranking algorithms satisfy the requirement of decreasing result value for each result set in result final are simple and intuitive and left to the reader. Based on the above factors to identify the social relationship semantics between two users, we next define the ranking factors and present the associated ranking algorithms to compute results in a social search engine.

5 Ranking Factors and Algorithms

In this section, we expand on our discussion of semantics of social relationships to introduce ranking factors. In the first subsection, we describe the ranking factors

and also describe methods to evaluate the result value of any result set for a given ranking factor. In the final subsection, we talk about the ranking algorithm employed to rank results and determine the final result set(s) from the result candidates. We start by introducing the 'diversity' factor based on the definition of social groups as discussed in section 4.1.

5.1 Diversity

The 'diversity' factor is based on the social group information of the querying user. The purpose of this factor is to maximize group representation in a result set such that the social diversity in a result set is maximized and a higher user distance between the users present in the result set can help user u to inspect results that members from the various groups of the network share on the platform. The diversity value is based on the user-distance method defined in section 4.1 and is defined next.

Definition 14. Diversity. The diversity of a result set, $RS(Q(u,q))$, consisting of ρ results is defined as the mean user distance(s) between each pair of users.

$$\triangle(RS(Q(u,q))) = \frac{\sum_{v,w \in RS(Q(u,q))} \omega(v,w)}{|\rho|^2} \tag{3}$$

Definition 15. Diversity Result Value. The result value of a result set for the 'diversity' factor is defined as equal to the diversity value of the result set itself. Thus,

$$RV(RS(Q(u,q)), \text{'Diversity'}) = \triangle(u, RS(Q(u,q))) = \frac{\sum_{v,w \in RS(Q(u,q))} \omega(v,w)}{|\rho|^2} \tag{4}$$

5.2 Degree

The 'degree' factor is based on the definition of 'degree' from section 4.2. The purpose of this factor is to select friends of the user performing a query who have the highest number of connections in the mutual friend network and define relevance in a social context as related to each contributing user's popularity in the network.

Definition 16. Degree Result Value. The result value of a result set for the 'degree' factor is defined as the average of the degree value of all users present in the result set. Thus,

$$RV(RS(Q(u,q)), \text{'Degeee'}) = \frac{\sum_{v \in RS(Q(u,q))} deg(v, MF(u))}{\rho} \tag{5}$$

5.3 Betweenness Centrality

The 'between centrality' factor is based on the definition of betweenness centrality of individual users in the mutual friend network, section 4.3. The primary goal of this factor is to provide scope to build result sets such that users with highest values of betweenness centrality are ranked higher and provide relevancy to search engine results.

Definition 17. Betweenness Centrality Result Value. The result value of a result set for the 'betweenness centrality' factor is defined as the average of the betweenness centrality value of all users present in the result set. Thus,

$$RV(RS(Q(u,q)), \text{'Betweenness Centrality'}) = \frac{\sum_{v \in RS(Q(u,q))} C_B(v, MF(u))}{\rho} \quad (6)$$

5.4 Closeness Centrality

Similar to betweenness centrality, the ranking factor 'closeness centrality' is based on the definition of closeness centrality from section 4.4. The motivation here is to include results from users with higher values of closeness centrality in the top ranked result set.

Definition 18. Closeness Centrality Result Value. The result value of a result set for the 'closeness centrality' factor is defined as the average of the closeness centrality value of all users present in the result set. Thus,

$$RV(RS(Q(u,q)), \text{'Closeness Centrality'}) = \frac{\sum_{v \in RS(Q(u,q))} C_C(v, MF(u))}{\rho} \quad (7)$$

5.5 Clustering Coefficient

Clustering coefficient is introduced as a ranking factor based on the definition provided in section 4.5. The purpose here is include results from users with higher local clustering coefficients first and continue the process till all entries from result candidates are placed in result sets of decreasing value.

Definition 19. Clustering Coefficient Result Value. The result value of a result set for the 'clustering coefficient' factor is defined as the average of the clustering coefficient value of all users present in the result set. Thus,

$$RV(RS(Q(u,q)), \text{'Clustering Coefficient'}) = \frac{\sum_{v \in RS(Q(u,q))} C_L(v, MF(u))}{\rho} \quad (8)$$

5.6 Time

We introduce 'time' as the final factor to rank results. The time-stamp of each shared information, $T(d)$ is considered to rank the result candidates to compute the final result. In contrast to the previous factors that were based on the social relationship shared between users, the 'time' fator is established to reflect the most recent activity by users in the context of the query. For example, in the context of a query related to 'budget', the 'time' factor can successfully determine search results that link to the most recently shared information related to 'budget'.

Definition 20. Time Result Value. The result value of a result set for the 'time' factor is defined as the average time-stamp of the information set present in the result set. Thus,

$$RV(RS(Q(u,q)), \text{'Time'}) = \frac{\sum_{d \in RS(Q(u,q))} T(d)}{\rho} \tag{9}$$

In addition to the above definition of a result value for the factor 'time', we also measure the standard deviation in time-stamp values of the information set present in the result set. The standard deviation value helps us understand the extent of 'freshness' or 'real-time' nature of the results. In the next section, we discuss the algorithms employed to compute final result set for each ranking factor.

5.7 Ranking Algorithms

The ranking algorithm generates the set of final results, $RF(Q(u,q))=\{RS_1(Q(u,q)), RS_2(Q(u,q)), .., RS_\alpha(Q(u,q))\}$ from the set of result candidates, $RC(Q(u,q))$. The steps associated with the ranking algorithms for each ranking factor are described next. We will start by recounting the terminologies associated with the set of result candidates, $RC(Q(u,q))$. The number of results in $RC(Q(u,q))$ is represented as λ, i.e. $\lambda = |RC(Q(u,q))|$. Also, the number of users in result candidates tuple list is denoted as λ_v and the number of documents by λ_d. The number of unique users in the above list is assumed as λ'_v. A result set, $RS(Q(u,q))$, contains ρ tuples of information.

Each information has a time-stamp data marked by $T(d)$. If a user has shared multiple pieces of information, the information set is sorted by the time-stamp, $T(d)$. The most recently shared information is ranked highest followed by information shared at later dates. The algorithm associated for 'diversity' factor is described next.

5.7.1 Diversity

The result value for the 'diversity' factor is based on the relationship shared between two users (user distance property) present in the result set. The steps involved in the ranking algorithm for 'diversity' are described next.

1. If the number of result candidates is less than or equal to the size of a result set, i.e. if $\lambda \leq \rho$, then only one result set is possible and $RF(Q(u,q)) = RC(Q(u,q))$.
2. If the number of result candidates is greater than the result size set and the number of unique users is equal to the result set size, i.e. if $\lambda > \rho$ and $\lambda'_v = \rho$, then $RS(Q(u,q))$ is constructed using the most recently shared post (using information from $T(d)$) of λ'_v users. This automatically ensures that maximum value of diversity is achieved in the result set. If the starting condition of result candidates processing is this step, then the result set becomes the first result set of the final result set, i.e. $RS_1(Q(u,q))$. Now, $RC_{\text{new}}(Q(u,q)) = RC(Q(u,q)) - RS_1(Q(u,q))\}$. The values related to λ and λ'_v are updated accordingly and in the next iterations to construct result set $RS_2(Q(u,q)),...,RS_\alpha(Q(u,q))$, the applicable steps are followed.
3. If the number of result candidates is greater than the result size set and the number of unique users is less than the result set size, i.e. if $\lambda > \rho$ and $\lambda'_v < \rho$, $\binom{\lambda}{\rho}$ possible result sets are constructed and using the user information available in each result set, result value for the 'diversity' factor is computed. The result set with the highest value of diversity is selected and $RC(Q(u,q))$ is updated to repeat the steps to compute next set of results. A user may contribute multiple times in the result set but the process ensures that the result set has the highest value of diversity. In the case of multiple result sets with equal value of 'diversity', knowledge about time-stamps of each shared information included in the result set is used to break the tie and the result set with the highest value of time-stamp (i.e. the result set with the most recently shared documents) is selected as the result.
4. If the number of result candidates is greater than the size of a result set and the number of unique users is also greater than the result set size, i.e. if $\lambda > \rho$ and $\lambda'_v > \rho$, we start by first constructing $\binom{\lambda'_v}{\rho}$ number of sets and compute the diversity value of each set. The set with the highest value of diversity is selected and documents associated with each user is selected to formulate the result set. The most recently shared document by users are used and in case of tie in diversity values, time-stamp values are used to break the tie and the set of most recently shared documents are declared as winner. The set of result candidates, $RC(Q(u,q))$, is updated and the steps are repeated till the set of result candidates has no more entries.

Based on the relationship shared between two users in a result set, the algorithm to rank results for the 'diversity' factor contrasts the corresponding ranking algorithm of other factors. Algorithm for other factors are presented next.

5.7.2 Degree, Betweenness Centrality, Closeness Centrality and Clustering Coefficient

The algorithm to rank results for 'degree', 'betweenness centrality', 'closeness centrality' and 'clustering coefficient' factors is similar in nature and the steps are described next:

1. If the number of result candidates is less than or equal to the size of a result set, i.e. if $\lambda \leq \rho$, then only one result set is possible and $RF(Q(u,q)) = RC(Q(u,q))$.
2. If the number of result candidates is greater than the result size set i.e. if $\lambda > \rho$, the results are ordered by the respective value (degree, betweenness centrality, closeness centrality or clustering coefficient value) of each user present in $RC(Q(u,q))$ and the user with highest value is ranked first. Multiple entries by a user of higher value are placed in the final result set before entries from a user with lower degree value are considered.

5.7.3 Time

The algorithm to rank results based on the 'time' factor is the simplest among all the factors. The set of information present in $RC(Q(u,q))$ is ordered according to their time-stamp value. The document shared most recently is ranked first followed by documents in decreasing value of time-stamp. The ordered set is finally used to construct α results sets and the final result set, $RF(Q(u,q))$.

This concludes our discussion on the ranking factors and the associated algorithms. In the next section, we discuss details about the implementation of the social search engine.

6 Social Search System Development

We built *InfoSearch* as a prototype social search engine over Facebook. *InfoSearch* is built as a Facebook application using the Facebook platform APIs and is available at `http://apps.facebook.com/infosearch`. Users are requested to authorize the application in order to use it. Once authorized, the three primary components of the application work together to deliver search results. The system architecture for the search engine is presented in Figure 7 and the components are described next.

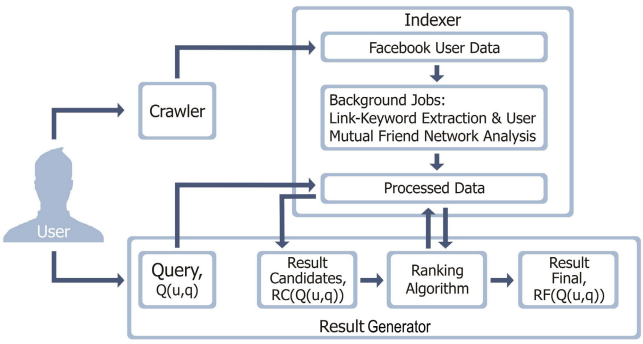


Fig. 7 Social Search Engine Architecture

6.1 Crawler

The purpose of the Crawler is to pull out information from the Facebook feed of each signed-in user using the Facebook API. The Facebook feed of a user consists of links, photos, and other updates from friends. In this work, the Crawler focuses on crawling the shared links to connect the web graph with the social graph. The Crawler is executed on a daily basis for each authorized user to retrieve the following data from their feed.

In our work, the Crawler employs the 'links' API provided by Facebook to crawl the various 'links' i.e. internet URLs shared by users on the Facebook platform. When called by the Crawler, the 'links' API returns a set of fields related to each link entry. Among the returned fields, we consider the following fields: a) 'id', b) 'from', c) 'link', d) 'name', e) 'description', f) 'message' and g) 'created_time' for the next component of our search engine. The Crawler also retrieves information about a user's friend list to build the ego and mutual friend network of a user. The Crawler uses the 'friends' and 'friends.getMutualFriends' API to retrieve information about the nodes and edges, respectively to build the ego network of a user. The Crawler also provides scope to expand our architecture to include other social network platforms by mapping the field lists of each returned link with fields used by the next two components of the architecture.

6.2 Indexer

The Indexer has two primary tasks. First, it analyzes the information retrieved by the Crawler to build an index of keywords for each shared URL. Second, the Indexer also performs the task of analyzing the mutual friend network of each user and build the corresponding user relationship data. Details of each task are described next.

Once the shared URLs are retrieved from the feed of each signed-in user, the next step is to build a keyword table for each URL with keywords extracted from the text retrieved from the URL. We use Yahoo!'s term extraction engine [27] for this purpose. The term extraction engine takes a string as input and outputs a result set of extracted terms. Additionally, we also use the Python-based topia.termextract library [22] to expand the keyword table. This library is based on text term extraction using the parts-of-speech tagging algorithm. We retrieve text from each URL and interpret the text using the aforementioned methods to finalize the set of keywords for each shared link. The second task of the Indexer is to analyze each signed-user's mutual friend network and determine the user property information (i.e. values of degree, betweenness centrality, etc) of each friend in the network. We use the 'R' implementation of 'kCliques' to build the social group information set [37].

To understand the impact of k in social group formation and accurate construction of social groups as users interact with *InfoSearch*, we built a Facebook application and surveyed users response for different values of k. We varied the value of k between 1 and 5 and asked users for their thoughts on the accuracy of social groups formed at different values of k. Conclusions from user responses were then used to determine the appropriate value of k for final result formulation in *InfoSearch*. In

our current implementation, we use a value of k equal to 3 to generate results for queries. We discuss details of this application and user feedback in Section 8.1.

The Indexer accomplishes the above two tasks by running data analytics background jobs on the raw data crawled from Facebook. The final processed data subsequently interacts with the third and final component, the result generator which is described next.

6.3 Result Generator

This is the final component in the system development. The purpose of this component is to a) process the user input query(ies), b) determine the result candidates, and c) formulate the final result set. In the first step, the user enters a query through the search engine web interface. At this step, users are also given the option to select their preferred way of ranking the possible results. In the next step, all documents related to the input query that originated from the friends of the user are retrieved. If no related documents are found and the query includes multiple keywords, the query is broken into multiple sub-queries and the search process is repeated to determine the related documents. If no documents are found at this stage, a 'no results found' message is sent to the user and the process stops. Otherwise, the set of related documents are promoted to potential result candidates and sent for processing by the ranking algorithms to determine the final result set. Based on the ranking factor selected by the user, the corresponding ranking algorithm is applied to the result candidates and the final result set is pushed forward to the application interface for display to the user.

In our current implementation, we set the number of results per result set, i.e. ρ as equal to 8. We implement a pagination style such that every result set of ρ results, i.e. $RS_1(Q(u,q)), RS_2(Q(u,q)), ..., RS_\alpha(Q(u,q))$ are placed on consecutive pages. Thus, the result sets are displayed to the user in the form of consecutive pages such that the first page displays the result set with highest value and decreases on later pages.

It is important to emphasize on the computational complexity involved in the final result construction at this stage. In traditional web search engines, final results for a variety of query keywords are pre-computed and result sets are cached for delivery to the user. In contrast, in a social search engine, as the number of result candidates and social context information present for each query varies, a result construction on the fly becomes a necessity and offers significant challenges to develop efficient and fast solutions. For example, during the process of determining final result sets using the 'diversity' factor, the number of sets possible for λ'_v unique users in the result candidates is $\binom{\lambda'_v}{\rho}$. The number of potential result sets for a relatively small number of unique users, say $\lambda'_v = 16$, the number of sets possible is $\binom{16}{8} = 12,870$. This number increases exponentially for higher number of users in result candidates. Iteration through such a large number of possible result sets takes a considerable amount of time and renders the search experience slow and inefficient. In the current development phase of *InfoSearch*, we focus on highlighting the challenges of building

social search engines and leave exploration of efficient algorithms for future works. However, as we will see during our user case studies in Section 8.2, Table 3, as the number of unique users in result candidates for a query can be substantially high, we resorted to using heuristic methods during the development process. In the final result set construction step, if the number of results, λ, and number of unique users, λ'_v, are both greater than the size of a result set, ρ, we consider the most recent 12 results sorted by 'time' and originating from 12 different users as constituent of the starting result candidates to construct the first final result set, introducing the next 8 results into result candidates list in addition to the remaining 4 results to generate the second final result set and so on. This step ensures we only have to construct $\binom{12}{8} = 495$ possible result sets before we decide the final result set at each iteration and users can enjoy the experience of receiving a quick result set for their query.

We also implement an additional feature to help users find information related to a specific friend or set of friends. This feature is implemented at the query step and the user has to specify the name of his/her friend(s) in conjunction with the query. In this particular situation, the retrieval process is limited to the set of information related to the specified user(s) only and the *time* factor is used to rank the results at this step. In the following section, we discuss the deployment of *InfoSearch* and present a few statistics on its current usage and performance.

7 User Statistics

We invited colleagues from our lab to use the application. *InfoSearch* was made available in March 2011. We present the following statistics analyzing the usage between March and December 2011. *InfoSearch* gained 25 signed-in users and through the signed-in user's Facebook feed, it has access to regular updates of 5,250 users. Each user has an average of 210 users in their ego network and their mutual friend graph has an average of 1414 edges.

During the time *InfoSearch* has been active, we have crawled links shared by 3,159 users. This is a very significant number because it tells us that, among the users *InfoSearch* has access to, 60% shared a web link with their friends in the social network. It is evident that the integration of web and social network graphs is taking place at a rapid pace and that the growth can have a significant impact on the way users search for information on the Internet.

The number of links shared by the users during this period is 31,075. The number of keywords extracted using the Yahoo! term extraction engine and the Python topia.termextract library is 1,065,835, which amounts to an average of 34 terms for each link. Additionally, we also consider the number of unique terms present in this pool to form a picture about the uniqueness in the shared content. We observe that the number of unique terms shared across all the links is 130,900, which results in an average of 4 terms per link. We next discuss case studies to understand the performance of social search engine results under different ranking factors and algorithms. We start by discussing results from our user study to determine the best value of k to formulate social groups.

8 User Studies

8.1 Social Group Analysis

An interpretation of the number and qualitative properties of social groups proposed by any method is a matter of subjective analysis to a particular user. In our definition of social groups, we mention the permissible upper-bound geodesic distance of k for two users to be a part a social group in the ego network of a user. A variation in the values of k can thus determine different social groups and consequently can lead to different (favorable or unfavorable) appreciation of the quantitative and qualitative properties of the social groups. To understand the value of k at which the users feel the social groups formed are best representative of their social network, we built a Facebook application[1] and sought out user feedback. We next describe the details.

A user must approve an application before the application can interact with the user. Once a user u approves the application to read their respective social data, information about their friends are fetched. In the second step, the fetched friend information is used to construct the mutual friend graph, $MF(u)$. Next, we construct social groups, $SG(u)$, starting with value of k equal to 1. We display the group formed to the user and sought out their feedback on two questions. In the first question, we asked users their opinion on the number of groups formed. The answer scores and their corresponding labels were a) 5, 'Too Many' b) 4, 'Many' c) 3, 'Perfect' d) 2, 'Less' and e) 1, 'Too Less'. In the second question, we asked participants of their feedback on the quality of the groups formed i.e. if the social groups formed were accurate representation of their real life groups. To obtain feedback for this question, we provide the participants the following scores along with the corresponding labels: a) 5, 'Yes, Perfectly' b) 4, 'To a good extent' c) 3, 'Average, could be better' d) 2, 'Too many related friends in separate groups' and e) 1, 'Too many unrelated friends in the same group'. We repeat the above step by incrementing the value of k for an upper limit of $k = 5$.

Table 1 User feedback scores on number of social groups detected

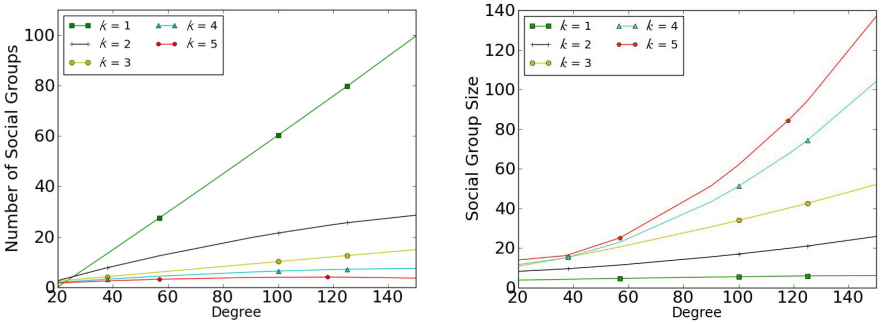
	Feedback Value	Standard Deviation
$k = 1$	3.84	0.84
$k = 2$	3.41	0.85
$k = 3$	3.03	0.67
$k = 4$	3.12	0.92
$k = 5$	2.25	1.14

Table 2 User feedback scores on quality of social groups detected

	Feedback Value	Standard Deviation
$k = 1$	3.54	0.91
$k = 2$	3.41	1.28
$k = 3$	3.80	1.31
$k = 4$	3.31	1.38
$k = 5$	2.88	1.46

Thirty users with varying size of friend lists signed into the application. Measurements from the logged-in user's egocentric networks are presented in Figure 8.

[1] The application is available at `http://apps.facebook.com/group_friends`



(a) Number of Social Groups **(b)** Average Size of Social Groups

Fig. 8 Logged-in user dataset analysis

We present results on the number of groups formed along with the average size of the groups for varying values of k for different user degrees in each of the figures. At $k = 1$, the number of groups formed grows linearly with the degree of the user. At higher values of k, we observe that the number of groups formed significantly drops with larger average group sizes. For example, at $k = 1$, number of groups is equal to 60 and average group size is equal to 5 for users with degree equal to 100. However, at $k = 2$, for the same users, the average size of the groups have risen to 15 while the number of groups has dropped to only 20. This happens because as we increase the value of k and correspondingly relax the requirements of member inclusion into a group, higher number of members are included into a single group including overlapping members. However, the more interesting observation comes when we compare the values obtained for $k = 4$ and $k = 5$. Since, we allow overlaps to exist across groups, if certain users exist over multiple groups for a given k, when we would allow a larger k, this overlapping user would cause the groups to collapse into a single group. Contrary to this assumption, we see only small changes in the values observed for $k = 4$ and $k = 5$ than for changes in values observed for $k = 3$ and $k = 4$, indicating that members in the mutual friend graph exist in small clusters that can be separated from each other at a certain cutoff level; $k = 4$ in this case.

Scores from the feedback analysis for the above two questions are presented in Table 1 and Table 2, respectively. We see the feedbacks on the number of social groups formed at $k = 3$ is approximately equal to 3, a score indicating a 'Perfect' division of the egocentric networks of the users into how they perceive their own social relationships to be divided in real life. It is also interesting to note in this section that the standard deviation at this instance is the least of all the feedbacks received.

User feedbacks on quality of the social groups formed are presented in Table 2. It is interesting to note that at values of k equal to $1, 2$ and 3, feedbacks indicate a score between 'Average, could be better' and 'To a good extend' indicating that the social groups detected are indeed accurate representation of how users perceive

their friends to be members of different sections in the real life. We thus conclude that a value of k equal to 3 is a good choice to compute social groups and form the basis of providing diversity based results to users in *InfoSearch* during any query.

8.2 Search Result Analysis

A social search engine generates unique results for every user. The subjective nature of results make it pointless to qualitatively compare with results from other search engines that generate identical results for all users. Thus, we cannot evaluate the results shown by *InfoSearch* for a query based on the results obtained from other web search engines. Instead, we focus on analyzing the impact of the ranking factors in the final result set for different users. We ask the following question: If a result set, $RS_i(Q(u,q))$, was generated using a particular ranking factor, what will be the result value of this result set for other ranking factors and how will the result value hold against similar values of result sets generated by other ranking factors? For example, in Figure 9, we compare diversity values of result sets generated by each ranking factor. We start by computing the final result for a given ranking factor and it's respective ranking algorithm. Once the final result has been computed and ranked result sets are available, we also evaluate the result value of each such result set for other ranking factors. Thus, for the example in Figure 9, we start by building the final result from the available result candidates for each ranking factor (i.e. 'diversity', 'degree', etc.) using the corresponding ranking algorithm. Once the result sets are ready, we compute the 'diversity' result value of the result set to compare and contrast the values in Figure 9. We perform similar actions to evaluate and discuss the result values for other ranking factors between Figures 10 and 14.

We perform user studies based on the information shared in the ego network of two authors of this work. The first author is labeled as 'userA' and the second author is labeled as 'userB'. userA has 246 members in his ego network. The number of edges shared between the members are 2235, that is, an average of 9.08 edges per member. userB has 1129 friends and the number of edges between the members are 7071, that results in an average of 6.26 members. Furthermore, the average clustering coefficient of each of the networks is 0.606 and 0.431 for 'userA' and 'userB' respectively. Aided with the visualizations presented in Figure 3, it is evident from these statistics that the respective ego networks are very different in topological characteristics and our next step is to understand how the ranking factors impact the final result set formation. We compare the results based on how the result value of each ranking factor holds up against the other ranking factors. For each ranking factor, we start by computing the final result set, $RF(Q(u,q))$. In the following discussions, we discuss the result value for the result set ranked highest i.e. we discuss the attributes of $RS_1(Q(u,q))$. We consider two queries for the user study: 'budget' and 'privacy' because of their relevancy among a large number of users in the social network. We present statistics related to each query for both users in Table 3.

The statistics in the table also illustrates the computation challenges to construct result set(s) in a social search engine setup as discussed in Section 6.3. In the above

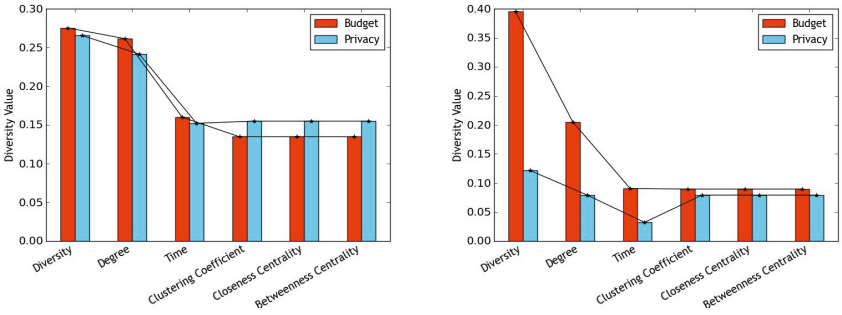
Table 3 Statistics on results candidates for query

	Query	Total number of results	Unique number of users sharing results
userA	'Privacy'	199	60
	'Budget'	30	13
userB	'Privacy'	1008	246
	'Budget'	121	49

examples, the worst case scenario is to construct a result set of ρ results from a possible result candidate of 1008 results originating from 246 unique users where we can construct $\binom{246}{8} = 2.96 \times 10^{14}$ sets to select the best result set. Clearly, this is a situation we want to avoid when we compute result for users on the fly. A consideration of this issue motivated us to exploit methods that will help us scale the computation and thus, finally in our result generation process, we consider only the 12 most recent result in the result candidate set to construct each result set. Next, we discuss the result values. We start by evaluating result values for the diversity factor.

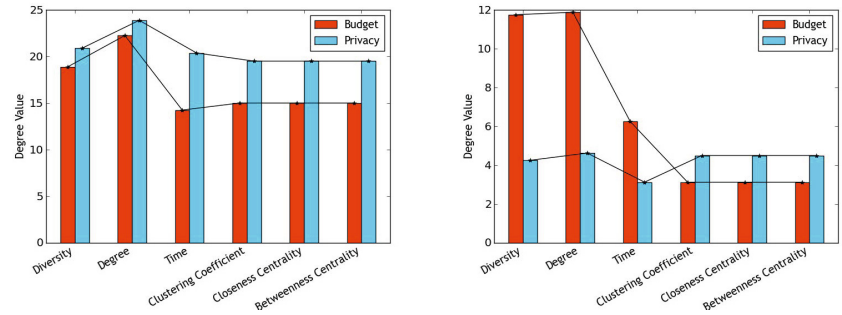
Diversity result value of a result set is given by $RV(RS(Q(u,q)),$ 'Diversity') and the values are plotted in Figure 9. The diversity values in the plot have been computed for $k = 3$. It is expected that the result sets produced using the diversity factor and it's corresponding ranking algorithms that aims to select the result set with the maximum value of diversity, has the highest values of diversity compared to the values of result sets generated by other factors. The plots confirm this hypothesis, however, it is interesting to note the difference in values of result sets computed using other factors. The consistency in decreasing values is best exemplified in the case of userB and query 'Budget'. userB's relatively large network (1129 friends) helps in retrieving results from a vast section of the network with high values of distance and corresponding diversity between the users. In contrast, diversity values for result sets formulated using the clustering and centrality measures are lowest in nature and shows signs of partiality in result formulation by contributions from only a few segments in the network.

We also observe the lowest diversity value related to any result set in the case for the result set computed by 'time' factor. In the context of a large number of possible result candidates for query 'privacy' for userB, diversity value is only 0.03 compared to the diversity value of 0.12 for the result set determined by the diversity factor itself. Similar patterns can also be observed for query 'Budget', values of 0.09 and 0.39 for results ranked by time and diversity respectively. We infer from this observation that information once shared by a member in a social group, has a tendency to flow between the members of the particular social group before it is shared by members of other social groups. This leads us to believe that result sets formed based on time of sharing can lead to information sources that originate within particular social groups and will have the lowest social diversity value. While the diversity based algorithm tries to maximize the value of social diversity in



(a) userA **(b)** userB

Fig. 9 Analysis for ranking factor 'Diversity'

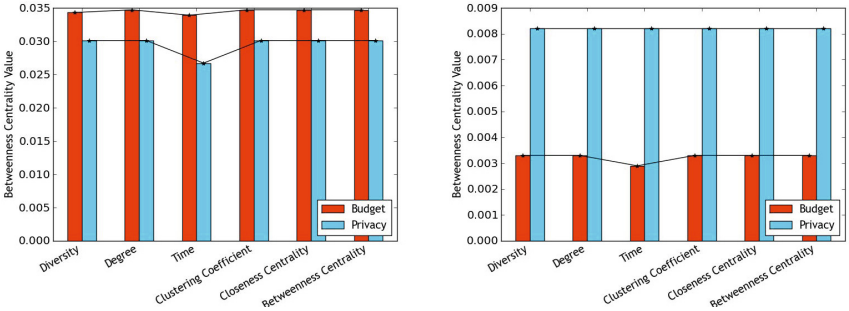


(a) userA **(b)** userB

Fig. 10 Analysis for ranking factor 'Degree'

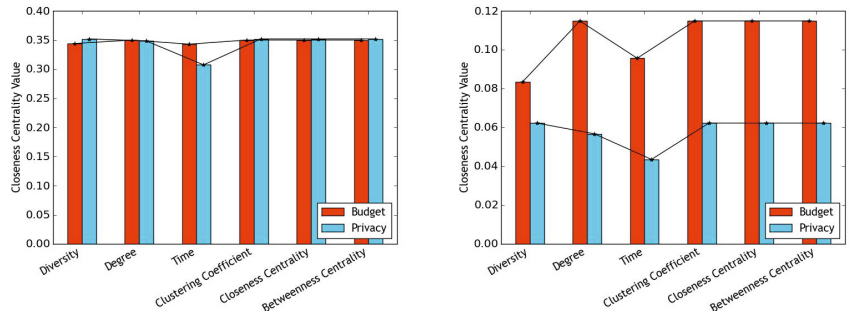
results, time factor, among the other factors mostly retrieve results that have the least value of social context present. We next discuss the degree values of result sets.

The degree value of a result set is given by $RV(RS(Q(u,q)),$'Degree'$)$ and the values are plotted in Figure 10. Similar to results ranked by 'diversity' factor which were expected to generate result sets with the highest values of diversity among any of the factors, the 'degree' value is also expected to be the highest among all the result sets for the result set generated by the 'degree' factor and it's corresponding ranking algorithm. The plots confirm the expectation. The values for queries 'Budget' and 'Privacy' for userA are 22.25 and 23.87 respectively compared to the second highest values generated by 'diversity' factor at 18.87 and 20.87, respectively. Similar trends are also observed for userB in Figure 10b. However, it is surprising to notice the difference between values when compared to the values generated by the 'degree' factor. The values for query 'Budget' for factors 'time', 'clustering coefficient', 'closeness centrality' and 'betweenness centrality', 14.25, 15, 15, 15 for userA and 6.25, 3.25, 3.12, 3.12 for userB, respectively, are significantly lower



(a) userA **(b)** userB

Fig. 11 Analysis for ranking factor 'Betweenness Centrality'

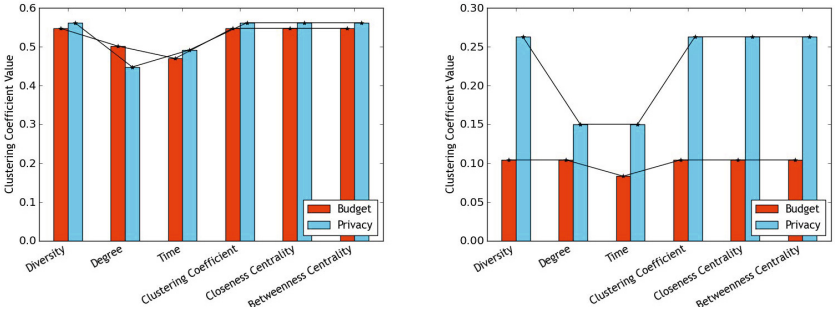


(a) userA **(b)** userB

Fig. 12 Analysis for ranking factor 'Closeness Centrality'

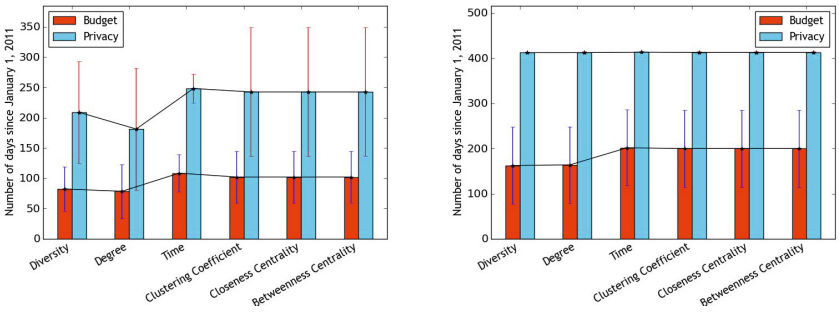
while the 'diversity' factor, 18.87 for userA and 11.75 for userB, is able to relatively match up with the values of the 'degree' factor, 22.25 for userA and 11.875 for userB. The relative matching in the results is significant because although developed for a different reason, the 'diversity' factor is successful in capturing the essence of the 'degree' factor and provide comparable values for the 'degree' metric, thus showcasing itself as a strong candidate to power social search engine ranking algorithms.

We next analyze the result values for the ranking factors based on centrality measures, i.e. 'betweenness centrality' and 'closeness centrality'. Analogous to the 'diversity' and 'degree' factors, result sets are also expected to have the maximum value of betweenness centrality and closeness centrality when the result sets were computed based on the respective factor and associated algorithm. We notice the phenomenon in the plots in Figures 11 and 12. Furthermore, we observe that the measures also generate similar result values for other factors. The highest value of betweenness centrality is observed to be 0.0345 for userA and 0.0033 for userB



(a) userA **(b)** userB

Fig. 13 Analysis for ranking factor 'Clustering Coefficient'



(a) userA **(b)** userB

Fig. 14 Analysis for ranking factor 'Time'

during analysis for query 'Budget' and 0.0301 for userA and 0.0082 for userB for query 'Privacy' the betweenness centrality factor (among other factors with equal values).

We see relatively low fluctuation in result values except for in the values generated by the 'time' factor based result set. The respective value for 'time' factor is 0.0339, 0.0029, 0.0267 and 0.0082, a percentage difference of 1.74%, 12.12%, 11.30% and 0%, respectively. This strengthens our previous argument that information has a tendency to flow between social groups before it spreads into a broader section of the ego network and a social search engine based solely on the 'time' factor thus fails to offer any advantage in terms of exploiting the prevalent social information. Next, we look at the 'clustering coefficient' result values. Unsurprisingly, we find a repeat of the same behavior here too with the 'time' factor offering the least value among all factors and failing to capture the social relationship based information into the result set. Finally, we investigate the 'time' characteristic of result sets.

We analyze time value of result sets from a reference date such that we can understand the relative 'freshness' of the data shared in the network. For example, if we observe two result set(s), we observe the average time-stamp value of shared information is 10 and 100 days in the future from the reference date, we term the result with the average time-stamp value of 100 days since the reference date to be more relevant and fresh to the user. Moreover, we also look at the standard deviation in the time-stamp values of the shared information and we term a result set with minimum values of deviation as the relevant result. Result values for the 'time' factor is presented in Figure 14.

The reference point for 'time' value analysis is placed on January 1^{st}, 2011 and the plots showcase number of days since the reference point. Thus, expectedly we observe the results generated based on 'time' factor has the maximum value compared to the respective value of result sets built using other factors. In the example of userA for query 'Budget', the value of result ranked using 'time' factor is 108 days whereas in contrast the lowest value is offered by the result set ranked by the 'degree' factor at 78 days. Furthermore, the corresponding deviation in the time-values are 30 days and 45 days respectively. Similar trends can also be observed in other cases. This happens because when results are ranked according to social relationship based factors, results that were shared a significantly long time ago are ranked higher in order to enrich the social value of the result set. Although not unexpected, a time based ranking of results thus, fails to accommodate social relationship semantics and provides a result set that is mostly partial to only a sub-section of the user's ego network. In the next section, we conclude our work with a discussion about future work.

9 Concluding Remarks

In this chapter, we described our efforts to build *InfoSearch* over the Facebook platform as a prototype social search engine and provide scope to users to search through the posts shared by their friends. In the process, we identified six important factors related to ranking search results for social search systems. Users can employ either one of the factors to rank results as they search through *InfoSearch*. Based on data collected through the Facebook feeds of two authors, we also performed user studies to understand the impact of ranking factors in the formation of result sets. We observed that 'time' based ranking of results, while providing the latest posts, fails to include sufficient social information in the result based on the value generated for both 'degree' and 'diversity' factors.

Among the factors based on semantics of social relationships between a user performing a query and a user sharing a piece of information, 'diversity' based factor provides sufficient social context into the result set as well as performs well in comparison to 'degree' factor to include time characteristics in the result set. We believe the area of social search engines has an immense potential in the area of information search and retrieval and we want to expand this work into multiple directions. First, we want to grow the usage of *InfoSearch* by inviting more users to use our system on

a regular basis and provide us feedback on their opinion about the quality of results formulated. Second, we want to extend the system architecture to include the scope of distributed databases and develop the application into a distributed system capable of handling thousands of queries at any given time. Third, we want to extend the factors involved in the ranking process to include other online social network platform focused factors like 'interaction intensity between users'. Finally, we aim to develop methodologies and standards to objectively evaluate social search engine results.

References

1. Adamic, L., Adar, E.: How to search a social network. Social Networks 27(3), 187–203 (2005), doi:10.1016/j.socnet.2005.01.007
2. Banerjee, A., Basu, S.: A social query model for decentralized search. In: Proceedings of the 2nd Workshop on Social Network Mining and Analysiss, vol. 124. ACM, New York (2008)
3. Bastian, M., Heymann, S., Jacomy, M.: Gephi: An open source software for exploring and manipulating networks. In: International AAAI Conference on Weblogs and Social Media (2009)
4. Baumes, J., Goldberg, M., Krishnamoorthy, M., Magdon-Ismail, M., Preston, N.: Finding communities by clustering a graph into overlapping subgraphs. In: International Conference on Applied Computing (2005)
5. Baumes, J., Goldberg, M., Magdon-Ismail, M.: Efficient identification of overlapping communities. In: Kantor, P., Muresan, G., Roberts, F., Zeng, D.D., Wang, F.-Y., Chen, H., Merkle, R.C. (eds.) ISI 2005. LNCS, vol. 3495, pp. 27–36. Springer, Heidelberg (2005)
6. Bhattacharyya, P., Rowe, J., Wu, S.F., Haigh, K., Lavesson, N., Johnson, H.: Your best might not be good enough: Ranking in collaborative social search engines. In: Proceedings of the 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (2011)
7. Borodin, A., Roberts, G.O., Rosenthal, J.S., Tsaparas, P.: Link analysis ranking: algorithms, theory, and experiments. ACM Transactions on Internet Technology 5(1), 231–297 (2005), doi:10.1145/1052934.1052942
8. Brandes, U.: A faster algorithm for betweenness centrality. Journal of Mathematical Sociology 25, 163–177 (2001)
9. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems 30(1-7), 107–117 (1998)
10. Cross, R., Parker, A., Borgatti, S.: A bird's-eye view: Using social network analysis to improve knowledge creation and sharing. IBM Institute for Business Value (2002)
11. Davitz, J., Yu, J., Basu, S., Gutelius, D., Harris, A.: iLink: search and routing in social networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 931–940. ACM (2007)
12. Derényi, I., Palla, G., Vicsek, T.: Clique percolation in random networks. Physical Review Letters 94(16), 160, 202 (2005)
13. Dhyani, D., Ng, W.K., Bhowmick, S.S.: A survey of Web metrics. ACM Computing Surveys 34(4), 469–503 (2002), doi:10.1145/592642.592645
14. Dodds, P.S., Muhamad, R., Watts, D.J.: An Experimental Study of Search in Global Social Networks. Science 301, 827–829 (2003)

15. Facebook: Introducing facebook graph search (2013), https://www.facebook.com/about/graphsearch
16. Fortunato, S.: Community detection in graphs. arXiv 906 (2009)
17. Girvan, M., Newman, M.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences 99(12), 7821 (2002)
18. Gregory, S.: A fast algorithm to find overlapping communities in networks. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part I. LNCS (LNAI), vol. 5211, pp. 408–423. Springer, Heidelberg (2008)
19. Gregory, S.: Finding Overlapping Communities Using Disjoint Community Detection Algorithms. In: Fortunato, S., Mangioni, G., Menezes, R., Nicosia, V. (eds.) Complex Networks. SCI, vol. 207, pp. 47–61. Springer, Heidelberg (2009)
20. Haynes, J., Perisic, I.: Mapping search relevance to social networks. In: Proceedings of the 3rd Workshop on Social Network Mining and Analysis - SNA-KDD 2009, vol. 9, pp. 1–7 (2009), doi:10.1145/1731011.1731013
21. Horowitz, D., Kamvar, S.D.: The anatomy of a large-scale social search engine. In: Proceedings of the 19th International Conference on World Wide Web - WWW 2010, p. 431 (2010), doi:10.1145/1772690.1772735
22. Index, P.P.: Content term extraction using pos tagging (2011), http://pypi.python.org/pypi/topia.termextract/
23. Lappas, T., Liu, K., Terzi, E.: Finding a team of experts in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 2009, p. 467 (2009), doi:10.1145/1557019.1557074
24. Marsden, P.: Egocentric and sociocentric measures of network centrality. Social Networks 24(4), 407–422 (2002)
25. Mike Cassidy, M.K.: An update to google social search (February 17, 2011), http://googleblog.blogspot.com/2011/02/update-to-google-social-search.html
26. Mislove, A., Gummadi, K., Druschel, P.: Exploiting social networks for internet search. In: 5th Workshop on Hot Topics in Networks (HotNets 2006), p. 79. Citeseer (2006)
27. Network, Y.D.: Term extraction documentation for yahoo! search (2011), http://developer.yahoo.com/search/content/V1/termExtraction.html
28. Newman, M.: Detecting community structure in networks. The European Physical Journal B-Condensed Matter and Complex Systems 38(2), 321–330 (2004)
29. Newman, M.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences 103(23), 8577 (2006)
30. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E 69(2), 026,113 (2004), doi:10.1103/PhysRevE.69.026113
31. Palla, G., Barabási, A., Vicsek, T.: Quantifying social group evolution. Nature-London 446(7136), 664 (2007)
32. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435, 814 (2005)
33. Plangprasopchok, A., Lerman, K.: Exploiting social annotation for automatic resource discovery. In: AAAI Workshop on Information Integration from the Web (2007)
34. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. Proceedings of the National Academy of Sciences 101(9), 2658 (2004)
35. Sabidussi, G.: The centrality index of a graph. Psychometrika 31(4), 581–603 (1966), http://www.springerlink.com/index/10.1007/BF02289527

36. Tyler, J., Wilkinson, D., Huberman, B.: Email as spectroscopy: Automated discovery of community structure within organizations. In: First International Conference on Communities and Technologies (2003)
37. Carey, V., Long, L., Gentleman, R.: Package rbgl (2011), http://cran.r-project.org/web/packages/RBGL/RBGL.pdf
38. Wasserman, S., Faust, K.: Social network analysis: Methods and applications. Cambridge university press (1994)
39. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature 393(6684), 440–442 (1998), http://dx.doi.org/10.1038/30918
40. Wingfield, N.: Facebook, microsoft deepen search ties (May 16, 2011), http://online.wsj.com/article/SB10001424052748703421204576327600877796140.html

Social Media in Disaster Relief

Usage Patterns, Data Mining Tools, and Current Research Directions

Peter M. Landwehr and Kathleen M. Carley

Abstract. As social media has become more integrated into peoples' daily lives, its users have begun turning to it in times of distress. People use Twitter, Facebook, YouTube, and other social media platforms to broadcast their needs, propagate rumors and news, and stay abreast of evolving crisis situations. Disaster relief organizations have begun to craft their efforts around pulling data about where aid is needed from social media and broadcasting their own needs and perceptions of the situation. They have begun deploying new software platforms to better analyze incoming data from social media, as well as to deploy new technologies to specifically harvest messages from disaster situations.

1 Introduction

In this chapter, we review the ways in which individuals and organizations have used social media in past disaster events and discuss ways in which the field will progress. In the first section, we cover the how both individuals and organizations have used social media in disaster situations. Our discussion emphasizes how both types of groups focus on searching for new information and disseminating information that they find to be useful. In general, facts about disasters collected from the small number of individuals located near the scene of a disaster are the most useful when dealing with specific disaster situations. Unfortunately, this data is rare and difficult to locate within the greater sea of social media postings related to the disaster.

We follow this by discussing a framework for considering how to analyze and use social media. This framework consists of several different use cases and

Peter M. Landwehr · Kathleen M. Carley
Carnegie Mellon University,
Pittsburgh, Pennsylvania
e-mail: {plandweh,kathleen.carley}@cs.cmu.edu

W.W. Chu (ed.), *Data Mining and Knowledge Discovery for Big Data*,
Studies in Big Data 1,
DOI: 10.1007/978-3-642-40837-3_7, © Springer-Verlag Berlin Heidelberg 2014

analytic steps: collecting social media data; managing a workflow for analyzing social media data; constructing a narrative from social media data; processing social media data to find relevant information; working with geolocation data; analyzing the text of social media postings; and broadcasting information using social media. Along with each step we provide reference examples of tools and libraries that can be used by analysts and first responders.

The chapter concludes with a section looking at current research into how we can better analyze and understand social media. Our discussion centers on methods for automatically classifying text and using visual analytics to gain new insights.

2 Usage Patterns in Disaster

The questions confronting people in a disaster are almost always the same: What happened? Are my friends and possessions safe? How can we remain safe? Social media is a new resource for addressing these old needs.

Locals at the site of the disaster who are posting information about what they are witnessing are in many ways the gold of the social media world, providing new, actionable information to their followers. They are few in number, and while their messages are sometimes reposted they often don't circulate broadly. Locating their content is an ongoing challenge akin to finding a needle in a haystack. Such local information can serve as an early alert system, leading traditional news sources [1].

While non-local users cannot provide reportage on the disaster, they can propagate local stories across the network and help them gain traction. By simply discussing a disaster or using hashtags associated with it they can contribute to other users' perception that the disaster is relevant. They can also collate data from other media sources, ferret out local users, and debunk false rumors as they begin to propagate. They can also serve as a workforce to sort through postings for the few that are cries for help, identify locations based on photos, find missing people in scanned video, and create maps where there are none.

Organizations fill a different role in the social media ecosystem. While individuals seek out information to preserve their own well-being, news media and aid groups use social media to help carry out their missions. Reporters look to social media to find stories and get feedback on their coverage. They and their parent organizations also often post links to breaking stories hosted on their own websites or being broadcast in the traditional media. Relief groups post requests for resources, announcements about their activities, and monitor social media for information they can use in their relief work.

Individuals within organizations are often charged with monitoring social media for any and all content that might be relevant to understanding the disaster as it relates to the organization's mission. This is a free-form search for information, conditioned only on the organization's role. It's similarly difficult to

constrain what an organization might post to their account beyond that it be relevant to this mission.

Along with press-releases and information about services, organizations may engage in "beaconing" behaviors by trying to solicit particular information or resources from individuals in their communities. Organizations will also carry out immediate dialogue with users, responding to their comments directly and via the same medium. Such efforts can help with the success of things like beaconing by making it clear that the organization takes the medium and its users seriously.

It is important to note that not all usage during a disaster is benign. Some social media users will start spreading false information to create additional panic. The occurrence of a cascade of damaging rumors (a "virtual firestorm") during a crisis can serve to undermine a first responder, hamper the relief effort, and lead to innocent victims being harmed [2]. Organizations have set up "fake" meeting places to identify those whom they wished to contain.

As we explore these different ways in which social media have been used in disaster, bear in mind that this framing is not necessarily crucial for any of these activities to occur. Individuals and organizations use social media to find and share information in standard contexts as well. In a disaster and its aftermath these activities are heightened in particular ways but should not be construed as necessarily restricted to it.

2.1 Individuals

Disasters rarely end instantaneously. Aftermaths can drag on for days, weeks, months, or years. (As we write this, three years after Haiti was struck by a devastating earthquake, thousands of individuals remain in tent cities [3].) Disaster researchers often divide disasters and disaster response into four phases: preparedness; response to the event; recovery, including rebuilding after the response; and mitigation, including enacting changes to minimize the impact of future events [4].

When people are confronted with a disaster they don't just seek to preserve their lives at a single critical moment. Users actively seek out information that can help them understand what's happening for a prolonged period of time. They try and connect with other members of local communities for support, aid, and understanding. Often, they will use technology to do so. ([5] as cited by [6].)

Shklovski et al. documented this process for individuals in California who were afflicted by wildfires in 2007 [6]. These fires dragged on for weeks, covering large swathes of rural countryside. Californians in at-risk areas found the news media unhelpful, citing a focus on stories about damage to celebrity homes. What locals wanted was general information about where fires were occurring and who was in danger. To combat this lack of knowledge, the Californians being studied had set up two different online forums for posting news and warnings. At the end of wildfire season, one of the subject forums was closed because it was no longer

useful. The other remained open as a community hub and remained part of its users' lives.

These researchers later witnessed a similar phenomenon among a community of musicians in New Orleans in the aftermath of Hurricane Katrina [7]. The musicians adopted SMS messaging, more regular cell phone use, and posting to online forums in order to stay in touch during the disaster. Like the Californians living in range of the wildfires, these New Orleans natives felt that the television media focused on the most dramatic aspects of the disaster while ignoring the majority of the afflicted. The victims used satellite images and message boards set up by the local newspaper to find information that was relevant to them. They turned to previously unused technologies to socialize in disaster, and in many cases adopted these new practices into their regular lives.

In both California and New Orleans, individuals turned to technological resources carry out established information seeking patterns via new media. Since these studies were carried out, we have seen the advent of Web 2.0 and the plethora of social media platforms that exist today. It is easier than ever to search the web for information about disaster, but filtering out rumor, falsehood, and off-topic discussion from the ocean of online content remains difficult. The outstanding research challenge remains helping people to find information they need and to post information so that it can be found.

While people don't intentionally confine themselves to a particular medium, they naturally favor those with which they are comfortable and those from which they believe they can gain more information. Since its introduction in 2007, Twitter has benefitted from generally positive media coverage [8]. Thanks to both this positive portrayal and its widespread adoption, the microblogging platform has become seen as an important source for disaster information. Leading up to Superstorm Sandy in 2012, blogs published guides for how to best search Twitter for data [9]. In the storm's wake, blogs and news media published stories about how much Twitter had been used [10, 11].

Despite the press coverage, Twitter isn't the dominant means of electronic communication. Its usage is the barest fraction of SMS and email [12]. While a personal email is often rich in meaningful content, Twitter's broadcast nature has meant that the relevant tweets sent during any event are buried in a sea of off-topic noise. Nonetheless, the ready availability of data, as well as the perception that the service is the "new thing" has made it a popular choice for academic research. Twitter is by no means insignificant – its millions of users are real- but it is perhaps overvalued. Even as we focus heavily on it in this chapter, we advise that you consider the platforms relative position and situate your findings correspondingly.

The ready availability of data from the platform has also made it a popular choice for academic research. This doesn't mean that Twitter is a particularly dominant communications platform: its usage is the barest fraction of SMS and email and it suffers as a data source from a great deal of noise generated by third party users. This also doesn't mean that Twitter should be dismissed as

insignificant. Rather, it highlights the fact that other platforms, especially SMS, should be given additional research more in keeping with their usage patterns.

The vast pool of research on how Twitter has been used outside of disaster is generally beyond this chapter's scope. However, it is useful for understanding the how the service has been generally used, so we provide a brief overview here.

Kwak et al. collected a very large corpus of Twitter users, tweets, trending topics, and social relations between users, and provide a large collection of summary statistics for each. The researchers make a variety of observations, not least of which is that there is little overlap in their data between the most followed users on Twitter and the users who are most retweeted. They also find that following has a low degree of reciprocity, and that users who follow each other tend to be in the same time zone [13]. Java et al. have used network methods to analyze Twitter to try and identify meaningful user communities. In the process, they categorized the bulk of twitter interactions as consisting of "Daily Chatter" (descriptions of routine life), conversations, information sharing, and reporting news. They also characterized users as primarily being defined as information sources, information seekers, and friends [14]. Naaman et al. collected tweets from approximately 125000 users over a prolonged period, developed nine overlapping categories for the tweets, and then identified two clusters of users: meformers, who often broadcast personal information, and informers, who generally shared different types of information [15]. Bakshy et al. tried to identify how one could successfully inject a particular idea into Twitter by influencing a particular user. The researchers consider a user to have "influence" based on when users retweet a URL that they have posted; the researchers caution that this requires a relatively strong signal to detect influence, but is also precisely measurable. While they identified certain users as possessing influence and causing cascades of information, they found it difficult to predict when a cascade would occur or which of these potentials might cause a cascade. The researchers concluded that the most cost-effective for propagating a particular URL or idea on twitter would be to seed many non-influential users. These users would have the potential to create many small information cascades which might then add up to one of relatively rare large cascades [16].

Research on how Twitter is used in disaster often takes the form of looking at data collected from a particular subset of users commenting on a disaster and looks at the particular features of their discussion. For example, Starbird et al. attempted to understand usage patterns during the 2009 Red River flood by qualitatively analyzing tweets collected during the flood period that used the terms "red river" and "redriver". The researchers identified two overlapping types of useful tweets by users: generative and synthetic. Generative tweets introduce new information via description of lived experience or factual commentary on an extant tweet [17]. Synthetic tweets pull in a variety of outside information and repackage it specifically for Twitter: a 140-character summary of a news story, for example, as might be produced by a news organization. While the authors noted other types of tweets, the generative and synthetic made up the kernel of the useful data that arrived during the disaster. Original tweets are also hard to find. They

made up less than 10% of the sample used by the researchers, and more than 80% of that small number were produced by individuals located within 6 hours driving time of the afflicted area.

Similarly, Sinappan et al. attempted to categorize tweets broadcast by Australians during the 2009 Black Saturday brush fires. Using another search-based approach, the authors coded the tweets using a modified version of Naaman et al.'s general tweet categorization scheme specifically for disasters. When looking at 1684 tweets captured, the researchers found that only 5% contained directly actionable information [18]. Similarly, only 4% of messages posted to the Chinese microblog service Sina Weibo after the Yushu Earthquake in 2010 related to actions that individuals could or needed to take [19]. Roughly 25% of the messages were tied to situation updates about Yushu, but a large number of them were from secondary sources, something also true for the data analyzed by Sinappan et al.

In her thesis research, Sarah Vieweg developed a new categorization system for the subset of tweets that contain useful information. Synthesizing tweets from four disasters and referencing the disaster research literature, she created three overarching categories (social, built, and physical environment) for useful tweets. These categories are themselves split into 35 subcategories that capture the message's content [20]. Sample categories include "Status – Hazard", "Advice – Information Space", and "Evacuation".

These phenomena (a small number of actionable tweets, a small number of tweets from locals providing primary source data) play out repeatedly in analyses of different disasters. The non-local tweets often play secondary roles that are important in the broader context of the disaster. Sutton witnessed this when researching Twitter discussions of the 2008 spill of 5.4 million cubic yards of coal ash into the Tennessee River [21]. While many of the Twitterers were local, Sutton describes them as using the medium as a "grassroots mechanism" for getting national media attention aimed at the disaster. They are the non-influential users trying to start local cascades.

While demanding that a retweet must include a particular URL is stringent, the basic idea of using retweets as a measure of endorsement is natural and useful. Starbird & Palin found this to be true in the tweets broadcast during the 2011 Egyptian uprising [22]. (Bear in mind that an uprising differs from conventional disasters as it features two opposing forces, not simply people in distress.) The researchers draw the same lines that they have before between locals and non-locals and the relative importance of these tweets for knowing the condition on the ground. However, they also note that retweets make up 58% of the corpus they collected, and that the most circulated tweets were all variants of a particular "progress bar" meme about uninstalling a dictator or installing democracy. The meme originated with Twitterers outside of Cairo but eventually made its way into the city proper, getting picked up by other Twitterers nearer the heart of the protest. The researchers characterize the meme as the "complex contagion" described by Centola & Macy, arguing that the remixing of the different meme elements "show some degree of shared understanding of its purpose". ([23] as

cited by [22].) Meme retweeting and remixing kept the protesters involved, and can be seen as a way that even those outside a developing crisis situation can try to connect themselves to it, possibly as a precursor to additional action.

In addition to trying to raise awareness of the disaster, the Twitterers responding to the coal ash spill in Tennessee also tried to debunk false rumors about the disaster's scope. Indeed, the segment of the twitter community affiliated with any particular disaster has taken on the job of suppressing rumors relating to it. NPR Reporter Andy Carvin, who gained acclaim covering global news events solely on Twitter, has likened his many followers to the staff of a news room: "rather than having news staff fulfilling the roles of producers, editors, researchers, etc., I have my Twitter followers playing all of those roles [24]." Carvin relies on the platform to eventually provide him with access to domain experts who can verify content or help him debunk it. For example, Carvin was able to work with his followers to determine that a prominent blog ostensibly written by a Syrian lesbian documenting the local unrest was actually a hoax [25]. Similarly, during Superstorm Sandy reporter Jack Stuef exposed user @comfortablysmug as spreading false information about what was happening in New York City. Many of @comfortablysmug's tweets were identified as false by other Twitter users, while Stuef found images from @comfortablysmug's Twitter profile in his YouTube and was able to determine the user's true identity [26, 27].

Mendoza et al. attempted to systematically analyze the practice of individual Twitterers debunking and supporting the various rumors that can arise as a disaster progresses [28]. The researchers identified tweets sent in the wake of the 2010 Chilean earthquake that been retweeted at least one thousand times and that were promulgating ideas externally verified as either true or false. They then looked at the responses that these tweets had elicited. None of the verified truths were substantially contested by Twitterers, while all of the falsehoods saw a number of tweets denying their accuracy. Additionally, the falsehoods were generally affirmed as true in other tweets more rarely than were the genuine truths. The exception to this was the widespread reporting of looting in certain areas of Santiago; tweets about this topic performed similarly to the other true tweets. The suggests that while generally rumors can be expected to be called out on Twitter, particular types of rumor will still fly under the radar and be hard to detect. The study suggests that true reports of disasters will not be regarded as controversial, which may be useful in automatically confirming their accuracy from social media data.

Contra the Mendoza et al. study, however, we emphasize that even if eventually corrected, falsehoods have been propagated on social networks for long enough to enter the mass media. @comfortablysmug's stories of flooding at the NYSE were rebroadcast by several major news outlets before Stuef outed him. In the aftermath of the 2013 Boston Marathon Bombings, Twitter users and Redditors incorrectly identified a missing Brown University student and individual mentioned on a police scanner as the bombing suspects [29, 30]. This caused a brief but potent online witch-hunt for which Reddit administration apologized [31]. The Boston

Police, which has made extensive use of Twitter before and after the bombing, published the facts of the case to the platform to counter the rumors [32].

We have mentioned that Twitter has gotten a great deal of research attention relative to other media used in disaster. While it remains the focus of this chapter, it is important that we acknowledge the ways that individuals are leveraging other social media in these circumstances. To simply focus on Twitter, when the reality is that an individual equipped with a smartphone can already function on any number of social media platforms at once. Technologies for analyzing social media will not remain confined to a single platform but will exploit as many as possible for data. They will leverage not just Twitter, but also RSS feeds, Facebook, SMS, Sina Weibo, Four Square, and others from a variety of different nations.

As mentioned at the start of this section, musicians in New Orleans adopted SMS messaging in the aftermath of Katrina to stay in touch. SMS's ability to directly connect individuals and the widespread availability of the technology on low-tech cellphones has made it critical in emergency situations. In the immediate aftermath of the 2010 Haiti Earthquake, a small group of actors from relief organizations and the US Government got DigiCel, Haiti's main cellular service provider, to reserve the SMS short code 4636 as a dedicated number for processing distress messages. These messages were archived and translated into English by Haitian expatriates mobilized by the organizers of "Mission 4636". Both expatriates and Haitians still on the island worked to promote the short code as a useful resource. By Week 3, Mission 4636 was dealing with such a volume of messages that it began working with the CrowdFlower and Samasource crowdsourcing platforms to better coordinate message translation [33].

From our perspective on how individuals use social media, SMS was key in this disaster because significant numbers of Haitians used low-tech cellphones that could access SMS services in the wake of the earthquake, and because the SMS infrastructure itself was still working. In that sense, it was the right medium for the time. AS has occurred with Twitter data in other disasters, the SMS data was rife with falsehoods despite being sent to a dedicated help line. According to the Harvard Humanitarian Initiative's (HHI's) study of relief organization responses to the Haiti earthquake, perhaps as many as 70% of the 4636 messages contained errors, such as requests to locate victims by people who knew the victims to be dead [34].

Photo sharing during disaster has also seen some degree of academic study, though more work is needed. In 2008, Liu et al. looked at how Flickr had been used in response to seven different disasters [35]. They observed that individuals were posting photos of damaged areas for a variety of different reasons, united by an over-arching theme of documenting the crisis. The different photographs can generally be categorized as depicting a particular event, capturing on-line social convergence (e.g. screen shots of Facebook posts), listing the missing, and showing personal belongings (taken for inventory purposes).

Flickr can be understood as fulfilling some of the same needs as text-based services: individuals post representative images of disaster sharing information

about what they understand the situation to be. It's used to help communities organize and share information. It's also used to serve practical, individual needs, such as inventorying possessions. The authors tie this back to the medium itself: a photograph is a richer data source than a 140 character message. Twitter isn't an efficient tool for cataloging possessions.

Regardless of precise intent, when placed in disaster situations individuals broadcast and examine data using social media. Academic research has tried to categorize these usages, often noting that actionable information is hard to find, that information can be false, and that primary source information from local users can be rare. Additionally, platform-specific practices can potentially be subverted for additional information: we can characterize true and false statements seen during a disaster based on the number of debunking statements noticed in response. We can infer that photographs taken in disasters of peoples' possessions are being used to inventory property.

Just as social media are being leveraged by individuals during disasters, so too are they being used by relief organizations, both government affiliated and independent. While the specific purposes behind the uses may be different and complementary, the uses of the platforms are similar. We discuss these in the next section.

2.2 Organizations

First responder organizations, which include government agencies, police, firemen, medical and public health organizations, military responders and not-for-profits play critical roles in disaster response. These groups generally have access to data and analytical tools that are not available to the public, as well as the resources needed to rescue and aid individuals who are in distress. While people may distrust the accounts of unknown strangers reporting rumors, these organizations have established brands that often temper or heighten critical attitudes towards their own postings. First responder organizations are increasingly turning to social media to identify actionable needs and orient the response, gauge the scope of impact of the event, provide information to the public, track an mitigate firestorms and counter false information, and to try to identify potential secondary disasters before they occur [36].

Where social media has provided individuals with new spaces in which to mingle and interact, it has provided organizations with new spaces in which to research ongoing disasters and communicate with both victims and the general public. St. Denis et al. explored this phenomenon by looking at how a Virtual Operations Support Team (VOST) dealt with the 2011 Shadow Lake Fire [37]. The concept of the VOST was developed by emergency manager Jeff Phillips as a way for an organization to coordinate its responses on and to social media coverage of a disaster, and has been propagated by other emergency managers [38–40]. According to Phillips, the VOST should "integrate[e] 'trusted agents' into [emergency management] operations by creating a virtual team whose focus

is to establish and monitor social media communication, manage communication channels with the public, and handle matters that can be executed remotely through digital means such as the management of donations or volunteers." This list of the roles taken on by the VOST encapsulates the ways that organizations active in the relief sphere must address social media when disasters strike.

After the start of the 2011 Shadow Lake Fire, the Portland branch of the National Incident Management Organization (NIMO) recruited an all-volunteer VOST in order to help coordinate their online responses. In their postmortem interviews, St. Denis and her fellow researchers found that the groups settled into a routine: NIMO would draft a press release each evening, consult with the VOST about the need for updates or amendments in the morning, and then release the statement with according changes. Meanwhile, the VOST took charge of updating a blog, Facebook page, and Twitter account set up by NIMO to provide updates on the fire. The VOST provided feedback to Facebook users who posted to group's wall, relaying information back to NIMO and did its best to maintain some sense of community among those coming to the page. Still, according to Eriksen, the VOST's most important accomplishment was locating a blogger who was concerned about fire trucks getting routed over unsuitable back roads. This blogger possessed niche, critical knowledge that couldn't have been found without the VOST, and NIMO was able to contact the blogger directly to get more information from him.

During the 2011 London riots, local police authorities used Twitter as a way to communicate with citizens. The bulk of the posted tweets, as analyzed by Panagiotopolous et al., encouraged people to participate in "cleanup" activities after the riots head ended, commented on how well communities were doing in coming together after the riots had ended, and described the situation on the ground; they also posted requests for information, albeit much more rarely. The researchers suggest that the authorities' posting about clean up actions may have played a role in getting the public out to help clean, though definitively proving this is beyond the paper's scope [41].

When Sarcevic et al. examined the practices of individuals affiliated with medical organizations using Twitter in Haiti, they observed a widespread phenomenon they termed "beaconing": the broadcasting of requests for information or material to Twitter because of uncertainty about how to obtain them [42]. While this practice is also used by individuals as a general facet of information-seeking, its application by individuals affiliated with organizations tied to the crisis is important because it suggests that the organization itself has a need that it can't address internally or through established contacts.

The Haiti earthquake itself was a critical proving ground for social media's use in disaster. In its immediate wake Haiti saw an influx of aid workers and organizations, many of which planned to use sophisticated technological solutions in order to help provide disaster relief. Meanwhile, other volunteer organizations operating remotely helped to collate social media data arriving from victims, analyze it, and get the results to other organizations on Haiti.

In the previous section, we briefly mentioned the Harvard Humanitarian Initiative's (HHI's) study of responses to the Haiti earthquake. This study focused on the responses of and relationship between official relief organizations (such as those operated by the UN) and these "volunteer and technical communities" (V&TCs), an umbrella term covering both volunteer, not-for-profit, and for-profit groups active in the disaster [34]. V&TCs outside of Haiti played a key role parsing and analyzing local distress. That said, the critical problem on the ground in Haiti was lack of access to the Internet and the aid organizations' lack of a single, unifying platform. Difficulty integrating the output of different programs made it hard to combine results and merge workflows. Unlike with the VOST, the interface between the V&TCs and humanitarian organizations was less well defined.

We mentioned Mission 4636 in the previous section as an example of how individuals used social media during disaster. It bears revisiting here from an organizational standpoint, as the project fits the HHI's description of a V&TC, and played a key role in addressing the disaster. It was also essentially a one-time effort; while it can be replicated, these particular volunteers have separated.

Ralph Munro, one of the lead organizers of Mission 4636, has emphasized the front-facing aspect of 4636. He stresses that the project's success was due to its being a largely Haitian initiative. Without a robust group of Haitian expatriates, neither the back-end translation nor the SMS shortcode would have been useful. Indeed, he suggests that the primary role of social media other than SMS during the crisis was as a recruiting and advertising platform. Volunteers working for 4636 claimed to have been posting alerts about the project to Facebook so often that they were being threatened with bans for acting like spammers [43]. If Mission 4636 wished to reorganize, the leaders would need to turn to social media to again advertise the service and recruit volunteers for support.

The critical accomplishments of Mission 4636 were getting promoted to the Haitian community for local use, organizing and motivating volunteers, and providing a consistent pipeline of data to other V&TCs and relief organizations. It was specifically intended to connect victims and relief organizations; they provided little in the way of feedback to those outside of the relief loop. While victims were aware of Mission 4636 through the existence of the short code, other relief groups operating off the ground were effectively individual to Haitians, interacting only with responders and the public. As such, we want to briefly highlight Haiti Ushahidi, one of the V&TCs that received data from Mission 4636 and had no specific on-the-ground presence.

Ushahidi is an online platform to which individuals can post reports about distress in disaster situations. These reports can then be coded to fit particular categories and get pinned to locations on a map. Developed by Ory Okolloh for use during the Kenyan election crisis of 2009, the platform has since been deployed in other crisis situations [44]. Haiti Ushahidi was a particular instance of the Ushahidi platform set up by students at Tufts University. In marked contrast with Mission 4636, the Haiti Ushahidi project was less well known to average

Haitians. The maps produced by the project, however, were used by groups on the ground, and the project is often mentioned in close connection with Mission 4636 despite functioning independently [34]. The Haiti Ushahidi project presented a better public face than did Mission 4636 despite processing significantly less information. Further, by choosing to release a large subset of the disaster messages they were working with to the public, they helped put a face on the disaster in a way that the directed channel of SMS generally does not.

During the 2011 East Japan earthquake, a group of computer scientists and engineers formed a new, one-off aid group called ANPI_NLP to help get relevant information from tweets [45]. The researchers sought to parse tweets to find references to individuals who had gone missing or been found and then updating records in Google Person Finder, a missing persons database.

While a one-time effort like Mission 4636, in general ANPI_NLP is a V&TC effort in the Ushahidi Haiti mold. The researchers didn't present themselves in a way that would be perceived by the Japanese populace, and the results that they produced were stored in a database maintained by another V&TC (Google) which then dealt with relief organizations. Where ANPI_NLP differed from Ushahidi Haiti was in using up-to-date natural language processing to speed up the task of extracting information from tweets. The researchers rapidly created a pipeline for morphologically analyzing tweets and that both extracted named entities and locations, and classified the nature of the information expressed. The researchers had to perform some manual coding to create gold standard data and to vet results, but in general this was an automated process. They also point out the existence of problems similar to those described by Munro: translation is difficult, and human resources are critical. To the members of ANPI_NLP, the solution lies in better automated systems, and in tools that can more rapidly adapt to training data.

Our discussion of how organizations used social media is framed by the understanding that at some level they use social media the same way individuals do: they search for information, and contribute in order to participate in the conversation as fits their mission. Relief organizations must deal with the larger challenge of managing their presence in particular social media spaces and must understand the information that is coming to them via the different interfaces. One way to deal with this is through a dedicated group such as a VOST. Further, a host of small organizations are appearing to help work with social media data in particular crises, leveraging local knowledge and deploying new technologies. In its report, the HHI both noted the importance of this small organization in the Haitian Earthquake's aftermath while also airing the concerns of relief workers that in Haiti it that few of these tools had an established, dependable reputation.

It is impossible to review all of the tools that exist to help relief organizations and analysts mine meaning from social media data. In this next section, we approach this challenge by provide a useful framework for considering them, as well as descriptions of different tools that fit into the framework.

3 A Data Analytics Framework and Associated Tools

A variety of different tools have been and are being developed and deployed to help people and institutions work with social media. In this section, our primary focus is on those that are useful to analysts trying to mine social media data, particularly from Twitter, in disaster response situations. They range from libraries for programming languages to sophisticated GUI-based tools for responders who need quick assessments of information to platforms for recruiting other workers to help with tasks. Technically savvy responders and analysts chain the output produced by multiple tools together in order to create meaningful results.

In this section we describe a mix of these tools, broken down into a rough framework corresponding to different data mining tasks. Some of our distinctions would not exist in a conventional data mining text, but speak to our particular focus on social media in disaster. More specifically, in this section we discuss tools that support data collection; that support workflow management by way of third-party tool interoperability and enabling data retrieval; that support narrative construction from fragments of social media data; that support data processing for quantitative analysis and disaster response; that support pining social media data to maps based on geolocation data; and that support quantitative text analysis for use with machine learning. We also cover an additional, slightly different category: those used to broadcast on social media and to manage collection and publication of data. These won't be relevant to analysts or investigators looking at specific data published on social media services, but can be important for developing a holistic understanding of how different platforms are being used. Such tools are of particular interest to organizations doing their best to manage all aspects of their social media presence.

3.1 Data Collection

The central problem for researches wanting to take a quantitative, data mining approach to analyzing social media data is that it can be hard to obtain, store, or trade. In Twitter's case, few canonical data sets are available for study due to the company's restrictions on data storage. The corpus of tweets made available for the 2011 TREC Twitter competition[1] is a useful exception, but is limited in scope.

While archives of data are useful, analysts and relief workers also need methods for gleaning facts from Twitter in real time, that limit the amount of effort that they have to put into monitoring social media.

If an analyst is skilled at programming, the basic way of approaching social media data is to obtain it using a website's API. Twitter[2], Flickr[3], and many other social media platforms invite developers to access some portion of the website's

[1] http://trec.nist.gov/data/tweets/
[2] https://dev.twitter.com
[3] https://secure.flickr.com/services/api/

data programmatically. In the case of Twitter, roughly 1% of tweets are made available via the API. These limits depend on the site and API. In the case of Twitter, if a researcher wants to access a larger percentage of the Twitter stream than is available from the API they must deal with a data warehouse such as Spinn3r[4] or GNIP[5], which provide access blog data, the full Twitter stream, and a variety of other social media data. Limitations on data consumption via API are dependent on each site's Terms of Use.

If an analyst doesn't wish to work directly with the API they can turn to third party tools that will obtain the data for them and possibly provide some analysis. For example, TweetTracker, developed at Arizona State University, allows users to filter the stream of tweets in real-time based on keyword and location [46]. These tweets are then archived and stored for future analysis. The ORA network analysis tool[6] supports importing ego network data from both individual Facebook accounts and email boxes [47–49]. Ushahidi (the company behind the platform of the same name) has worked on its own tool, SwiftRiver[7], which uses crowd-based validation of data. As different RSS and Twitter streams are passed into the platform, users can remotely coordinate to annotate particular items regarding their accuracy or inaccuracy. Tools such as Social Radar, CRAFT, and SORASCS (discussed in more detail in the next section) provide platforms in which multiple tools can interoperate to create flexible disaster response systems and scalable data storage systems that support social media collection and analysis. Within these platforms, third-party tools can be used as components of larger workflows; a data collection tool such as TweetTracker can be paired with different analytical tools such as ORA to provide richer insights into data.

Yahoo!'s Pipes platform[8] is another option in this area, albeit a middle ground between pure coding and pure GUI solutions. It allows users to tie together a mixture of data from different RSS feeds, conditioned on different events occurring. Different pipes can be configured via an API or a graphical user interface. In a similar vein, CMU's Rapid Ethnographic Assessment (REA) system allows users to pull in data from Facebook, RSS feeds and Lexis-Nexis.

3.2 Workflow Management

Researchers wanting to take a "big data" approach to dealing with social media are faced with a plethora of challenges. As described above, social media data can be difficult to store, obtain, or trade. Additionally, the quantity of data makes it difficult to intuit critical patterns and characteristics when exploring. There is also no inherent guarantee of accuracy regarding the data's provenance. A fourth problem is that the data providers often do not maintain archives of the messages,

[4] http://spinn3r.com
[5] http://gnip.com
[6] http://www.casos.cs.cmu.edu/projects/ora/
[7] http://www.ushahidi.com/products/swiftriver-platform
[8] http://pipes.yahoo.com

so if all messages back to a particular date are needed, a database needs to be built and maintained with the relevant data and all associated meta-data. No one tool exists to address all these challenges. As we will see in subsequent parts of this section, many different tools are emerging to handle pieces of these tasks. Correspondingly, new tools are emerging to manage workflows between these more focused tools and the larger process of cleaning and analyzing social media data.

Social Radar, CRAFT, and SORASCS[9] are three tools that address this problem. Each is a web-based system that supports disaster response by helping analysts and responders chain together third-party tools for sequential data analyses. All three tools work by collecting social media data from a data warehouse or via a particular third-party tool that access a social media platform's API. The collected data is then archived and can be sent to different integrated tools (or sequences of tools) for further processing. These tools often address text-mining, network analysis, sentiment analysis, geo-spatial analysis, and visualization. While some are used interactively, others process data in a silent and opaque manner, converting them from one form to another.

Many of the tools incorporated in Social Radar, developed by MITRE, are aimed at detecting sentiment in Twitter [50, 51]. It provides a web interface for looking at trends in Twitter over time such as total sentiment (derived from the presence of particular sentiment charged terms), heavily retweeted users, and the prevalence of particular keywords.

CRAFT, developed by General Dynamics, is similar to these other workflow management tools but also supports an associated environment for general mashups. Files can be linked to Google Drive, and the platform supports a "playback" mode that allows disaster response training exercises to be run with archived social media data collected during prior disasters.

SORASCS, developed at the CASOS Center at Carnegie Mellon University, supports workflow management and sharing [52, 53]. Unlike CRAFT and Social Radar, which require outside tools to be integrated before deployment, SORASCS is an open architecture to which analysts can independently attach their own tools. It allows analysts to preserve, share, and modify particular workflows by saving them to files. SORASCS's open design would make it eligible to serve as a coordinating under-structure behind CRAFT or Social Radar. While the latter tools have stronger user interfaces from a crisis responder's perspective, they provide no facilities to preserve particular workflows for future use. Unlike CRAFT and Social Radar, SORASCS does not necessarily convert all data into a common database; the user is responsible for supplying a database component themselves. In a sense, SORASCS is at a different level of application hierarchy than CRAFT and Social Radar. It could serve as middleware using either platform as a front end. This could provide some benefits to analysts because Social Radar

[9] http://www.casos.cs.cmu.edu/projects/project.php?ID=20&Name=SORASCS

and CRAFT put the third-party tools in an open unstructured environment and don't support the development of automated and streamlined workflows as does SORASCS.

3.3 Narrative Construction

Social media data, composed of textual and other artifacts produced by millions of individuals, can be construed as a digital history of some aspects of the modern world. To parse the history of a particular disaster –or any other event- requires tools for composing narratives.

Appropriately aggregated data can naturally lend itself to this end. Indeed, data mining's focus on using big data demands that analysts use a combination of aggregation and culling for story-telling. Tools such as TweetTracker, ORA, and Social Radar can be used to plot the use of particular keywords and topics over time. As these terms fall into and out of use, they tell the story of what issues matter to particular users. ORA, as a network visualization tool, can be used to display the changing relationships between sets of entities graphically. In the case of Twitter data, this may refer to the relationships between individuals, individuals and the topics or keywords they have mentioned, and the topics and keywords themselves. These relationships can be rendered as a static snapshot or as a series of networks evolving over time. Newspapers have also turned to sophisticated visualization programming libraries in order to tell stories. The New York Times, for example, uses the D3.js JavaScript library[10] to create graphics for data-driven news stories [54–57].

It can also important to understand the course of an event through a collection of specific tweets or other social media postings, each of which provides a fragment of the story. Andy Carvin of NPR has made heavy use of the Storify[11] platform to collate individual social media postings to document news events [58]. Blogging tools such as Blogger[12], WordPress[13], and Tumblr[14] can be used solely for reposting entries, thus providing some measure of the service provided by Storify. Timeline publishing services such as Dipity[15] provide another alternative for describing the chronology of a particular event. By focusing on the specific rather than the aggregate these methods differ from conventional data mining approaches. However, given the emotional appeal of individual stories over general descriptions, analysts may want to direct some of their efforts towards finding those individual stories within the larger collection of data that can best serve as representatives of the whole.

[10] http://d3js.org
[11] http://storify.com
[12] http://www.blogger.com
[13] http://www.wordpress.com
[14] http://www.tumblr.com
[15] http://www.dipity.com

3.4 Data Processing for Relevance

Collected social media data must be processed to determine its meaning. There are a variety of ways in which data can be processed, and relief workers must focus on those that can best cater to a particular set of needs: predicting if a disaster is going to occur, assessing the scope of an ongoing disaster, identifying the key entities and actors involved in a disaster, and a variety of case-specific needs. Many of the tools we have already been discussed have been used by relief workers to address different parts of these challenges. Several critical methods for helping resolve this challenge are by leveraging keywords, annotating the data using crowdsourcing, using sentiment dictionaries to code text, and leveraging network analysis to identify key entities

3.4.1 Keyword-Based Labeling

Searching social media for particular disaster-related keywords is a simple but often effective technique for tracking disaster information. Because people often post news relating to disasters before it is reported in the mass media, a keyword search on a social network can provide early news about a disaster. Twitter determines its "trending topics" by processing large numbers of tweets to determine when keywords move into and out of currency [59]. When using a tool like TweetTracker to find disaster news, the underlying calls to the API are often simply looking for words mentioning certain keywords. Similarly, data warehouses like GNIP will often provide a separate listing of keywords that they have determined to be relevant in the requested tweets. While crude, individuals in distress who engaging in beaconing behaviors on Twitter to seek aid aren't trying to be deceitful and so will likely use the obvious and expected keywords. That said, keyword based searches have limits: individuals can make typos and spelling mistakes, and the particular keywords relevant to a disaster can evolve and change. It is a static approach to a dynamic situation.

For social media that isn't text based, an analyst can attempt to initially reduce the quantity of data by using any sort of qualitative textual label assigned to the particular object – the tags assigned to a Flickr image, for example. The assumption is that even if the choice of a particular tag or keyword will cause us to miss a few images, because the vast majority will be retained the amount of useful structure lost will be insignificant. This assumption needs additional empirical study. Martin et al found that tags are acceptable for the general flow but miss local information [60]. In crisis response, such local information may be critical.

3.4.2 Crowdsourcing-Based Labeling

While keyword-based coding can be useful for culling data down to general matches, the reduced data must often still be codified for relevance, actionability, and accuracy. This can be partially accomplished by automated processing of the data using trained machine learning algorithms, as in the ANPI_NLP project, but

is often handled manually. A human workforce with domain expertise can be used to provide sophisticated labeling to disaster data.

We've discussed the role played by Ushahidi in Haiti, but the platform bears revisiting here. Individual Ushahidi deployments can be used to categorize disaster reports and then post them to a map. This system provides a basic architecture for splitting the coding task across a group of individuals in order to streamline the completion of particular tasks. Analysts can also label messages post-facto, making Ushahidi a useful system for individuals seeking to place particular messages onto a map. The QuickNets platform[16], built using Ushahidi's source code as a base, further subdivides the crowdsourcing process in order to make coding tasks easier for individuals to complete.

When a crowdsourcing workforce for coding data must be raised quickly, the fastest method is to use a dedicated crowdsourcing platform. Amazon Mechanical Turk[17] is the archetypal example of an online labor market but there are many alternatives. As Mission 43636's popularity increased during the Haiti earthquake's aftermath, it switched from its informal organization system over to using CrowdFlower[18] and Samasource[19] to managing their many volunteer workers who spoke Kreyol and could translate the text.

Volunteers will often feel motivated to contribute time and energy to addressing disasters and working with disaster data, particularly for very large disasters. Dedicated communities of "Crisis Mappers" have formed around the idea of collecting geospatial data from afflicted regions and annotating it with relevant information[20]. Similarly, sparked.com[21] has focused on recruiting volunteers interested in contributing to meaningful causes. The best annotators for data may not be those obtained from a crowdsourcing marketplace but rather from within these and other communities of skilled volunteers with a specific investment in helping to resolve disasters.

3.4.3 Sentiment-Based Labeling

By measuring sentiment first responders can gauge the attitudes of populations to the ongoing disaster response and determine how they should adapt their activities. The field is very broad, and its state as of 2008 is described in detail by Pang and Lee [61]. The TweetTracker-ORA combination, Social Radar, Ushahidi, Google Crisis Maps, and ESRI ArcGIS are all being adapted to better incorporate methods for dealing with sentiment data. (We discuss the latter two programs further in the Geolocation section.)

[16] http://www.quick-nets.org
[17] http://www.mturk.com
[18] http://crowdflower.com
[19] http://samasource.org
[20] http://crisismappers.net
[21] http://sparked.com

While training an algorithm to code social media data with sentiment information tool may be beyond the scope of most analysts working during a disaster, a dictionary labeling of terms with defined sentiment analysis is not. While simple and potentially prone to error, the method is rapid and lends itself to the brevity of social media. Several different dictionaries exist, but notably ones include: Linguistic Inquiry and Word Count (LIWC), developed by a team of judges evaluating large lists of words[22]; the dictionaries of affective meanings collected by Heise when surveying different populations[23]; and SentiWordNet[24], developed by Baccianella, Esuli, and Sebastiani by choosing a collecting of seed words on WordNet and carrying out random walks across the network.

3.4.4 Network Analysis for Relevance

As noted at the start of this section, responders must be able to identify the key entities and players in responding to the disaster. While the coding methods described above can help users filter social media data, network methods can help analysts and responders look at the structure of social media data in order to infer relevant structural information in the communications themselves.

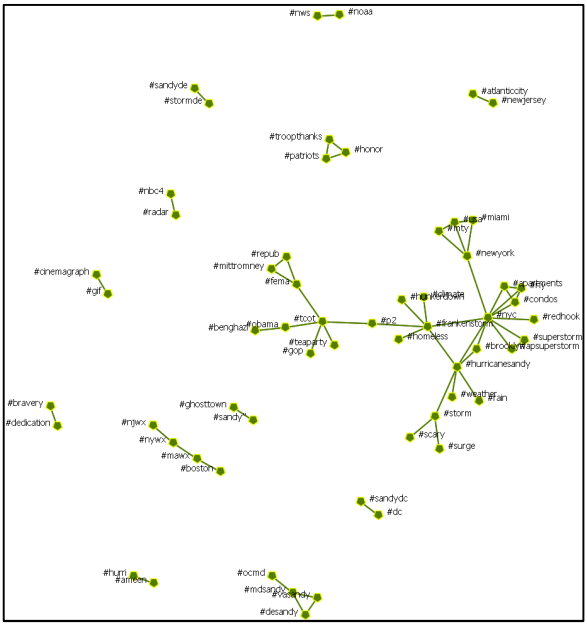


Fig. 1 A network diagram of hashtags used in New York City related to Hurricane Sandy on October 29, 2012. Hashtags are linked if they co-occurred at least five times. (Reproduced from http://www.pfeffer.at/sandy/)

Network data connects victims and responders to both locations and the needs they mentioned. These expressions can be used to identify critical actors and places that must be reached by responders. For ease of use, the ORA network

[22] http://www.liwc.net
[23] http://www.indiana.edu/~socpsy/ACT/data.html
[24] http://sentiwordnet.isti.cnr.it

analysis tool bakes a variety of useful social network metrics into reports to provide overall assessments of different situations. One of these reports has been designed to work with data from TweetTracker. It transforms the data to extract networks of retweets, hashtag co-occurrences, users and content, user and locations, and popular keyword distributions. It then processes these networks to identify influential Twitter users, core topics, and changing regions of concern. A similar technology has been built with ORA using REA for analyzing Lexis-Nexis data[25]. This technology has been use with respect to natural and man-initiated crises [62]. Its simplicity lends itself to first response. Figure 1 shows a network created from Twitter data using ORA.

For analysts who wish to go deeper and possibly conduct richer data mining on network structures, ORA supports the extraction of a variety of different social network metrics. Other GUI-based tools such as Gephi[26] and Cytoscape[27] also provide methods for analysts to approach the data, but a variety exist. In contrast, if the analyst wants to take a programming approach and develop their own network metrics they may want to work with the statnet package[28] for R or the NetworkX library[29] for Python.

3.5 Geolocation

The classic tool used for geo-spatial analysis in the crisis mapping area is ESRI ArcGIS[30]. ArcGIS is widely used by large number of response units including many police departments and military units. It supports pinning a variety of latitude/longitude data to maps, as well as visualizing changes in its distribution over time. In addition, ArcGIS supports a full complement of spatial analytics, and a layered visualization scheme. ArcGIS can import and export shapefiles, demarcations of geographic shapes, and KML, the XML-based markup language developed for use with Google Earth[31]. An increasing number of crisis-mapping tools, particularly those used by the large first responders, are exporting data in KML to support interoperability. Open source GIS tools are appearing that contain many of the features inherent in ArcGIS.

However, since the advent of Google Maps[32] eight years ago, an increasing number of crisis response tools are making use of it as an alternative. Since then, the quantity of data and tools available for working with geospatial data has only

[25] Illustrative results generating using ORA with Sandy data can be seen at `http://www.pfeffer.at/sandy/`
[26] `http://gephi.org`
[27] `http://www.cytoscape.org`
[28] `https://statnet.csde.washington.edu/trac`
[29] `http://networkx.lanl.gov`
[30] `http://www.esri.com/software/arcgis`
[31] `http://www.google.com/earth/index.html`
[32] `http://maps.google.com`

increased. According to the HHI's report, the V&TC community active in the Haiti earthquake particularly shone in its use of geospatial data. This is due to the dedicated work of the crisis mapping community and the willing participation of organizations with access to satellite imagery in crisis situations. In Haiti, a partnership between Google and GeoEye provided high-resolution images of the disaster area from above. With the right data, communities could annotate maps and workers on the ground could plan their activities.

Even when corporate entities do not provide such useful material, the community is able to rely on open platforms like the mapping site Open StreetMap[33], which has become a staple of the crisis community. All of the mapping data on OpenStreetMap has been contributed by volunteers; individuals upload GPS data to the site, and then annotate and edit it to keep it current. To deal with situations where internet access is limited or where users don't have access to GPS equipment, Michael Magurski released first the Walking Papers[34] and then Field Papers[35] tools. These allow users to download, print, annotate, and then upload the annotations to OpenStreetMap.

Google Maps has a growing presence in the crisis mapping community as well, and Google has itself devoted resources to creating maps specifically of crisis situations. They've provided crisis maps for specific incidents such as Superstorm Sandy that have been annotated with a variety of user data culled from the web [63]. Google also maintains a real-time crisis map[36] that uses similar culling of data to provide updates about potential and on-going crisis situations.

The TweetTracker tool developed at ASU visualizes extracted tweets on maps and lets users set spatial bounding boxes for selecting tweets by placing squares on maps. (See Figure 2 for an example of an exported map.) ORA also supports visualizing networks and other data on maps. It can import and export shape files and KML. In addition ORA allows users to cluster entities based on their particular region and then use that clustering as an element of a social network analysis.

While it would be incorrect to consider the challenge of properly representing data that has been connected with physical locations a solved problem, at this point there are a variety of tools that allow users to place information with specific latitudes and longitudes on a map. The research challenges are no longer about rendering these points in an informative manner. They are about developing new algorithms for deriving data from geographical clusters, and analyzing and forecasting the geographic distributions of social media postings in specific disasters.

[33] http://www.openstreetmap.org
[34] http://walking-papers.org
[35] http://fieldpapers.org
[36] http://google.org/crisismap/

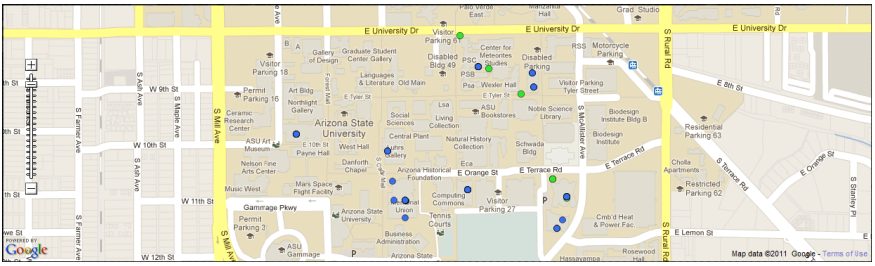


Fig. 2 A Google Map of locations from which tweets were received during a disaster simulation exercise. Tweet data was obtained using TweetTracker. See [64] for more information on the exercise.

3.6 Text Analysis

Covering the complete realm of automated text analysis is beyond this paper's scope. The field is immense and growing. Part of this expansion has incorporated the development of a variety of tools to make it easier to break down text and treat as quantitative data.

In general, toolkits in this area rely on the analyst being moderately familiar with programming languages. A GUI-based tool will be markedly easier for an analyst to work with if they lack the time or ability to code, or if they fail to thoroughly familiarize themselves with the language before a disaster strikes. They may also be difficult for first responders to integrate into a workflow, depending on the other tools they are using.

For Java, LingPipe[37] provides a useful library for tokenizing sentences, calculating sentiment, and stemming words, among other features. MALLET[38], developed by the University of Massachusetts at Amherst, provides some similar functions but focuses on using text for machine learning. If an analyst would rather not work with code directly, they can use the packages as precompiled binaries. Weka[39], a toolkit for running machine learning experiments developed at the University of Waikato, while distributed as an application with a GUI, can also be used as a Java library. While not specifically for text, like all machine learning packages it can be trained on textual features that have been quantified.

For Python, the Natural Language Toolkit[40] (NLTK) provides some of the same functionality as LingPipe. NLTK supports tokenization, stemming, text tagging and other standard natural language processing techniques. While incorporating some learning algorithms, analysts may want to investigate dedicated solutions like mlpy.[41]

[37] http://alias-i.com/lingpipe/index.html
[38] http://mallet.cs.umass.edu
[39] http://www.cs.waikato.ac.nz/ml/weka/
[40] http://nltk.org
[41] http://mlpy.sourceforge.net

A large variety of machine learning models for working with text have been implement as packages for the statistical language R. The tm package[42] bundles together standard natural language processing features for working with unstructured text. Once parsed, other packages oriented specifically towards data mining can be used with the text.

GUI-based tools for working with text data also exist, and may be easier for first responders to integrate into their workflows than a coding solution. One good example is AutoMap[43], a tool developed at the CASOS Center at Carnegie Mellon University that supports both GUI-based cleaning and an XML-based scripting language [65]. Like NLTK and other tools mentioned, AutoMap provides a number of methods for cleaning text documents like stemming words to their base forms, deleting stop words, and calculating the frequency of different multi-word sequences. AutoMap's scripting GUI makes it relatively easy to improvise and modify cleaning processes on the fly. The program has also been significantly integrated with ORA, allowing analysts to use network metrics to identify prominent co-occurrences of particular words or entities mentioned in documents. These networks of texts can also be visualized and –if referencing geospatial data- can be pinned to maps. This approach was used by a team of Arizona State University and Carnegie Mellon University researchers with data from Superstorm Sandy to compare the difference in content between Twitter and the news media.

One difficulty of working with text data posted to Twitter and other microblogs is that it often doesn't fit the conventions expected in ordinary text. When ANPI_NLP developed their named entity recognizer, for example, they had to first train a morphological analyzer to correctly split a tweet into names. Analysts generally expect to have to train their own parsers when working with microblog syntax. While not a general purpose named entity recognizer, Gimpel et al. have developed a tokenizer and part-of-speech tagger for Twitter[44] that has since been improved by Owaputi et al. [66, 67]. The POS tagger correctly classifies emoticons and the roles of various acronyms ("lol", "srsly"). While not critical for disaster on its own, in combination with the methods used by ANPI_NLP this could improve the speed and accuracy of other algorithms.

Translation of messages posted to social media in other countries remains a pressing problem, as we have discussed when describing the SMS messages translated by Mission 4636. This problem was also seen during the Egyptian Revolution and in the Yushu earthquake in China. While crowdsourcing markets are a proven solution for this problem, machine translation can also be used for potentially faster results. Google, for example, provides access to an API for automatic translation.[45] These will be less effective than native speakers of a particular language, but if it isn't possible to reasonably mobilize (or afford to mobilize) such a platform, machine translation is one possible alternative.

[42] http://tm.r-forge.r-project.org
[43] http://www.casos.cs.cmu.edu/projects/automap/
[44] http://www.ark.cs.cmu.edu/TweetNLP/
[45] http://developers.google.com/translate/

3.7 Broadcasting

Broadcasting tools largely fall outside of the practical use case for analysts. They are, however, relevant for first responders attempting to leverage social media, so we mention them here briefly. One example of a broadcasting tool is HootSuite[46], which allows users to manage profiles on multiple social networks, time the broadcasting of particular tweets, and perform some analytics similar to those mentioned in our discussion of tools that can be used for data retrieval. TweetDeck[47], an application provided by Twitter, provides a few similar functions but only for Twitter: users can use the software to control multiple Twitter accounts, subdivide followers into different groups, and schedule particular tweets to be posted at certain times.

Regardless of these relatively sophisticated tools, first responders will often interact with followers through the main interfaces of whatever particular social media service they are using. If Twitter, it may simply be their organization's account from the web, or the smartphone application of an organization member on the ground.

4 Research Directions

A common need felt by both people and organizations who turn to social media in disaster is knowing what is happening on the ground as rapidly as possible. Solving this problem has become the thrust of many ongoing research projects in the field. That being said, it is important to recognize that there are two very different audiences to whom this chapter is speaking: first-responders and disaster researchers. Each group needs different tools to pursue their own ends. First responders need easy to use simple tools with pre-defined workflows, specialized interfaces, dashboards, and maps. The time constraints of disasters prevent them from turning to powerful but less intuitive or rapid tools such as programming languages. In contrast, disaster researchers need to be able to use and create new methods, new types of visualizations, with workflows that they develop as part of the research. In this case, real-time performance is less important than the ability to perform sophisticated analyses. A particular type of research, translational research is needed in the disaster response area that supports the movement of those findings and tools discovered or invented by disaster researchers that are the most valuable to first responder from the laboratory into the field [68].

We now discuss two families of approaches to this challenge. We will begin with attempts to leverage machine learning and crowdsourcing to automatically classify individuals based on whether they provide useful information. We will then move on to discussing several different methods for visualizing social media data to provide immediate, intuitive feedback.

[46] `http://hootsuite.com`
[47] `http://tweetdeck.com`

In our section on individuals, we discussed the challenge of locating tweets that contributed to situational awareness and brought up the work of several researchers who have developed different categorizations for twitter messages. As mentioned earlier, Vieweg has developed a hierarchy of three overarching categories and 35 specific categories for situationally aware tweets [20]. She also experimented with the possibility of using VerbNet to automatically categorize tweets according to her model.

VerbNet is a lexicon of English verbs. It is a collection of verbs linked together based on a variety of different features including word senses, syntactic frames, and thematic roles, similar to both WordNet [69, 70]. Because tweets are generally only one or two sentences long, the verb can often be used as a critical identifier of a tweet's meaning. Vieweg identified nine VerbNet classes that were routinely present in her collection of situationally aware tweets. Testing on a large sample of both situationally aware and ordinary tweets, she found that 32.6% of a random sample of 4000 coded tweets contained SA data. While not perfect, systems incorporating these VerbNet codes is one step towards correctly validating data without human intervention.

Verma, along with Vieweg and several other researchers, tested the possibility of training a machine learning classifier to identify situationally aware tweets in a variety of disasters [71]. Working with the same Twitter data used by Vieweg, the researchers trained a Maximum Entropy classifier to reach between 84.1% and 88.8% accuracy on each data set. Prior to training, the researchers generated not only unigram and bigram features but also predicted subjectivity/objectivity, formal/informal register, and personal/impersonal tone as predicted by several other classifiers. The data were also coded with parts of speech tags, with a primary focus on identifying adjective use.

Similarly, Starbird et al. have experimented with using Support Vector Machines to try and identify the small number of individuals tweeting locally [72]. Using tweets broadcast during Occupy Wall Street, the researchers trained their classifier on a set of profile features such as times retweeted, number of followers, and whether stated profile location changed over time. Their final classifier still only correctly classifies 67.9% of those tweeting locally. While useful, there is still significant room for improvement.

Given the effectiveness of using crowdsourcing to classify disaster data, there is a strong argument to be made for feeding volunteers that has been classified with some level of error and expecting them to filter out the bad from the good. Another possibility is to integrate the volunteer crowd with the algorithm itself, having the users correct and retrain the algorithm on the fly. Settles has implemented an example of one such system, Dualist [73]. Users interact with the program by both coding documents with correct labels and by correcting labels assigned by the classifier. The importance of the accomplishment in this case is not simply the integration of a user into a conventional machine learning classifier but also the interface for the classification. This is not just a problem of algorithm design but also of constructing a useful interface.

The research projects we have discussed so far have focused on trying to find the useful tweets within the broader pool of data. Some researchers have taken an alternate approach, opting to find general information from the general mass of tweets. For example, Sakaki et al. have used Twitter data to detect earthquake epicenters [74]. Using the small number of tweets that have location data for references to earthquakes, they combine both support vector machines and particle filters to account for the uncertainty of the reported physical locations and then calculate the likely epicenter. Their system is effective but contingent on having a large number of tweets tagged with particular locations.

Similarly, the Google Flu Trends project[48] uses search queries made to Google to identify outbreaks of influenza [75]. Flu Trends is a specialized version of Google Trends in general, which tries to identify trending searches on Google just as Twitter tries to identify trending topics discussed by its users. The tool's success depends on both the large number of searches and also a lack of bias in the search data.

Going beyond microblog text, Fontugne et al. have investigated Flickr's potential for disaster detection [76]. The researchers have developed a prototype system that tracks uploaded photographs, highlighting particular labels that are being uploaded by multiple users at once. Their method captured large bursts of activity in Miyagi prefecture in Japan after the Tohoku earthquake. While the system shows potency as an alarm system, the researchers also point out that only 7% of the photographs taken within 24 hours of the earthquake were uploaded within that 24 hours. This is a dramatically different usage pattern from Twitter, and one that should impact proposed research to leverage Flickr data.

Visualizations of social media data is another ongoing challenge for helping users comprehend the sea of social media information. While crowdsourcing and machine learning can help us prepare data, it is often a visualization that helps individuals understand what the data is saying.

Word clouds have become a staple of modern visualization, as websites such as Wordle[49] have made them easy to create from any readily available text. Researchers have also looked into optimizing the patterns of words in word clouds to make them easier to interpret [77]. One notable example of their practical use is the Eddi system developed by Bernstein et al. [78]. Eddi assigns a set of topic labels to particular tweets by treating them as web search queries and then identifying prominent terms in the resultant searches. These topic labels are displayed as tag clouds that can be used to identify prominent subject of discussion. Note that Eddi's primary achievement is its insightful method of finding categorizations for tweets. However, the system relies on simple tag cloud systems as a key component of its visualization scheme.

ORA also incorporates a word cloud visualization. When fed longitudinal data, it allows the user to render a sequence of word clouds as networks that can be monitored changing over time. This is then supplemented with the ability to track

[48] http://www.google.org/flutrends/
[49] http://www.wordle.net/

the criticality of topics (e.g., Hashtags) and actors (e.g. Tweeters) in the different clouds, tracking how different topics have come into or dropped out of prominence over the course of an event.

Kas et al. have had success using tree maps to display tweets prominently associated with particular topics [79]. The researchers calculate the co-occurrences of all words in tweets collected on particular topics, filter words based on how often they co-occur, and then calculate popularity within particular topics. The most prominent topic keywords are then placed in a tree map, sized based on the square roots of their overall frequencies. The researchers carried out a small user study comparing the effectiveness of using word clouds and tree maps to display the ranked words from Twitter. They found that in general tree maps were significantly more useful; test subjects both better identified data presented in the tree maps relative to that presented in the word clouds but also significantly preferred using the tree map visualization.

Word clouds and tree maps are both relatively established forms of visualization. Both methods are constrained by only displaying a static view of the world. Social media, however, is often in flux. To understand a particular sequence of events it can be useful to get back to the originator of a particular comment, tweet, or image in order to understand how it has come to have significance. Shahaf et al. have developed a new, alternative visualization, the metro map, that addresses this problem for longer documents but has potential for being adapted to the Twitter space [80]. The metro map visualization links together sequences of documents based on shared features. Documents are represented as “stations”, like a traditional metro map, arranged roughly chronologically. The documents are tied together by directed “tracks” derived from the amount of overlap in coherence, coverage and connectivity in the actual text of the documents. Coherence is measured based on the overlapping content of articles, coverage as the number of topics mentioned across the collection of documents, and connectivity as the number of connections that exist.

The visualizations we have discussed have all focused on social media as a general source of data. We cannot point to particular examples of visualizations of social media data that are disaster specific. For example, there is no visualization scheme based on Vieweg’s categories for social media messages posted in disaster. This is a notable gap, and one that research needs to speak to. Visualizations that cater to a specific end can be much more effective than a general tool. For example, Kamvar and Harris’s “We Feel Fine”, a set of visualizations of individual emotions on Twitter, has caused users to engage in introspection and personal probing [81]. This is partly due to the text, which consists of personal statements, and partly due to the way in which the text has been represented. Visualizations designed to highlight the features of disaster could provoke similarly reach responses from users while also speaking to relief workers and analyst’s needs to understand the situation on the ground.

5 Conclusion

In this chapter, we have reviewed how social media is used in disaster by individuals, first responders, and disaster researchers. We have also introduced a variety of software tools that can be used by analysts to work with social media, the utility of which will vary depending on whether the analyst is a first responder or a disaster researcher. We have concluded with a discussion of several different directions in which some of the research on social media usage in disaster is currently heading.

For individuals impacted by the event, we have sought to highlight that in crises people turn to technology in order to find information and to find each other. Some malicious individuals will turn to social media to spread havoc. Social media platforms like Twitter become avenues for people to both seek information and express distress when they aren't certain where else to go. It's also a venue for publicizing disasters, for becoming involved in the large pool of social interactions surrounding a particular disaster, and for propagating false information related to a disaster.

For first responders, a critical concern is that a small amount of locally actionable information is being lost within a large pool of irrelevant noise. Locating this information remains a key challenge. Researchers have been and continue to develop schemes for categorizing the different types of messages sent in disasters. They have also looked at how users of social media respond to the propagation of falsehoods, and at how groups of organized individuals can be mobilized to crowdsource the categorization of distress messages.

When organizations impacted by the event turn to social media, they do so for similar reasons to individuals: to find new information about ongoing disasters, to communicate with individuals looking to them as authorities, and to stay in contact with followers. First responders and relief organizations use social media in these ways as well. In addition, they will use teams of individuals to monitor social media to find the critical pieces of information posted by niche users that they can use in planning disaster responses. They also post their own updates and information, providing authority in what is often a sea of rumors, and stopping firestorms of false information. Volunteer-based communities come together to analyze social media data, and a slew of new tools have been developed by organizations to help them turn this new data into actionable information.

While some tools for handling social media data have been developed specifically in the context of helping to resolve disasters, many others have been developed for the broader market. For example, the same company that developed the Ushahidi platform for collating disaster information also created the SwiftRiver tool for collating different streams of social media information. While SwiftRiver has definite application during disaster, it can also be used in broader contexts to track the development of any sort of chain of events. Maintaining the distinction between disaster-focused tools and those that are more generally applicable can be counter-productive. Rather, we propose considering tools as

being situated in one of seven categories: data collection, workflow management, narrative construction, data processing, geolocation, text analysis, and broadcasting.

Current research speaks to these issues by trying to speed up our ability to comprehend what is being said on social media. This often takes the form of attempting to fit automated classifiers to data sets, as with Vieweg's fitting of VerbNet terms to tweets sent in distress. Given the large pools of volunteers interested in working with crisis data as well as the many markets for crowdsourced labor, other research has looked at the possibility of combining machine learning algorithms with human vetting, either by using machine learning to reduce the size of the data such that it can be handled by humans or by using humans to interactively train the machine learning algorithm. Researchers have also approached this problem from the standpoint of visualization. An insightful visual representation can rapidly summarize a large quantity of social media data. While word clouds and tree maps have been demonstrated to be useful, and metro maps provide an avenue for moving forward, the field remains open for new ideas in visualization. There is also a need for visualizations of disaster data that emphasizes the disaster aspect in tandem with that of social media. Researchers should look to these studies of communication in disaster and craft new visualizations that specifically highlight those interactions.

These two research threads are not intended to be an exhaustive catalog of the future. Rather, they are two trends that have fallen out of some of the ways in which social media has been used by individuals and organizations. No matter how the field progresses, how social media is being used should remain its guiding star. Only by understanding the stresses on individuals and organizations during disaster can research help them improve.

Acknowledgments. The authors would like to thank Dr. Huan Liu of Arizona State University for his great help in bringing this chapter to fruition.

References

1. Pfeffer, J., Carley, K.M.: Social Networks, Social Media, Social Change. In: Nicholson, D.M., Schmorrow, D.D. (eds.) Adv. Des. Cross-Cult. Act. Part Ii, pp. 273–282. CRC Press (2012)
2. Pfeffer, J., Zorbach, T., Carley, K.M.: Understanding online firestorms: Negative word of mouth dynamics in social media networks. J. Mark. Commun. (2013)
3. Moloney, A.: Haiti must act to address housing crisis - Oxfam. Thompson Reuters Found. (2013)
4. Drabek, T.E.: Human System Responses to Disaster: An Inventory of Sociological Findings. Springer, New York (1986)
5. Dynes, R.R.: Organized Behavior in Disaster. Heath (1970)
6. Shklovski, I., Palen, L., Sutton, J.: Finding community through information and communication technology in disaster response. In: Proc. 2008 ACM Conf. Comput. Support. Coop. Work, pp. 127–136. ACM, San Diego (2008)

7. Shklovski, I., Burke, M., Kiesler, S., Kraut, R.: Technology Adoption and Use in the Aftermath of Hurricane Katrina in New Orleans. Am. Behav. Sci. 53, 1228–1246 (2010), doi:10.1177/0002764209356252
8. Arcenaux, N., Weiss, A.S.: Seems stupid until you try it: press coverage of Twitter, 2006-9. New Media Soc. 12, 1262–1279 (2010), doi:10.1177/1461444809360773
9. Sullivan, D.: Tracking Hurricane Sandy News Through Twitter. Mark. Land. (2012)
10. Carr, D.: How Hurricane Sandy Slapped the Sarcasm Out of Twitter, New York Media Decod. (2012)
11. Laird, S.: Sandy Sparks 20 Million Tweets. Mashable (2012)
12. Munro, R., Manning, C.D.: Short message communications: users, topics, and in-language processing. In: Proc. 2nd ACM Symp. Comput. Dev., pp. 1–10. ACM, Atlanta (2012)
13. Kwak, H., Lee, C., Park, H., Moon, S.: What is Twitter, a social network or a news media? In: Proc. 19th Int. Conf. World Wide Web, pp. 591–600. ACM, Raleigh (2010)
14. Java, A., Song, X., Finin, T., Tseng, B.: Why we twitter: understanding microblogging usage and communities. In: Proc. 9th Webkdd 1st Sna-Kdd 2007 Work. Web Min. Soc. Netw. Anal., pp. 56–65. ACM, San Jose (2007)
15. Naaman, M., Boase, J., Lai, C.-H.: Is it really about me?: message content in social awareness streams. In: Proc. 2010 ACM Conf. Comput. Support. Coop. Work, Cscw, pp. 189–192. ACM, Savannah (2010)
16. Bakshy, E., Hofman, J.M., Mason, W.A., Watts, D.J.: Everyone's an influencer: quantifying influence on twitter. In: Proc. Fourth ACM Int. Conf. Web Search Data Min., pp. 65–74. ACM, Hong Kong (2011)
17. Starbird, K., Palen, L., Hughes, A.L., Vieweg, S.E.: Chatter on the red: what hazards threat reveals about the social life of microblogged information. In: Proc. 2010 ACM Conf. Comput. Support. Coop. Work, pp. 241–250. ACM, Savannah (2010)
18. Sinnappan, S., Farrell, C., Stewart, E.: Priceless Tweets! A Study on Twitter Messages Posted During Crisis: Black Saturday. In: Proc. 2010 Australas. Conf. Inf. Syst. Acis (2010)
19. Qu, Y., Huang, C., Zhang, P., Zhang, J.: Microblogging after a major disaster in China: a case study of the 2010 Yushu earthquake. In: Proc. Acm 2011 Conf. Comput. Support. Coop. Work, Cscsw, pp. 25–34. ACM, Hangzhou (2011)
20. Vieweg, S.E.: Situational Awareness in Mass Emergency: A Behavioral and Linguistic Analysis of Microblogged Communications. University of Colorado at Boulder (2012)
21. Sutton, J.: Twittering Tennessee: Distributed Networks and Collaboration Following a Technological Disaster. In: Proc. 7th Int. Conf. Inf. Syst. Crisis Response Manag. (2010)
22. Starbird, K., Palen, L. (How) will the revolution be retweeted?: information diffusion and the 2011 Egyptian uprising. In: Proc. ACM 2012 Conf. Comput. Support. Coop. Work, Cscw, pp. 7–16. ACM, Seattle (2012)
23. Centola, D., Macy, M.: Complex Contagions and the Weakness of Long Ties. Am. J. Sociol. 113, 702–734 (2007)
24. NPR Staff, "Distant Witness": Social Media's "Journalism Revolution." Talk Naton (2013)
25. @TwitterMedia NPR's Andy Carvin Uses Twitter to Debunk A Hoax. #OnlyOnTwitter
26. Kaczynski, A.: How One Well-Connected Pseudonymous Twitter Spread Fake News About Hurricane Sandy. Buzzfeed Polit (2012)

27. Stuef, J.: The Man Behind @ComfortablySmug, Hurricane Sandy's Worst Twitter Villain. Buzzfeed Fwd. (2012)
28. Mendoza, M., Poblete, B., Castillo, C.: Twitter under crisis: can we trust what we RT? In: Proc. First Work. Soc. Media Anal. Soma, pp. 71–79. ACM, Washington D.C. (2010)
29. Madrigal, A.C.: It Wasn't Sunil Tripathi: The Anatomy of a Misinformation Disaster. The Atlantic (2013)
30. Weinstein, A.: Everybody Named the Wrong Boston Suspects Last Night and Promptly Forgot. Gawker (2013)
31. Martin, E.: Reflections on the Recent Boston Crisis. Reddit Blog (2013)
32. Keller, J.: How Boston Police Won the Twitter Wars During the Marathon Bomber Hunt. Bloom. Bussinessweek (2013)
33. Mission 4636, Collaborating organizations and History. Mission 4636 (2010)
34. Harvard Humanitarian Initiative, Disaster Relief 2.0: The future of Information Sharing in Humanitarian Emergencies. Harvard Humanitarian Initiative, UN Office for the Coordination of Humanitarian Affairs, United Nations Foundation (2011)
35. Liu, S.B., Palen, L., Sutton, J., et al.: In search of the bigger picture: The emergent role of online photo sharing in times of disaster. In: Proc. 5th Int. Conf. Inf. Syst. Crisis Response Manag. (2008)
36. Cohen, S.E.: Sandy Marked a Shift for Social Media Use in Disasters. Emerg. Manag. (2013)
37. St. Denis, L.A., Hughes, A.L., Palen, L.: Trial by Fire: The Deployment of Trusted Digital Volunteers in the 2011 Shadow Lake Fire. In: Proc. 9th Int. Conf. Inf. Syst. Crisis Response Manag. (2012)
38. Reuter, S.: What is a Virtual Operations Support Team? Idisaster 20 (2012)
39. Stephens, K.: Understanding VOSTs (Virtual Operations Support Teams) Hint: It's All About Trust. West. Mass Smem (2012)
40. VOSG.us, About. Virtual Oper. Support Group (2011)
41. Panagiotopoulos, P., Ziaee Bigdeli, A., Sams, S.: "5 Days in August" – How London Local Authorities Used Twitter During the 2011 Riots. In: Scholl, H.J., Janssen, M., Wimmer, M.A., Moe, C.E., Flak, L.S. (eds.) EGOV 2012. LNCS, vol. 7443, pp. 102–113. Springer, Heidelberg (2012)
42. Sarcevic, A., Palen, L., White, J., et al.: "Beacons of hope" in decentralized coordination: learning from on-the-ground medical twitterers during the 2010 Haiti earthquake. In: Proc. ACM 2012 Conf. Comput. Support. Coop. Work, Cscw, pp. 47–56. ACM, Seattle (2012)
43. Munro, R.: Crowdsourcing and the crisis-affected community: Lessons learned and looking forward from Mission 4636. Inf. Retr. 16, 210–266 (2013), doi:10.1007/s10791-012-9203-2
44. Okolloh, O.: Ushahidi, or "testimony": Web 2.0 tools for crowdsourcing crisis information. Particip. Learn. Action 59, 65–70 (2009)
45. Neubig, G., Yuichiroh, M., Masato, H., Koji, M.: Safety Information Mining — What can NLP do in a disaster—. In: Proc. 5th Int. Jt. Conf. Nat. Lang. Process. Asian Federation of Natural Language Processing, Chiang Mai, Thailand, pp. 965–973 (2011)
46. Kumar, S., Barbier, G., Abbasi, M.A., Liu, H.: TweetTracker: An Analysis Tool for Humanitarian and Disaster Relief. In: Proc. 2011 Int. Aaai Conf. Weblogs Soc. Media, pp. 661–662. AAAI, Barcelona (2011)

47. Carley, K.M., Reminga, J., Storrick, J., Columbus, D.: ORA User's Guide, Carnegie Mellon University, School of Computer Science, Institute for Software Research, Pittsburgh, Pennsylvania (2013)
48. Carley, K.M., Columbus, D.: Basic Lessons in ORA and AutoMap 2011. Carnegie Mellon University, Pittsburgh (2011)
49. Carley, K.M., Pfeffer, J.: Dynamic Network Analysis (DNA) and ORA. Adv. Des. Cross-Cult. Act. Part (2012)
50. Costa, B., Boiney, J.: Social Radar. MITRE, McLean, Virginia, USA (2012)
51. Mathieu, J., Fulk, M., Lorber, M., et al.: Social Radar Workflows, Dashboards, and Environments. MITRE, Bedford (2012)
52. Schmerl, B., Garlan, D., Dwivedi, V., et al.: SORASCS: a case study in SOA-based platform design for socio-cultural analysis. In: Proc. 33rd Int. Conf. Softw. Eng., pp. 643–652. ACM, Waikiki (2011)
53. Garlan, D., Schmerl, B., Dwivedi, V., et al.: Specifying Workflows in SORASCS to Automate and Share Common HSCB Processes. In: Proc. Hscb Focus. Integrating Soc. Sci. Theory Anal. Methods Oper. Use (2011), doi:10.1.1.190.7086
54. Bostock, M., Ogievetsky, V., Heer, J.: D3 Data-Driven Documents. IEEE Trans. Vis. Comput. Graph. 17, 2301–2309 (2011), doi:10.1109/TVCG.2011.185
55. Bostock, M., Carter, S.: Wind Speeds Along Hurricane Sandy's Path - Interactive Feature, New York (2012)
56. Bostock, M., Ericson, M., Leonhardt, D., Marsh, B.: Across U.S. Companies, Tax Rates Vary Greatly, New York (2013)
57. Bostock, M., Bradsher, K.: China Still Dominates, but Some Manufacturers Look Else-where, New York (2013)
58. Carvin, A.: Andy Carvin's Social Stories. Andy Carvins Soc. Stories
59. Lin, J., Snow, R., Morgan, W.: Smoothing techniques for adaptive online language models: topic tracking in tweet streams. In: Proc. 17th Acm Sigkdd Int. Conf. Knowl. Discov. Data Min., pp. 422–429. ACM, San Diego (2011)
60. Martin, M.K., Pfeffer, J., Carley, K.M.: Network text analysis of conceptual overlap in interviews, newspaper articles and keywords. Soc. Netw. Anal. Min. (forthcoming)
61. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. Now Publishers (2008)
62. Carley, K.M., Pfeffer, J., Morstatter, F., et al.: Near Real Time Assessment of Social Media Using Geo-Temporal Network Analytics. In: Proc. 2013 Ieeeacm Int. Conf. Adv. Soc. Networks Anal. Min. (2013)
63. Schroeder, S.: Google Launches Crisis Map for Hurricane Sandy. Mashable (2012)
64. Abbasi, M.-A., Kumar, S., Filho, J.A.A., Liu, H.: Lessons Learned in Using Social Media for Disaster Relief - ASU Crisis Response Game (2012)
65. Carley, K.M., Columbus, D., Landwehr, P.: AutoMap User's Guide, Carnegie-Mellon University, School of Computer Science, Institute for Software Research, Pittsburgh, Pennsylvania (2013)
66. Gimpel, K., Schneider, N., O'Connor, B., et al.: Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments. In: Proc. 49th Annu. Meet. Assoc. Comput. Linguist. Hum. Lang. Technol. (2011)
67. Owoputi, O., O'Connor, B., Dyer, C., et al.: Part-of-Speech Tagging for Twitter: Word Clusters and Other Advances. Carnegie Mellon University, Machine Learning Department, Pittsburgh, Pennsylvania, USA (2012)
68. Woolf, S.H.: The Meaning of Translational Research and Why It Matters. J. Am. Med. Assoc. 299, 211–213 (2008), doi:10.1001/jama.2007.26

69. Kipper Schuler, K.: VerbNet: A broad-coverage, comprehensive verb lexicon. University of Pennsylvania (2005)
70. Fellbaum, C.: WordNet: an electronic lexical database. The MIT Press (1998)
71. Verma, S., Vieweg, S.E., Corvey, W.J., et al.: Natural Language Processing to the Rescue? Extracting "Situational Awareness" Tweets During Mass Emergency. In: Proc. 2011 Int. Aaai Conf. Weblogs Soc. Media (2011)
72. Starbird, K., Muzny, G., Palen, L.: Learning from the Crowd: Collaborative Filtering Techniques for Identifying On-the-Ground Twitterers during Mass Disruptions. In: Proc. 9th Int. Conf. Inf. Syst. Crisis Response Manag. Iscram (2012)
73. Settles, B.: Closing the loop: fast, interactive semi-supervised annotation with queries on features and instances. In: Proc. Conf. Empir. Methods Nat. Lang., pp. 1467–1478. Association for Computational Linguistics, Edinburgh (2011)
74. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes Twitter users: real-time event detection by social sensors. Proc. 19th Int. Conf. World Wide Web, pp. 851–860. ACM, Raleigh (2010)
75. Carneiro, H.A., Mylonakis, E.: Google Trends: A Web-Based Tool for Real-Time Surveillance of Disease Outbreaks. Clin. Infect Dis. 49, 1557–1564 (2009), doi:10.1086/630200
76. Fontugne, R., Cho, K., Won, Y., Fukuda, K.: Disasters seen through Flickr cameras. In: Proc. Spec. Work. Internet Disasters, pp. 1–10. ACM, Tokyo (2011)
77. Rivadeneira, A.W., Gruen, D.M., Muller, M.J., Millen, D.R.: Getting our head in the clouds: toward evaluation studies of tagclouds. In: Proc. Sigchi Conf. Hum. Factors Comput. Syst., pp. 995–998. ACM, San Jose (2007)
78. Bernstein, M.S., Suh, B., Hong, L., et al.: Eddi: interactive topic-based browsing of social status streams. In: Proc. 23rd Annu. ACM Symp. User Interface Softw. Technol., pp. 303–312. ACM, New York (2010)
79. Kas, M., Suh, B.: Visual Summarization for Topical Clusters in Twitter Streams (forthcoming, 2013)
80. Shahaf, D., Guestrin, C., Horvitz, E.: Trains of thought: generating information maps. In: Proc. 21st Int. Conf. World Wide Web, pp. 899–908. ACM, Lyon (2012)
81. Kamvar, S.D., Harris, J.: We feel fine and searching the emotional web. In: Proc. Fourth ACM Int. Conf. Web Search Data Min., pp. 117–126. ACM, Hong Kong (2011)

A Generalized Approach for Social Network Integration and Analysis with Privacy Preservation

Chris Yang and Bhavani Thuraisingham

Abstract. Social network analysis is very useful in discovering the embedded knowledge in social network structures, which is applicable in many practical domains including homeland security, publish safety, epidemiology, public health, electronic commerce, marketing, and social science. However, social network data is usually distributed and no single organization is able to capture the global social network. For example, a law enforcement unit in Region A has the criminal social network data of her region; similarly, another law enforcement unit in Region B has another criminal social network data of Region B. Unfortunately, due the privacy concerns, these law enforcement units may not be allowed to share the data, and therefore, neither of them can benefit by analyzing the integrated social network that combines the data from the social networks in Region A and Region B. In this chapter, we discuss aspects of sharing the insensitive and generalized information of social networks to support social network analysis while preserving the privacy at the same time. We discuss the generalization approach to construct a generalized social network in which only insensitive and generalized information is shared. We will also discuss the integration of the generalized information and how it can satisfy a prescribed level of privacy leakage tolerance which is measured independently to the privacy-preserving techniques.

1 Introduction

Social networks have drawn substantial attention in the recent years due to the advance of Web 2.0 technologies. Aggregating social network data becomes easier

Chris Yang
Drexel University, Philadelphia, PA
e-mail: chris.yang@drexel.edu

Bhavani Thuraisingham
The University of Texas at Dallas, Richardson, TX
e-mail: bhavani.thuraisingham@utdallas.edu

W.W. Chu (ed.), *Data Mining and Knowledge Discovery for Big Data*,
Studies in Big Data 1,
DOI: 10.1007/978-3-642-40837-3_8, © Springer-Verlag Berlin Heidelberg 2014

through crawling the user interactions in Internet [77]. Social network analysis discovers knowledge hidden in the structure of social networks which is useful in many domains such as marketing, epidemiology, homeland security, sociology, psychology, and management. Social network data is usually owned by an individual organization or government agency. However, each organization or agency usually has a partial social network from the data aggregated in their own source. Knowledge cannot be extracted accurately if only partial information is available. Sharing of social networks between organizations enables knowledge discovery from an integrated social network obtained from multiple sources. However, the information sharing between organizations is usually prohibited due to the concern of privacy preservation; especially a social network often contains sensitive information of individuals. Early research on privacy preservation focuses on relational data and some recent researches extend it to social network data. Techniques such as *k-degree anonymity* and *k-anonymity* achieved by edge or node perturbation are proposed. However, the anonymized social network is designed for studying the global network properties. It is not applicable for integration of social networks or other social network analysis and mining tasks such as identifying the leading person or gateway. A recent study has also shown that a substantial distortion to the network structure can be caused by perturbation. Such distortion may cause errors in social network analysis and mining. In this chapter we discuss aspects of sharing the insensitive and generalized information to support social network analysis and mining while preserving the privacy at the same time.

We will motivate the problem with the following scenario. Consider two local law enforcement units A and B which have their own criminal social networks, G_A and G_B. Each of these criminal social networks is a partial network of the regional criminal social networks covering the areas policed by A and B. The criminal intelligence officer of A may not be able to identify the close distance between suspects i and j by analyzing G_A because i and j are connected through k in G_B but k is not available in G_A. Similarly, the criminal intelligence officers of B may not be able to determine the significance of suspect k by conducting centrality analysis on G_B because k makes little influence on the actors in G_B but substantial influence on the actors in G_A. By integrating G_A and G_B, the criminal intelligence officers of A and B are able to discover the knowledge that otherwise they cannot.

In this chapter, we will discuss our generalization approach for integrating social networks with privacy preservation. In Section 2 we will first provide some information on the application of social network for terrorism analysis and the need for privacy. Limitation of current approaches will be discussed in Section 3. Our approach is discussed in Section 4. Directions are discussed in Section 5.

2 Social Network Analysis

A *social network* is a network of actors with the edges corresponding to their ties. A social network is represented as a graph, $G = (V, E)$, in which V is a set of nodes corresponding to actors and E is a set of edges ($E \subseteq V \times V$) corresponding to the ties of the respective actors.

Many *social network analysis techniques* have been investigated in the literature. *Centrality measures* and *similarity measures* are two popular measurements. In general, centrality measures determine the relative significance of a node in a social network. In centrality measures, degree centrality, closeness centrality, and betweenness centrality are the typical measures. Degree centrality of a node (u) measures the ratio of the degree of u and the number of nodes other than u in the network. Closeness centrality of u measures the inverse of the total distance between u and all other nodes in the network. Betweenness centrality of u measures the number of shortest paths between any two nodes in the network that passes through u out of all the shortest paths between any two nodes. In general, centrality measures determine the relative significance of a node in a social network. Similarity measures compute the similarity between two subgroups within a social network. L1-Norm, L2-Norm, mutual information, and clustering coefficient are some common measures in similarity measures.

Recent development in *link mining* [22] of social networks focuses on *object ranking* [6,32,54], *object classification* [9,25,36,43,52], *group detection* [1,33,34,51,65,67], *entity resolution* [6,7,15], *link prediction* [13,37,38,40,53], *subgraph discovery* [31,35,75], *graph classification* [20,21], and *graph generative models* [20,68]. Object ranking [6,32,54] utilizes the link structure of a network to prioritize the objects which are represented as nodes in a network. The PageRank and HITS are the most prominent algorithms of link-based object ranking applied in Web information retrieval. Object classification [9,25,36,43,52] aims at labeling nodes of a social network from a set of categorical values by exploiting the correlation of related objects. Group detection [1,33,34,51,65,67] clusters nodes of a social network into distinct groups that share common characteristics using techniques such as graph partitioning [51], agglomerative clustering, edge betweenness [65], and stochastic modeling [67]. Entity resolution [6,7,15] determines the identity of a node that it is referring to in the real world. Such techniques are widely used in co-reference resolution, object consolidation, and deduplication. Link prediction [13,37,38,40,53] tries to predict the existence of a link between two nodes in a social network based on the attributes of the nodes and other related links. For example, it has been utilized to predict the potential interaction between actors in a social event such as blogs and forums. Subgraph discovery [31,35,75] recognizes interesting subgraphs with specific patterns in a set of social networks. Graph classification [20,21] determines if a social network is a positive or negative example of a specific class of network. Graph generative modeling [20,68] develops different random graph distributions to study the structural properties of social networks such as World Wide Web, communication networks, citation networks, and biological networks.

Some *recent applications* in *epidemiology, expert identification, criminal/ terrorist social network, academic social network* and *social network visualization* are found in the literature. For example, several models of social networks have been applied in epidemiology [46]. Population in an epidemiology social network can be divided into four groups: susceptible (S), exposed (E), infected (I), and recovered (R). SIR and SEIR models try to map bond percolation onto a social network. SIS model is used to model diseases where a long last immunity is not

present. Expert identification [3] develops mechanisms to identify experts in social and route queries to the identified experts. Criminal/Terrorist social network [74,76,77,79] aims at identifying the roles of terrorists and criminals by mining the patterns in a social network. Academic social network [62] models provide topical aspects of publications, authors and publication venues, and also a search service of experts and their associations. Co-authorship and co-citations networks are the typical networks in this study [11,26]. Social network visualization [81] provides network visualization techniques to analyze the dynamic interactions of individuals in a network.

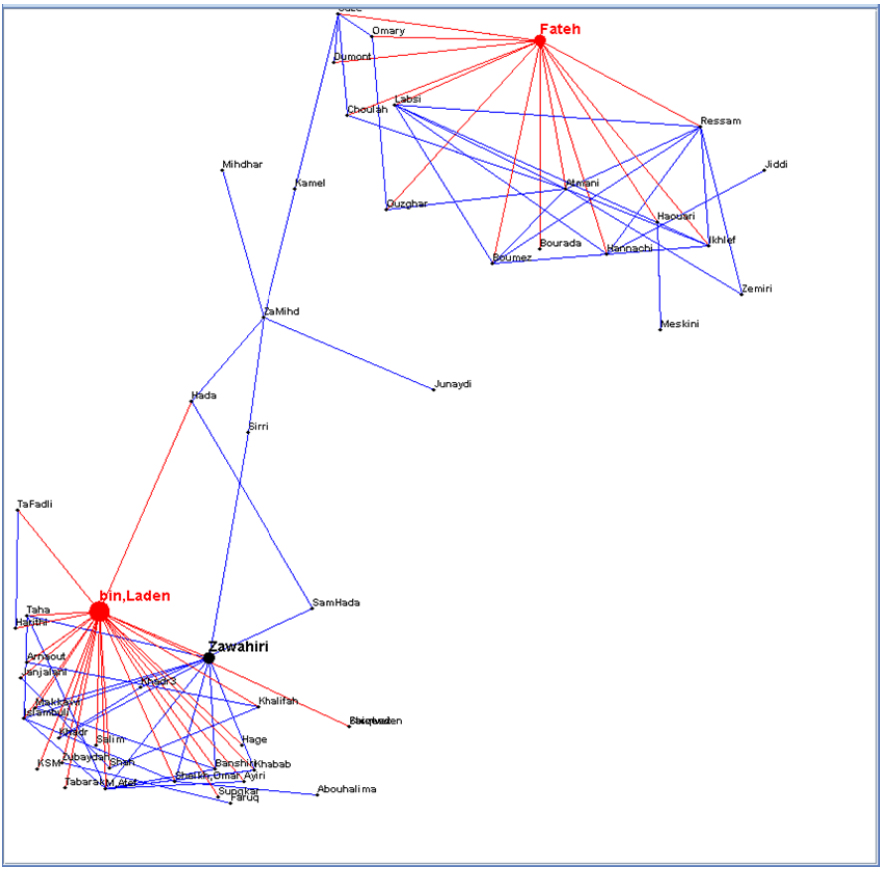


Fig. 1 Illustration of social network analysis and visualization [81]

In our previous work, we have demonstrated how to utilize social network analysis to identify the leaders, terrorist sub-groups, or gatekeeper between two or more sub-groups [76,77,78,80,81]. Such analysis is beneficial to law enforcement and crime intelligence agencies in their investigations. Visualization tools were also developed to explore the details in social networks and support the

investigators in determining the path between any two actors of interest. For example, Figure 1 illustrates the two sub-groups led by Bin Laden and Fateh and the connecting paths and gateways between the two leaders. Without this social network analysis and these visualization tools, the huge volume of aggregated data may not be meaningful to the agencies for their tasks.

3 Limitations of Current Approaches for Privacy-Preserving Social Networks

As discussed in Section 2, social network analysis is powerful in discovering embedded knowledge in the social network structure. However, extracting knowledge from a partial social network will indeed misinform us about the importance of actors or relationships between actors due to the missing information in the incomplete social network. Several attempts have been made to publish social network data for analysis. However, the *k*-anonymity approach to be discussed below does not allow social network integration so that social network analysis can be conducted on a global social network incorporating multiple social networks from different sources and yet preserve the privacy.

In our approach, we do not intend to publish individual social networks with privacy preservation. Instead, our objectives are generalizing the individual social networks so that sensitive information is preserved and multiple generalized social networks are integrated as a global social network. In this case, social network analysis can be conducted on the integrated social network which combines the generalized information of multiple social networks. The result of analyzing the integrated social network will be more accurate and knowledge on the global network can be discovered.

Privacy Preservation of Relational Data

There is a desire to publish an anonymized version of relational data owned by an organization to the public so that data mining and analytics can be conducted while the identity of individuals cannot be determined so that no one can recognize the sensitive attribute values of a particular record. Privacy preservation is important in data publishing. A simple approach of privacy preservation of relational data is removing attributes that uniquely identifying a person, such as names and identification numbers. It is a typical approach of de-identification. However, given the knowledge of some private information of the person, such as a harmless set of attributes including ages, gender, and zip code, a trivial linking attack can identify this person again even if the names and identification numbers are removed. This set of attributes supporting linking attack is known as quasi-identifiers. Below is an illustration of the linking attack.

Charles is a registered voter. Table 1 shows the medical records in a hospital and the voter registration records in the state. Both records include Charles. If the medical records are de-identified, we will not be able to tell that Charles has HIV, which is considered as sensitive and private information. However, the voter

registration records are not de-identified. If a hacker conducts a linking attack and cross check both the medical records and the voter registration records, he will be able to find that there is one person who have the same values in the set of quasi-identifiers, [Age = 29, Sex = M, Location = 35667], and therefore, he is able to conclude that Charles has HIV.

Table 1 Medical records and voter registration records in information sharing using attribute removing

Medical Records – Removing Names

Name	Age	Sex	Location	Disease
Peter	8	M	00330	Viral Infection
Paul	14	M	01540	Viral Infection
Andrew	18	M	18533	Viral Infection
Stephen	20	M	14666	Viral Infection
Charles	29	M	35667	HIV
Gordon	30	M	43986	Cancer
Linda	35	F	31147	Cancer
Mary	39	F	45246	Cancer
Stella	45	F	85103	Heart Disease
Angel	51	F	96214	HIV

quasi−identifiers

Voter Registration Records

Name	Age	Sex	Location
Charles	29	M	35667
Paul	14	M	01540
David	25	M	00338

…

A number of approaches for *privacy preservation of relational data* have been developed, for example, *k-anonymity* [60], *l-diversity* [44], *t-closeness* [39], *m-invariance* [72], *δ-presence* [50], and *k*-support anonymity [61].

k-anonymity [60] is the first attempt of privacy preservation of relational data by ensuring at least *k* records with respect to every set of quasi-identifier attributes are indistinguishable. If every record in a table is indistinguishable from at least *k*-1 other records with respect to every set of quasi-identifier attributes, this table satisfies the property of *k*-anonymity. Table 2 illustrates the de-identified medical records with *k*-anonymity. There are three quasi groups. In each quasi group, the age and location are generalized. For example, Age in quasi group A is generalized to [5,20], Age in quasi group B is generalized to [20,40], and Age in quasi group C is generalized to [41,60]. Similarly, Location in these quasi groups are also generalized to a range of values. In this case, if a hacker knows the age of a person in the voter registration records, he will not be able to link to a particular

person in the medical records because there are *k*-1 other persons who has the age in the same range. However, *k*-anonymity fails when there is a lack of diversity in the sensitive attributes. For example, there is a lack of diversity of the attribute values of disease in the quasi group with age = [5,20] and location = [00300, 02000]. One can see that all the values of the attribute Sex are M and all the values of the attribute Disease are Viral Infection in this quasi group. As a result, a hacker is able to link Paul (Age = 14, Location = 00332) to quasi group A and determine that Paul has viral infection.

Table 2 Medical records and voter registration records using *k*-anonymity

Medical Records – *k*-anonmity

Age	**Sex**	**Location**	**Disease**	
[5,20]	M	[00300,02000]	Viral Infection	quasi group *A*
	M		Viral Infection	
	M		Viral Infection	
	M		Viral Infection	
[20,40]	M	[20001,50000]	HIV	quasi group *B*
	M		Cancer	
	F		Cancer	
	F		Cancer	
[41,60]	F	[80000,99999]	Heart Disease	quasi group *C*
	F		HIV	

Voter Registration Records

Name	**Age**	**Sex**	**Location**
Peter	29	M	35667
Paul	14	M	00332
David	25	M	00338

…

l-diversity [44] ensures that there are at least *l* well-represented values of the attributes for every set of quasi-identifier attributes. The weakness is that one can still estimate the probability of a particular sensitive value. *m-invariance* [72] ensures that each set of quasi-identifier attributes has at least m tuples, each with a unique set of sensitive values. There is at most 1/*m* confidence in determining the sensitive values. Others enhanced techniques of *k*-anonymity and *l*-diversity with personalization, such as personalized anonymity [71] and (α,*k*)-anonymity [70], allow users to specify the degree of privacy protection or specify a threshold α on the relative frequency of the sensitive data. Versatile publishing [28] anonymizes sub-table to guarantees privacy rules.

Privacy preservation of relational data has also been applied in statistical database. *Query restriction* [29,48], *output perturbation* [8,14,16], and *data modification* [2,47,73] are three major approaches. *Query restriction* [29,48]

rejects certain queries when a leak of sensitive values is possible by combining the results of previous queries. *Output perturbation* [8,14,16] adds noise to the result of a query to produce a perturbed version. *Data modification* [2,47,73] prepares an adequately anonymized version of relational data to a query. *Cryptography approach* of privacy preservation of relational data aims at developing a protocol of data exchange between multiple private parties. It tries to minimize the information revealed by each party. For example, top-*k* search [66] reports the top-*k* tuples in the union of the data in several parties. However, the techniques on preserving the privacy of relational data cannot be directly applied on social network data. In the recent years, these techniques were extended for preserving the privacy of social network data.

Privacy Preservation of Social Network Data

The current research on privacy preservation of social network data (or graphs) focuses on the purpose of data publishing. A naïve approach is removing the identities of all nodes but only revealing the edges of a social network. In this case, the global network properties are preserved for other research applications assuming that the identities of nodes are not of interest in the research applications. However, Backstorm et al. [4] proved that it is possible to discover whether edges between specific targeted pairs of nodes exist or not by active or passive attacks. Based on the uniqueness of small random subgraphs embedded in a social network, one can infer the identities of nodes by solving a set of restricted isomorphism problems. Active attacks refer to planting well structured subgraphs in a published social network and then discovering the links between targeted nodes by identifying the planted structures. Passive attacks refer to identifying a node by its association with neighbors and then identifying other nodes that are linked to this association. Such attacks can also be considered as neighborhood attacks.

In order to tackle active and passive attacks and preserve the privacy of node identities in a social network, there are several anonymization models proposed in the recent literature: *k-candidate anonymity* [23], *k-degree anonymity* [42], *and k-anonymity* [89]. Such anonymization models are proposed to increase the difficulty of being attacked based on the notion of *k*-anonymity in relational data. *k*-candidate anonymity [23] defines that there are at least *k* candidates in a graph *G* that satisfies a given query *Q*. *k*-degree anonymity [42] defines that, for every node *v* in a graph *G*, there are at least *k*-1 other nodes in *G* that have the same degree as *v*. *k*-anonymity [89] has the strictest constraint. It defines that, for every node *v* in a graph *G*, there are at least *k*-1 other nodes in *G* such that their anonymized neighborhoods are isomorphic. Zheleva [88] proposed an edge anonymization model for social networks with labeled edges rather than labeled nodes.

The technique to achieve the above anonymities is *edge or node perturbation* [23,42,89]. By adding and/or deleting edges and/or nodes, a perturbed graph is generated to satisfy the anonymity requirement. Adversaries can only have a confident of 1/*k* to discover the identity of a node by neighborhood attacks.

Since the current research on privacy preservation of social network data focuses on preserving the node identities in data publishing, the anonymized social network can only be used to study the global network properties but may not be applicable to other social network analysis tasks. In addition, the sets of nodes and edges in a perturbed social network are different from the set of nodes and edges in the original social network. As reported by Zhou and Pei [89], the number of edges added can be as high as 6% of the original number of edges in a social network. A recent study [87] has investigated how edge and node perturbation can change certain network properties. Such distortion may cause significant errors in certain social network analysis tasks such as centrality measurement although the global properties can be maintained. In this research, we not only preserve the identities of nodes, but also the social network structures (i.e. edges).

The limitations of current social network privacy preservation techniques include: (a) It preserves the identities of nodes in a social network but it does not preserve the network structure (i.e. edges) of a social network, (b) the anonymization approach prohibits the integration of social networks, (c) the perturbation changes the connectivity of nodes and it can significantly distort the social network analysis result, (d) the existing privacy preservation techniques have not considered the application of social network analysis.

Another approach of privacy-preserving social network analysis is secure multi-party computation (SMC) [86]. In SMC, there is a set of functions that multiple parties wish to jointly compute and each party has its own private inputs [41]. By preserving the private inputs, SMC uses cryptography technology to compute the joint function [9,18,30,56]. However, there are disadvantages of the cryptography approach. The encrypted data can be attacked and recovered by the malicious party. The complexity of SMC is high which may not be computationally feasible for large scale social network data.

4 Our Approach

Instead of using edge or node perturbation or secure multi-party computation approaches, we propose to use a subgraph generalization approach to preserve the sensitive data and yet share the insensitive data. The social network owned by each party will be decomposed to multiple subgraphs. Each subgraph will be generalized as a probabilistic model depending on the sensitive and insensitive data available as well as the objective of the social network analysis and mining tasks. The probabilistic models of the generalized subgraphs from multiple sources will then be integrated for social network analysis and mining. The social network analysis and mining will be conducted on the global and generalized social network rather than the partial social network owned by each party. The knowledge that cannot be captured in individual social networks will be discovered in the integrated global social network.

By using such approach, it will overcome the limitations of the errors produced by the perturbation approach and yet allow integration of multiple social networks. It also avoids the attack of the encrypted data in SMC approach because the shared

data are insensitive. The complexity of this approach will also be reduced substantially.

Our Definition of Privacy

Given two or more social networks (G_1, G_2, …) from different organizations (O_1, O_2, …), the objective is achieving *more accurate social network analysis and mining results* by integrating *the shared crucial and insensitive information* between these social networks and at the same time *preserving the sensitive information* with *a prescribed level of privacy leakage tolerance*. Each organization O_i has a piece of social network G_i, which is part of the whole picture – a social network $\boldsymbol{G}$ constructed by integrating all G_i. Conducting the social network analysis task on $\boldsymbol{G}$, one can obtain the exact social network analysis result from the integrated information. However, conducting the social network analysis task on any G_i, one can never achieve the exact social network analysis result because of the missing information. By integrating G_i and some *generalized information* of G_j, O_i should be able to achieve more accurate social network analysis results although it is not the exact social network analysis result. That means if O_i can obtain generalized information from all other organizations, O_i will be able to obtain a social network analysis result much closer to the exact social network analysis result than that obtained from G_i alone.

The adversary attack can be active or passive attacks. Active attacks refer to planting well-structured subgraphs in a social network and then discovering the links between targeted nodes by identifying the planted structures. Passive attacks refer to identifying a node by its association with neighbors and then identifying other nodes that are linked to this association. Such attacks can also be considered as neighborhood attacks.

The generalized information in our approach is a probabilistic model of the general property of a social network. As a result, it does not release the sensitive information of a particular social network. In addition, not all information is useful for a particular social network analysis task. To determine how to generate the generalized information, one may decide what the crucial information for the designated social network analysis task is.

Integration points are crucial to integrate the probabilistic models of multiple social networks. These integration points must be insensitive information to the parties that are involved in the process. A piece of information is not sensitive when it is known to both parties; however, other information that is related to such insensitive information is still considered sensitive. For example, when a suspect is referred from a law enforcement unit to another law enforcement unit, the identity of this suspect is insensitive to both units but the identities of other acquaintances who are associated with this suspect are sensitive. A piece of information can also be known to both parties when such information is available from a common source. For example, when a suspect is reported in the national news or his identity is available in a national database, the identity of this suspect is known to all law enforcement units.

Any generalized information is still subject to privacy leakage depending on the background knowledge owned by the other parties. As a result, we need to ensure that a specified tolerance of privacy leakage is satisfied. The measure of privacy leakage must be independent to the techniques in generating and integrating generalized information of social networks. Privacy means that no party should be able to learn anything more than the insensitive information shared by other parties and the prescribed output of the social network analysis tasks. If any adversary attack can be applied to learn any private and sensitive data, there is a privacy leakage. In this problem, the shared insensitive information is the generalized information and the identity of the insensitive nodes which are the integration points. The prescribed outputs of the social network analysis tasks are the centrality measures or similarity measures such as closeness centrality of a node.

The leakage of private information includes the identities of sensitive nodes and the adjacency (i.e. edges) of any two nodes regardless if any of these nodes are sensitive or insensitive. If any of the active or passive attacks can be applied on the generalized information or the output of the social network analysis tasks to learn the abovementioned private information, there is a privacy leakage. Any privacy preservation technique should protect the exact identity of sensitive nodes or the adjacency between any two nodes. Table 3 presents the definitions of the tolerance of privacy leakage on a sensitive node, on the adjacency between an insensitive node and a sensitive node, and on the adjacency between two sensitive nodes:

Table 3 Definitions of τ-tolerance of privacy leakage

τ-tolerance of privacy leakage on an sensitive node: (1) The identity of a sensitive node cannot be identified as one of τ or fewer possible known identities.
τ-tolerance of privacy leakage on the adjacency between an insensitive node and a sensitive node: (1) The identity of an insensitive node is known but its adjacency with other sensitive nodes is not known. (2) The adjacent nodes cannot be identified as one of τ or fewer possible sensitive nodes.
$\tau_1\tau_2$-tolerance of privacy leakage on the adjacency between two sensitive nodes: (1) The identity of a sensitive node *A* cannot be identified as one of τ_1 or fewer possible known identities. (2) The adjacent node of this sensitive node *A* cannot be identified as one of τ_2 or fewer possible known identities

Most attacks cannot discover the exact identity or adjacency given a reasonable privacy preservation technique. However, many attacks are able to narrow down the identity to a few possible known identities. Ideally, a privacy-preserving technique should achieve ∞-tolerance, which means no attack can find a clue of the possible identity of a sensitive node. In reality, it is almost impossible to achieve ∞-tolerance due to the background knowledge possessed by the adversaries. However, a good privacy-preserving technique should reduce privacy leakage as much as possible, which means achieving a higher value of τ in privacy leakage.

The generalized information in this problem is the probabilistic models of the generalized social networks instead of a perturbed model using the *k*-anonymity approach. As a result, the τ-tolerance of privacy leakage is independent to the generalization technique. In addition, it preserves both the identities and network structures. By integrating the probabilistic models of multiple generalized social networks, the objective is achieving a better performance of social network analysis tasks. At the same time, neither the probabilistic models nor the social network analysis results should release private information that may violate the prescribed τ-tolerance of privacy leakage when it is under adversary attacks.

A Framework of Information Sharing and Privacy Preservation for Integrating Social Networks

In this chapter, we use the framework of information sharing and privacy preservation for integrating social networks as shown in Figure 2.

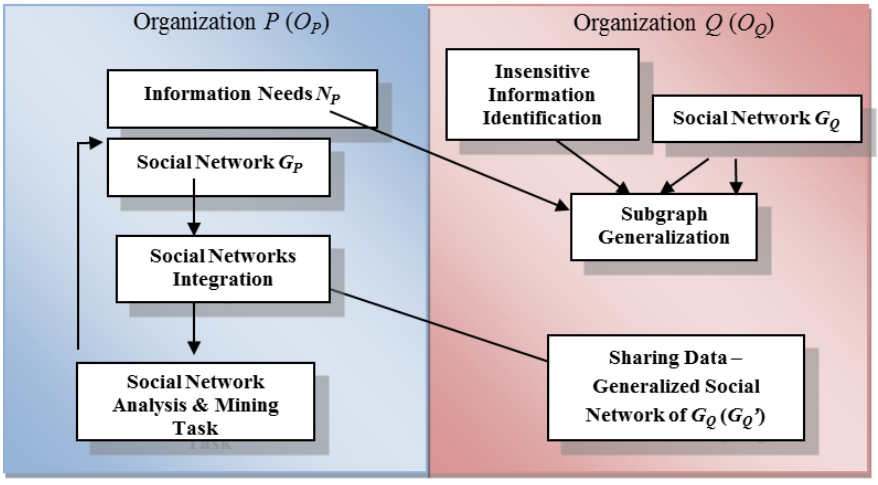


Fig. 2 A proposed framework of information sharing and privacy preservation for integrating social networks

Assuming organization P (O_P) and organization Q (O_Q) have social networks G_P and G_Q respectively, O_P needs to conduct a social network analysis task but G_P is only a partial social network for the social network analysis task. If there is not

any privacy concern, one can integrate G_P and G_Q to generate an integrated $\boldsymbol{G}$ and obtain a better social network analysis result. Due to privacy concern, O_Q cannot release G_Q to O_P but only shares the generalized information of G_Q to O_P. At the same time, O_P does not need all data from O_Q but only those that are critical for the social network analysis task. The objectives are maximizing the information sharing that is useful for the social network analysis task but preserving the sensitive information to satisfy the pre-scribed τ-tolerance of privacy leakage and achieve more accurate social network analysis results.

Sharing Insensitive Information

Thuraisingham [63,64] discussed a coalition of dynamic data sharing, in which security and integrity policies are enforced. As reported in the Washington Post in September 2008 [49], there was no systematic mechanism for sharing intelligence between private companies or between companies and the government. It also emphasized that the government should take actions on developing a mechanism to share unclassified information while some information should remain classified. Without information sharing, the US developed products and technology which could be easily stolen with little effort. The key point is differentiating the sensitive and insensitive information to permit necessary information sharing while protecting the privacy. A well-developed privacy policy provides a mechanism to determine what information should be shared based on the information needs and the trust degree of the information requesting party.

In our framework as presented in Figure 1, we propose that the information shared between two parties should be based upon the *information needs* to satisfy the social network analysis task, the *identification of insensitive information* between the two parties, and the *information available* in the social network. When we perform social network data sharing, we need to consider what kinds of information has the highest utility to accomplish a particular social network analysis task. We need to determine the insensitive data to be shared and serve as the integration points between two social networks so that the generalized information can be integrated.

In our research problem, both identities and network structure are considered sensitive but only generalized information is shared. However, we also consider a small number of identities are insensitive. The identities of these nodes are known to the public or insensitive to both organizations who are sharing the information. we define the sensitivity of an identity u between two organizations O_p and O_q as *sensitivity* (u, O_p, O_q):

$$sensitivity(u, O_p, O_q) = \begin{cases} 0 & if\ Refer_{O_p}(u, O_q) = 1, Refer_{O_q}(u, O_p) = 1, or\ Source_{O_p O_q}(u) = 1 \\ 1 & else \end{cases}$$

where $Refer_x(u, y) = 1$ when x make a referral of u to y and $Source_{x,y}(u) = 1$ when u can be obtained from a common source of x and y

In this work, we focus on the fundamental centrality analysis. To compute different centrality measures, attributes such as the degree of nodes and the

shortest distance between nodes [19,24] have high utilities. When information is shared with another organization, some sensitive information must be preserved but the generalized information can be released so that more accurate estimation of the required information for centrality measures can be obtained. Attributes for consideration in generalization includes number of nodes in a sub-group, the diameter of a sub-group, the distributions of node degrees, and the eccentricity of the insensitive nodes.

Generalization

A subgraph generalization generates a generalized version of a social network, in which a connected subgraph is transformed as a *generalized node* and only *generalized information* will be presented in the generalized node. The generalized information is the probabilistic model of the attributes. A subgraph of $G = (V, E)$ is denoted as $G' = (V', E')$ where $V' \subset V$, $E' \subset E$, $E' \subseteq V' \times V'$. G' is a connected subgraph if there is a path for each pair of nodes in G'. We only consider a connected subgraph when we conduct subgraph generalization. The edge that links from other nodes in the network to any nodes of the subgraph will be connected to the generalized node. The generalized social network protects all sensitive information while releasing the crucial and non-sensitive information to the information requesting party for social network integration and the intended social network analysis task. A mechanism is needed to (i) *identify the subgraphs* for generalization, (ii) *determine the connectivity* between the set of generalized nodes in the generalized social network, and (iii) *construct the generalized information* to be shared.

The constructed subgraphs must be mutually exclusive and exhaustive. A node v can only be part of a subgraph but not any other subgraphs. The union of nodes from all subgraphs V_1', V_2', …, V_n' should be equal to V, the original set of nodes in G. To construct a subgraph for generalization, there are a few alternatives including *n-neighborhood*, and *k-connectivity.*

n-neighborhood

For a node $v \in G$, the i^{th} neighbor of v is $N_i(v) = \{u \in G : d(u,v) = i\}$ where $d(u,v)$ is the distance between u and v. Given a target node, v which can be an insensitive node, the n-neighborhood graph of v is denoted as $n\text{-}Neighbor(v,G)$. $n\text{-}Neighbor(v,G) = (V^i, E^i)$ such that $V^i = \{ u \in G : d(u,v) \leq n\}$ and $E^i \subset E$, $E^i \subseteq V^i \times V^i$.

k-connectivity

The connectivity $\kappa(G)$ of a graph G is the minimum number of nodes whose removal results in a disconnected graph. The edge connectivity $\kappa'(G)$ of a graph G is the minimum number of edges whose removal results in a disconnected graph. A graph is k-connected if $\kappa(G) \geq k$ and it is k-edge connected if $\kappa'(G) \geq k$. If a graph is k-edge connected, two or more connected subgraphs (components) that are disconnected from each other are created after removing the k edges.

Subgraphs can further be generated if the subgraphs being created are also *k*-edge connected.

We illustrate the sub-graph generalization using the *K*-nearest neighbor (*KNN*) method. Let $SP^D(v, v_i^C)$ be the distance of the shortest path between v and v_i^C. When v is assigned to the subgraph G_i in subgraph generation, $SP^D(v, v_i^C)$ must be shorter than or equal to $SP^D(v, v_j^C)$ where $j = 1, 2, .., K$ and $j \neq i$. Secondly, an edge exists between two generalized nodes G_i and G_j in the generalized graph G' if and only if there is an edge between any two nodes in G such that one from each generalized node, G_i and G_j.

The *KNN* subgraph generation algorithm is presented below:

```
length=1;
Step 1
V= V - {v_1^C, v_2^C , … v_K^C };
Step 2
While V ≠ Ø
        Step 3
     For each v_j ∈  V
Step 4
     For each i = 1 to K
          Step 5
          IF(SP^D(v_j,  v_i^C ) == length);
Step 6
                    V_i = V_i + v_j;
Step 7
                    V= V – v_j;
Step 8
          End For;
               Step 9
       End For;
               Step 10
length++;
Step 11
End While
            Step 12
For each (v_i,v_j) ∈ E
Step 13
    IF( Subgraph(v_i) == Subgraph(v_j) )
Step 14
    // Subgraph(v_i) is the subgraph such that v_i ∈ Subgraph(v_i)
      G_k = Subgraph(v_i)
              Step 15
   E_k = E_k + (v_i,v_j)
          Step 16
```

ELSE
Step 17
Create an edge between *Subgraph*(v_i) and *Subgraph*(v_j) and add it to E'
Step 18
End For
Step 19

Figure 3 illustrates the subgraph generation by *KNN* method. *G* has seven nodes including v_1 and v_2. If we take v_1 and v_2 as the insensitive nodes and we are going to generate two subgraphs by *1NN* method, all other nodes will be assigned to one of the two subgraphs depending on their shortest distances with v_1 and v_2. Two subgraphs G_1 and G_2 are generated as illustrated in Figure 3.

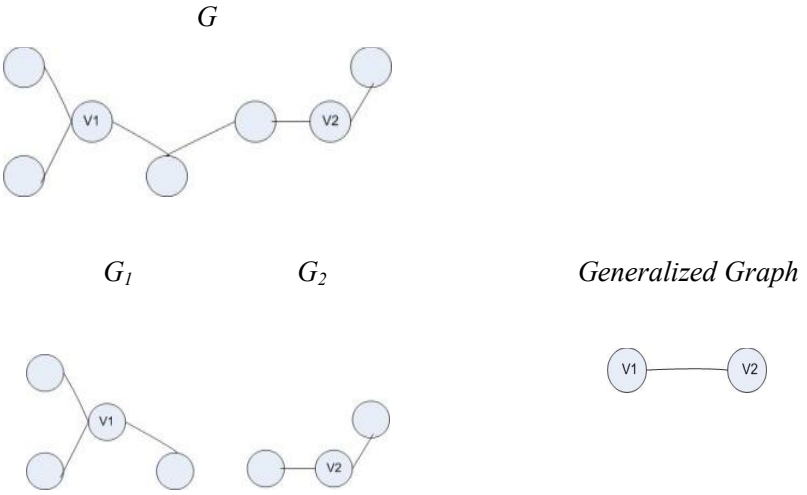


Fig. 3 Illustrations of generating subgraphs

The *KNN* subgraph generation algorithm creates *K* subgraphs G_1, G_2, …, G_K from *G*. Each subgraph, G_i, has a set of nodes, V_i, and a set of edges, E_i. Edges between subgraphs, *E'*, are also created. A generalized graph, *G'*, is constructed where each generalized node corresponds a subgraph G_i and labeled by the insensitive node, v_i^C.

Probabilistic Model of Generalized Information

For each generalized node $v_j' \in V_i'$, we determine the generalized information to be shared. The generalized information should achieve the following objectives: (i) is useful for the social network analysis task after integration, (ii) preserves the sensitive information, and (iii) is minimal so that unnecessary information is not released. The generalized information of V_i' can be the probabilistic model of the

distance between any two nodes v_j and v_k in V_i', $P(Distance(v_j, v_j) = d)$, v_j, $v_k \in V_i$' [82]. The construction of subgraphs plays an important role in determining the generalized information to be shared and the usefulness of the generalized information.

In addition to the utility of the generalized information, the development of the subgraph construction algorithms must take the privacy leakage into consideration. By taking the generalized subgraphs and the generalized information of each subgraphs, attacks can be designed to discover identities and adjacencies of sensitive and insensitive nodes.

Integrating Generalized Social Network for Social Network Analysis Task

Figure 4 presents the overview of the subgraph generalization approach. Taking the generalized social network from the multiple organizations, we need to *develop techniques to make use of the shared data with the existing social network to accomplish the intended social network analysis task* as illustrated in Figure 3. For example, if the social network analysis task is computing the closeness centrality, we need to develop the technique to make use of the additional information from the generalized nodes to obtain accurate estimations of the distances between nodes in a social network [19,24].

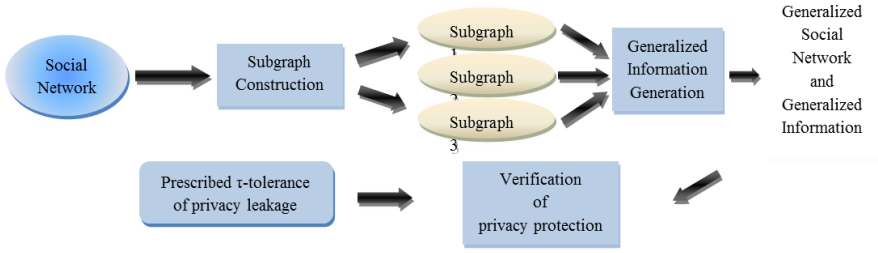


Fig. 4 Framework of subgraph generalization approach

The result of a social network analysis task is denoted as $\Im(G)$ where G is a social network. The social network analysis result of Organization P is $\Im(G_P)$. If Organization Q shares its social network, G_Q, with Organization P, P can integrate G_P and G_Q to $\boldsymbol{G}$ and obtain a social network analysis result $\Im(\boldsymbol{G})$. The accuracy of $\Im(\boldsymbol{G})$ is much higher than that of $\Im(G_P)$; however, Q cannot share G_Q with P due the privacy concern but only the generalized social network, G_Q'. The integration technique here should be capable to utilize the useful information in G_Q' and integrate with G_P, which is denoted as $I(G_P, G_Q')$. The accuracy of $\Im(I(G_P, G_Q'))$ should be close to $\Im(\boldsymbol{G})$ and significantly better than $\Im(G_P)$.

We have conducted an experiment using the Global Salafi Jihad terrorist social network [81]. There are 366 nodes and 1275 ties in this network with four major terrorist groups including Central Staff of al Qaeda, Core Arab, Southeast Asia, and Maghreb Arab. We have applied the *KNN* sub-graph generalization technique

and utilized the closeness centrality as the social network analysis task. It is found that using the generalization approach can improve the performance of closeness centrality by over 35%. This result shows that the proposed approach of integrating social networks is promising.

5 Conclusion

Social network analysis is important for extracting hidden knowledge in a community. It is particularly important for investigating the terrorist and criminal communication patterns and the structure of their organization. Unfortunately, most law enforcement and intelligence units only own a small piece of the social network. Due to privacy concerns, these pieces of data cannot be shared among the units. Therefore, the utility of each piece of information is limited. In this chapter, we introduce a generalization approach for preserving privacy and integrating multiple social networks. The integrated social network will provide better information for us to conduct social network analysis such as computing the centrality. In this chapter, we have also discussed the τ-tolerance, which specifies the level of privacy leakage that must be protected. Our experimental result also shows that the generalization approach and social network integration produce promising performance.

References

1. Adibi, Chalupsky, H., Melz, E., Valente, A.: The KOJAK Group Finder: Connecting the Dots via Intergrated Knowledge-based and Statistical Reasoning. In: Innovative Applications of Artificial Intelligence Conference (2004)
2. Agrawal, R., Srikant, R., Thomas, D.: Privacy Preserving OLAP. In: ACM SIGMOD 2005 (2005)
3. Ahmad, M.A., Srivastava, J.: An Ant Colony Optimization Approach to Expert Identification in Social Networks. In: Liu, H., Salerno, J.J., Young, M.J. (eds.) Social Computing, Behavioral Modeling, and Prediction. Springer (2008)
4. Backstrom, L., Dwork, C., Kleinberg, J.: Wherefore Art Thou R3579X? Anonymized Social Networks, Hidden Patterns, and Structural Steganography. In: WWW 2007, Banff, Alberta, Canada (2007)
5. Bhatt, R., Chaoji, V., Parekh, R.: Predicting Product Adoption in Large-Scale Social Networks. In: ACM CIKM, Toronto, Ontario (2010)
6. Bhattacharya, I., Getoor, L.: Iterative Record Linkage for Cleaning and Integration. In: SIGMOD 2004 Workshop on Research Issues on Data Mining and Knowledge Discovery (2004)
7. Bhattacharya, I., Getoor, L.: Entity Resolution in Graphs. Technical Report 4758, Computer Science Department, University of Maryland (2005)
8. Blum, A., Dwork, C., McSherry, F., Nissim, K.: Practical Privacy: the Sulq Framework. In: ACM PODS 2005 (2005)
9. Brickell, J., Shmatikov, V.: Privacy-Preserving Graph Algorithms in the Semi-honest Model. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 236–252. Springer, Heidelberg (2005)

10. Chakrabarti, S., Dom, B., Indyk, P.: Enhanced Hypertext Categorization using Hyperlinks. In: ACM SIGMOD 1998 (1998)
11. Chau, A.Y.K., Yang, C.C.: The Shift towards Multi-Disciplinarily in Information Science. Journal of the American Society for Information Science and Technology (2008)
12. Chen, H., Yang, C.C.: Intelligence and Security Informatics: Techniques and Applications. Springer (2008)
13. Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., Slattery, S.: Learning to Construct Knowledge Bases from the World Wide Web. Artificial Intelligence 118, 69–114 (2000)
14. Dinur, I., Nissim, K.: Revealing Information While Preserving Privacy. In: ACM PODS 2003 (2003)
15. Dong, X., Halevy, A., Madhavan, J.: Reference Reconciliation in Complex Information Spaces. In: ACM SIGMOD International Conference on Management of Data (2005)
16. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating Noise to Sensitivity in Private Data Analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006)
17. Frantz, T., Carley, K.M.: A Formal Characterization of Cellular Networks. Technical Report CMU-ISRI-05-109, Carnegie Mellon University (2005)
18. Frikken, K.B., Golle, P.: Private Social Network Analysis: How to Assemble Pieces of a Graphy Privately. In: The 5th ACM Workshop on Privacy in Electronic Society (WPES 2006), Alexandria, VA (2006)
19. Gao, J., Qiu, H., Jiang, X., Wang, T., Yang, D.: Fast Top-K Simple Shortest Discovery in Graphs. In: ACM CIKM, Toronto, Ontario (2010)
20. Gartner, T.: Exponential and Geometric Kernels for Graphs. In: NIPS Workshop on Unreal Data: Principles of Modeling Nonvectorial Data (2002)
21. Gartner, T.: A Survey of Kernels for Structured Data. ACM SIGKDD Explorations 5, 49–58 (2003)
22. Getoor, L., Diehl, C.P.: Link Mining: A Survey. ACM SIGKDD Explorations 7, 3–12 (2005)
23. Hay, M., Miklau, G., Jensen, D., Weis, P., Srivastava, S.: Anonymizing Social Networks. Technical Report 07-19, University of Massachusetts, Amherst (2007)
24. Gubichev, A., Bedathur, S., Seufert, S., Weikum, G.: Fast and Accurate Estimation of Shortest Paths in Large Graphs. In: ACM CIKM, Toronto, Ontario (2010)
25. Himmel, R., Zucker, S.: On the Foundations of Relaxation Labeling Process. IEEE Transactions on Pattern Analysis and Machine Intelligence, 267–287 (1983)
26. Huang, J., Sun, H., Han, J., Deng, H., Sun, Y., Liu, Y.: SHRINK: A Structural Clustering Algorithm for Detecting Hierarchical Communities in Networks. In: ACM CIKM, Toronto, Ontario (2010)
27. Huang, J., Zhuang, Z., Li, J., Giles, C.L.: Collaboration Over Time: Characterizing and Modeling Network Evolution. In: ACM WSDM 2008 Palo Alto, CA (2008)
28. Jin, X., Zhang, M., Zhang, N., Das, G.: Versatile Publishing for Privacy Preservation. In: ACM KDD, Washington, DC (2010)
29. Kenthapadi, K., Mishra, N., Nissim, K.: Simulatable Auditing. In: PODS 2005 (2005)
30. Kerschbaum, F., Schaad, A.: Privacy-Preserving Social Network Analysis for Criminal Investigations. In: Proceedings of the ACM Workshop on Privacy in Electronic Society, Alexandria, VA (2008)

31. Ketkar, N., Holder, L., Cook, D.: Comparison of Graph-based and Logic-based Multi-relational Data Mining. In: ACM SIGKDD Explorations, vol. 7 (December 2005)
32. Kleinberg, J.: Authoritative Sources in a Hyperlinked Environment. Journal of the ACM 46, 604–632 (1999)
33. Kubica, J., Moore, A., Schneider, J., Yang, Y.: Stochastic Link and Group Detection. In: National Conference on Artificial Intelligence: American Association for Artificial Intelligence (2002)
34. Kubica, J., Moore, A., Schneider, J.: Tractable Group Detection on Large Link Data Sets. In: IEEE International Conference on Data Mining (2003)
35. Kuramochi, M., Karypis, G.: Frequent Subgraph Discover. In: IEEE International Conference on Data Mining (2001)
36. Lafferty, L., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: International Conference on Machine Learning (2001)
37. Leroy, V., Cambazoglu, B.B., Bonchi, F.: Cold Start Link Prediction. In: ACM SIGKDD, Washington, DC (2010)
38. Leung, C.W., Lim, E., Lo, D., Weng, J.: Mining Ineresting Link Formation Rules in Social Networks. In: ACM CIKM, Toronto, Ontario (2010)
39. Li, N., Li, T.: t-closeness: Privacy Beyond k-anonymity and ldiversity. In: ICDE 2007 (2007)
40. Liben-Nowell, D., Kleinberg, J.: The Link Prediction Problem for Social Networks. In: International Conference on Information and Knowledge Management, CIKM 2003 (2003)
41. Lindell, Y., Pinkas, B.: Secure Multiparty Computation for Privacy-Preserving Data Mining. The Journal of Privacy and Confidentiality 1(1), 59–98 (2009)
42. Liu, K., Terzi, E.: Towards Identity Anonymization on Graphs. In: ACM SIGMOD 2008. ACM Press, Vancouver (2008)
43. Lu, Q., Getoor, L.: Link-based Classification. In: International Conference on Machine Learning (2003)
44. Machanavajjhala, A., Gehrke, J., Kifer, D.: L-diversity: Privacy beyond k-anonymity. In: ICDE 2006 (2006)
45. Merugu, S., Ghosh, J.: A Distributed Learning Framework for Heterogeneous Data Sources. In: ACM KDD 2005, Chicago, Illinois, USA (2005)
46. Morris, M.: Network Epidemiology: A Handmbook for Survey Design and Data Collection. Oxford University Press, London (2004)
47. Muralidhar, K., Sarathy, R.: Security of Random Data Perturbation Methods. ACM Transactions on Database Systems 24, 487–493 (1999)
48. Nabar, S.U., Marthi, B., Kenthapadi, K., Mishra, N., Motwani, R.: Towards Robustness in Query Auditing. In: VLDB, pp. 151-162 (2006)
49. Nakashima, E.: "Cyber Attack Data-Sharing is Lacking, Congress Told," the Washington Post, p. D02 (September 19, 2008), http://www.washingtonpost.com/wp-dyn/content/article/2008/09/18/AR2008091803730.html
50. Nergiz, M.E., Atzori, M., Clifton, C.: Hiding the Presence of Individuals from Shared Database. In: SIGMOD 2007 (2007)
51. Newman, M.E.J.: Detecting Community Structure in Networks. European Physical Journal B 38, 321–330 (2004)

52. Oh, H.J., Myeaeng, S.H., Lee, M.H.: A Practical Hypertext Categorization Method using Links and Incrementally Available Class Information. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (2000)
53. O'Madadhain, J., Hutchins, J., Smyth, P.: Prediction and Ranking Algorithms for Even-based Network Data. ACM SIGKDD Explorations 7 (December 2005)
54. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. Technical Report, Standford University (1998)
55. Sageman, M.: Understanding Terror Networks. University of Pennsylvania Press (2004)
56. Sakuma, J., Kobayashi, S.: Link Analysis for Private Weighted Graphs. In: Proceedings of ACM SIGIR 2009, Boston, MA, pp. 235–242 (2009)
57. Samarati, P.: Protecting Respondents' Identities in Microdata Release. IEEE Transactions on Knowledge and Data Engineering 13, 1010–1027 (2001)
58. Srivastava, J., Pathak, N., Mane, S., Ahmad, M.A.: Data Mining for Social Network Analysis. Tutorial Notes in the 2006 IEEE International Conference on Data Mining, Hong Kong, December 18-22 (2006)
59. Sweeney, L.: Uniqueness of Simple Demographics in the US Population. Technical Report, Carnegie Mellon University (2000)
60. Sweeney, L.: K-Anonymity: A Model for Protecting Privacy. International Journal of Uncertainty Fuzziness Knowledge-based Systems 10, 557–570 (2002)
61. Tai, C., Yu, P.S., Chen, M.: k-Support Anonymity Based on Pseudo Taxonomy for Outsourcing of Frequent Itemset Mining. In: ACM SIGKDD, Washington, DC (2010)
62. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: ArnetMiner: Extraction and Mining of Academic Social Networks. In: ACM KDD 2008. ACM Press, Las Vegas (2008)
63. Thuraisingham, B.: Security Issues for Federated Databases Systems. In: Computers and Security. North Holland (1994)
64. Thuraisingham, B.: Assured Information Sharing: Technologies, Challenges and Directions. In: Chen, H., Yang, C.C. (eds.) Intelligence and Security Informatics: Technqiues and Applications. SCI, vol. 135, pp. 1–15. Springer, Heidelberg (2008)
65. Tyler, J.R., Wilkinson, D.M., Huberman, B.A.: Email as Spectroscopy: Automated Discovery of Community Structure within Organizations, The Netherlands (2003)
66. Vaidya, R.J., Clifton, C.: Privacy-preserving top-k queries. In: International Conference of Data Engineering (2005)
67. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge University Press, Cambridge (1994)
68. Watts, D.J., Strogatz, S.H.: Collective Dynamics of "Small-wolrd" Networks. Nature 339, 440–442 (1998)
69. Wolfe, A.P., Jensen, D.: Playing Multiple Roles: Discovering Overlapping Roles in Social Networks. In: ICML 2004 Workshop on Statistical Relational Learning and its Connections to Other Fields (2004)
70. Wong, R.C., Li, J., Fu, A., Wang, K.: (a,k)-Anonymity: An enhanced k-Anonymity Model for Privacy-Preserving Data Publishing. In: SIGKDD, Philadelphia, PA (2006)
71. Xiao, X., Tao, Y.: Personalized Privacy Preservation. In: SIGMOD, Chicago, Illinois (2006)
72. Xiao, X., Tao, Y.: m-invariance: Towards Privacy Preserving Republication of Dynamic Datasets. In: ACM SIGMOD 2007. ACM Press (2007)
73. Xiao, X., Tao, Y.: Dynamic Anonymization: Accurate Statistical Analysis with Privacy Preservation. In: ACM SIGMOD 2008. ACM Press, Vancouver (2008)

74. Xu, J., Chen, H.: CrimeNet Explorer: A Framework for Criminal Network Knowledge Discovery. ACM Transactions on Information Systems 23, 201–226 (2005)
75. Yan, X., Han, J.: gSpan: Graph-based Substructure Pattern Mining. In: International Conference on Data Mining (2002)
76. Yang, C.C., Liu, N., Sageman, M.: Analyzing the Terrorist Social Networks with Visualization Tools. In: IEEE International Conference on Intelligence and Security Informatics, San Diego, CA (2006)
77. Yang, C.C., Ng, T.D.: Terrorism and Crime Related Weblog Social Network: Link, Content Analysis and Information Visualization. In: IEEE International Conference on Intelligence and Security Informatics, New Brunswick, NJ (2007)
78. Yang, C.C., Ng, T.D., Wang, J.-H., Wei, C.-P., Chen, H.: Analyzing and Visualizing Gray Web Forum Structure. In: Yang, C.C., et al. (eds.) PAISI 2007. LNCS, vol. 4430, pp. 21–33. Springer, Heidelberg (2007)
79. Yang, C.C.: Information Sharing and Privacy Protection of Terrorist or Criminal Social Networks. In: IEEE International Conference on Intelligence and Security Informatics, Taipei, Taiwan, pp. 40–45 (2008)
80. Yang, C.C., Ng, T.D.: Analyzing Content Development and Visualizing Social Interactions in Web Forum. In: IEEE International Conference on Intelligence and Security Informatics Taipei, Taiwan (2008)
81. Yang, C.C., Sageman, M.: Analysis of Terrorist Social Networks with Fractal Views. Journal of Information Science (2009)
82. Yang, C.C., Tang, X.: Social Networks Integration and Privacy Preservation using Subgraph Generalization. In: Proceedings of AMC SIGKDD Workshop on CyberSecurity and Intelligence Informatics, Paris, France (June 28, 2009)
83. Yang, C.C., Tang, X., Thuraisingham, B.: An Analysis of User Influence Ranking Algorithms on Dark Web Forums. In: Proceedings of ACM SIGKDD Workshop on Intelligence and Security Informatics (ISI-KDD), Washington, D.C. (July 25, 2010)
84. Yang, C.C., Thuraisingham, B.: Privacy-Preserved Social Network Integration and Analysis for Security Informatics. IEEE Intelligent Systems 25(3), 88–90 (2010)
85. Yang, X., Asur, S., Parthasarathy, S., Mehta, S.: A Visual-Analytic Toolkit for Dynamic Interaction Graphs. In: ACM KDD 2008, Las Vegas, Nevada (2008)
86. Yao, A.: Protocols for Secure Computations. In: Proceedings of the Annual IEEE Symposium on Foundations of Computer Science, vol. 23 (1982)
87. Ying, X., Wu, X.: Randomizing Social Networks: A Spectrum Preserving Approach. In: SIAM International Conference on Data Mining (SDM 2008), Atlanta, GA (2008)
88. Zheleva, E., Getoor, L.: Preserving the Privacy of Sensitive Relationships in Graph Data. In: Bonchi, F., Malin, B., Saygın, Y. (eds.) PInKDD 2007. LNCS, vol. 4890, pp. 153–171. Springer, Heidelberg (2008)
89. Zhou, B., Pei, J.: Preserving Privacy in Social Networks against Neighborhood Attacks. In: IEEE International Conference on Data Engineering (2008)

A Clustering Approach to Constrained Binary Matrix Factorization

Peng Jiang, Jiming Peng, Michael Heath, and Rui Yang

Abstract. In general, *binary matrix factorization* (BMF) refers to the problem of finding two binary matrices of low rank such that the difference between their matrix product and a given binary matrix is minimal. BMF has served as an important tool in dimension reduction for high-dimensional data sets with binary attributes and has been successfully employed in numerous applications. In the existing literature on BMF, the matrix product is not required to be binary. We call this *unconstrained* BMF (UBMF) and similarly *constrained* BMF (CBMF) if the matrix product is required to be binary. In this paper, we first introduce two specific variants of CBMF and discuss their relation to other dimensional reduction models such as UBMF. Then we propose alternating update procedures for CBMF. In every iteration of the proposed procedure, we solve a specific binary linear programming (BLP) problem to update the involved matrix argument. We explore the relationship between the BLP subproblem and clustering to develop an effective 2-approximation algorithm for CBMF when the underlying matrix has very low rank. The proposed algorithm can also provide a 2-approximation to rank-1 UBMF. We also develop a randomized algorithm for CBMF and estimate the approximation ratio of the solution obtained. Numerical experiments show that the proposed algorithm for UBMF finds better solutions in less CPU time than several other algorithms in the literature, and the solution obtained from CBMF is very close to that of UBMF.

Keywords: Binary matrix factorization, binary quadratic programming, k-means clustering, approximation algorithm.

Peng Jiang · Michael Heath
Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, 61801
e-mail: {pjiang2,heath}@illinois.edu

Jiming Peng · Rui Yang
Department of ISE, University of Illinois at Urbana-Champaign, Urbana, IL, 61801
e-mail: {pengj,ruiyang1}@illinois.edu

W.W. Chu (ed.), *Data Mining and Knowledge Discovery for Big Data,*
Studies in Big Data 1,
DOI: 10.1007/978-3-642-40837-3_9, © Springer-Verlag Berlin Heidelberg 2014

1 Introduction

Given a binary matrix $G \in \{0,1\}^{m\times n}$, the problem of *binary matrix factorization* (BMF) is to find two binary matrices $U \in \{0,1\}^{m\times k}$ and $W \in \{0,1\}^{k\times n}$ so that the distance between G and the matrix product UW is minimal. In the existing literature, the distance is measured by the square of the Frobenius norm, leading to an objective function $\|G - UW\|_F^2$. BMF arises naturally in applications involving binary data sets, such as association rule mining for agaricus-lepiota mushroom data sets [11], biclustering structure identification for gene expression data sets [28, 29], pattern discovery for gene expression pattern images [24], digits reconstruction for USPS data sets [21], mining high-dimensional discrete-attribute data [12, 13], market basket data clustering [16], and document clustering [29].

Binary data sets occupy a special place in data analysis [16], and it is of great interest to discover underlying clusters and discrete patterns. Numerous techniques such as Principal Component Analysis (PCA) [25] have been proposed to deal with continuous data. For nonnegative matrices, nonnegative matrix factorization (NMF) [14, 15, 17, 30] is used to discover meaningful patterns in data sets. However, these methods cannot be directly applied to analyze binary data sets. The presence of binary features poses a great challenge in the analysis of binary data sets, and it generally leads to NP-hard problems.

In 2003, Koyutürk et al. [11] first proposed an algorithm called PROXIMUS to solve BMF via recursive partitioning. Koyutürk et al. [12] further showed that BMF is NP-hard because it can be formulated as an integer programming problem with 2^{m+n} feasible solutions, even for rank-1 BMF. They showed in [13] that there is no theoretical guarantee on the quality of the solution produced by PROXIMUS. Lin et al. [18] proposed an algorithm theoretically equivalent to PROXIMUS but with lower computation cost. Shen et al. [24] proposed a 2-approximation algorithm for rank-1 BMF by reformulating it as a 0-1 integer linear problem (ILP). Gillis and Glineur [7] gave an upper bound for BMF by finding the maximum edge bicliques in the bipartite graph whose adjacency matrix is G. They also proved that rank-1 BMF is NP-hard.

As discussed above, the matrix product UW is generally not required to be binary for BMF. We call this *unconstrained* BMF (UBMF). Since the matrix G is binary, it is often desirable to have a matrix product that is also binary. We call the resulting problem *constrained* BMF (CBMF), where the matrix product is restricted to the class of binary matrices. CBMF is well suited for certain classes of applications. For example, given a collection of text documents, one may be interested in classifying the documents into groups or clusters based on similarity of content. When CBMF is used for the classification, it is natural to stipulate that each document in the corpus be assigned to only one cluster, in which case the resulting matrix product must be binary.

We note that when the matrix product UW is binary, then there is no difference between the squared Frobenius norm and the l_1 norm of the matrix $G - UW$. As shown in recent study [2], use of the l_1 norm is very helpful in the pursuit of sparse solutions to various problems. However, in the present literature on BMF, the squared Frobenius norm has been used as the objective function. Since in BMF, we are seeking an solution that minimizes the number of nonzero elements of the matrix $G - UW$ whenever UW is binary, we thus propose to use the l_1 norm as the objective function in our new BMF model. As we shall see later, while such a change will not change the objective function value, it will change substantially the solution process.

While CBMF is appealing both in theory and in practical applications, it introduces many quadratic constraints into the corresponding optimization problem, making it extremely hard to solve. The primary target of this work is to introduce two variants of CBMF that involve only linear constraints to ensure that the resulting matrix product is binary. In particular, we explore the relationship between the two variants of CBMF and special classes of clustering problems and use this relation to develop effective approximation algorithms for CBMF. As a byproduct, we also develop an effective approximation algorithm for rank-1 UBMF. A randomized algorithm for CBMF is proposed, along with an estimate of the quality of the solution obtained. Our numerical experiments show that the proposed CBMF models can provide good solutions to classification problems with binary data. Compared with other existing solvers for UBMF in the literature, the algorithms proposed in this work can provide solutions of competitive quality in less computational time.

We note that in [22], Miettinen et al. proposed another way to decompose a binary matrix by solving the so-called discrete basis problem (DBP), where the standard matrix product UW in UBMF is replaced by the boolean product $U \otimes W$. They also considered a special variant of DBP (called binary k-median problem (BKMP)), and suggested a 10-approximation algorithm for BKMP. As we shall see later, BKMP can be viewed as a more restrictive version of a specific variant of CBMF. Consequently, the less restrictive CBMF can always lead to a better objective value. The great flexibility of CBMF also allows us to align the sparse rows or columns of the matrix G with the origin in a suitable space associated with the input data and focus mainly on the identification of some large and dense submatrices of G, which is the primary target in UBMF. Moreover, we propose to solve two variants of CBMF to obtain a better matrix factorization. Such a strategy allows us to effectively obtain a 2-approximation to rank-1 UBMF (which is still NP-hard, as shown in [12]), and the proposed algorithm for rank-1 UBMF is a substantial improvement over several existing algorithms for the same problem in the literature [12, 24].

The paper is organized as follows. In Section 2, we introduce the CBMF problem and present two special variants of CBMF. We also discuss various relationships between UBMF and CBMF. In Section 3, we explore the relationships between the two variants of CBMF and special classes of clustering

problems. A simple way to obtain the so-called l_1 center of a given cluster is also proposed. In Section 4, we present two effective approximation algorithms for CBMF: one deterministic and one randomized. In Section 5, we introduce further variants of CBMF, and these extended CBMF models form a hierarchical approach to UBMF. A simple iterative update scheme is proposed to solve the subproblems in UBMF and extended CBMF. In Section 6, we present test results for the proposed algorithms on both synthetic and real data sets and compare them with existing algorithms. Finally, we offer concluding remarks in Section 7.

A brief note about the notation we use: For any matrix G, g_i denotes its i-th column, and G_{ji} (or $g_{i(j)}$) denotes the j-th element of g_i. We also use g_0 to denote the origin in a suitable space.

2 Unconstrained and Constrained BMF

Given $G \in \{0,1\}^{m\times n}$ and integer $k \ll \min(m,n)$, the *unconstrained binary matrix factorization* (UBMF) problem of rank k is defined as

$$\begin{aligned} \min_{U,W} \quad & \|G - UW\|_F^2 \\ s.t. \quad & U \in \{0,1\}^{m\times k},\ W \in \{0,1\}^{k\times n}. \end{aligned} \tag{1}$$

Note that in the above model, the matrix product UW is not required to be binary. As pointed out in the introduction, since the matrix G is binary, it is often desirable to have a binary matrix product, which leads to the *constrained binary matrix factorization* (CBMF) problem

$$\begin{aligned} \min_{U,W} \quad & \|G - UW\|_F^2 \\ s.t. \quad & U \in \{0,1\}^{m\times k},\ W \in \{0,1\}^{k\times n}, \\ & UW \in \{0,1\}^{m\times n}. \end{aligned} \tag{2}$$

If we replace the squared Frobenius norm in problem (2) by the l_1 norm, then we end up with the optimization problem

$$\begin{aligned} \min_{U,W} \quad & \|G - UW\|_1 \\ s.t. \quad & U \in \{0,1\}^{m\times k},\ W \in \{0,1\}^{k\times n}, \\ & UW \in \{0,1\}^{m\times n}. \end{aligned} \tag{3}$$

The quadratic constraints make problem (3) very hard to solve. To see this, let us temporarily fix one matrix, say U, then we end up with a BLP with linear constraints, which is still nontrivial to solve [4]. One way to reduce the difficulty of problem (3) is to replace the hard quadratic constraints by linear constraints that will ensure that the resulting matrix product remains

binary. For this purpose, we introduce the following two specific variants of CBMF:

$$\begin{aligned} \min_{U,W} \ & \|G - UW\|_1 \\ s.t. \ & U \in \{0,1\}^{m\times k},\ W \in \{0,1\}^{k\times n}, \\ & Ue_k \le e_m. \end{aligned} \tag{4}$$

$$\begin{aligned} \min_{U,W} \ & \|G - UW\|_1 \\ s.t. \ & U \in \{0,1\}^{m\times k},\ W \in \{0,1\}^{k\times n}, \\ & W^T e_k \le e_n. \end{aligned} \tag{5}$$

Here $e_k \in \mathbb{R}^{k\times 1}$ and $e_m \in \mathbb{R}^{m\times 1}$ are vectors of all ones. The constraint $Ue_k \le e_m$ (or $W^T e_k \le e_n$) ensures that every row of U (or every column of W) contains at most one nonzero element, and thus it guarantees that UW is a binary matrix.

Another interesting observation is that for a binary matrix U, all its columns are orthogonal to each other if and only if all the constraints $Ue_k \le e_m$ hold. In other words, the orthogonality of a binary matrix B can be retained by imposing some linear constraints on the matrix itself. This is very different from the case of generic matrices. For example, so-called nonnegative principal component analysis [27] also imposes the orthogonal requirement on the involved matrix argument, and it leads to a challenging optimization problem.

Note that the product matrix is guaranteed to be a binary matrix when $k = 1$. Therefore, we immediately have the following result.

Proposition 2.1. *If $k = 1$, then problems (1) and (3) are equivalent.*

Our next result establishes the relationship between the variants of CBMF and general CBMF when $k = 2$.

Proposition 2.2. *If $k = 2$, then problem (3) is equivalent to either problem (4) or (5).*

Proof. It suffices to prove that if (U, W) is a feasible pair for problem (3), then it must satisfy either $Ue_k \le e_m$ or $W^T e_k \le e_n$. Suppose to the contrary that both constraints $Ue_k \le e_m$ and $W^T e_k \le e_n$ fail to hold, i.e., the i-th row of U and the j-th column of W satisfy

$$U_{i1} + U_{i2} = 2, \quad W_{1j} + W_{2j} = 2.$$

Then it follows immediately that

$$[UW]_{ij} = U_{i1}W_{1j} + U_{i2}W_{2j} = 2 > 1,$$

contradicting to the assumption that (U, W) is a feasible pair for problem (3). Therefore, we have either $Ue_k \leq e_m$ or $W^T e_k \leq e_n$. This completes the proof of the proposition. □

Inspired by Propositions 2.1 and 2.2, one may conjecture that problems (1) and (3) are equivalent when $k = 2$. The following example disproves such a conjecture. Let

$$G = \begin{pmatrix} 1&1&1&1&0&0 \\ 1&1&1&1&0&0 \\ 1&1&1&1&1&1 \\ 0&0&0&1&1&1 \\ 0&0&0&1&1&1 \\ 0&0&0&1&1&1 \end{pmatrix}, \quad U = \begin{pmatrix} 1&0 \\ 1&0 \\ 1&1 \\ 0&1 \\ 0&1 \\ 0&1 \end{pmatrix}, \quad W = \begin{pmatrix} 1&1&1&1&0&0 \\ 0&0&0&1&1&1 \end{pmatrix}.$$

Then one can verify that the matrix pair (U, W) is the unique optimal solution to problem (1), but it is infeasible for problem (3).

We note that if $k \geq 3$, then problem (3) is not equivalent to problem (4) or (5). This can be seen from the following example. Consider the matrix pair (U, W) given by

$$U = \begin{pmatrix} 1&0&1 \\ 0&0&1 \end{pmatrix}, \quad W^T = \begin{pmatrix} 0&1&1 \\ 1&1&0 \end{pmatrix}.$$

One can easily see that (U, W) is a feasible solution to problem (3) but not a feasible solution to problem (4) or (5).

3 Equivalence between CBMF and Clustering

In this section, we explore the relationship between CBMF and classes of special clustering problems. We first consider problem (5). Let us temporarily fix U and consider the resulting subproblem

$$\begin{aligned} \min \ & f(W) = \sum_{i=1}^{n} \|g_i - Uw_i\|_1 \\ s.t. \ & e_k^T w_i \leq 1, \ i = 1, \cdots, n; \\ & w_i \in \{0,1\}^k, \ i = 1, \cdots, n. \end{aligned} \tag{6}$$

It is easy to see that the optimal solution to the above problem can be obtained as follows:

$$w_i(j) = \begin{cases} 1 & \text{if } u_j = \arg\min_{l=0,1,\cdots,k} \|g_i - u_l\|_1 \\ 0 & \text{otherwise} \end{cases}.$$

If $w_i(j) = 1$, we say g_i is assigned to u_j, otherwise g_i is assigned to u_0, the origin of the space $\mathbb{R}^m$. Thus problem (6) amounts to assigning each point g_i

to the nearest centroid in the set $S = \{u_0, u_1, \ldots, u_k\}$. Consequentially, we can cast CBMF (5) as the following specific clustering problem:

$$\begin{aligned} \min_{u_1,\ldots,u_k} \quad & \textstyle\sum_{i=1}^{n} \min_{l=0,1,\cdots,k} \|g_i - u_l\|_1 \\ s.t. \quad & u_j \in \{0,1\}^m, \quad j = 1, \ldots, k. \end{aligned} \tag{7}$$

Though W is not explicitly defined in (7), it is trivial to verify the following result.[1]

Theorem 3.1. *Problems (5) and (7) are equivalent in the sense that they have the same optimal solution set and objective value.*

We remark that problem (7) is very close to classical k-means clustering [20] with two exceptions: One is that one additional center u_0 is used in the assignment process. This additional center allows CBMF to align many sparse columns of G with u_0 and perform the clustering task only for the relatively dense columns of G. Intuitively, this will help to reduce the objective function value in BMF. It is also interesting to note that in [22], the authors consider a more restricted version of problem (5) with constraint $W^T e_k = e_n$ (called BKMP). In other words, every column of G must be assigned to a cluster in BKMP, which shows a key difference between BKMP and CBMF.

Another difference between problem (7) and the classical k-means clustering is that we use the l_1 distance in (7), while the Euclidean distance is used in k-means. A popular approach for k-means clustering is to update the assignment matrix and the cluster center iteratively. Note that in classical k-means clustering, the cluster center is simply the geometric center of all the data points in that cluster.

We next discuss how to find a cluster center to minimize the sum of the l_1 distances. For convenience, we call it the l_1 center of the cluster. Given a cluster consisting of binary data points $C_V = \{v_1, \cdots, v_p\}$, we consider the optimization problem

$$\min \sum_{i=1}^{p} \|v_i - v_c\|_1, \tag{8}$$

for which we have

Theorem 3.2. *Suppose that all the data points v_i of a cluster C_V are binary. Then the l_1 center of the cluster is also binary and can be computed by rounding the geometric center of the cluster to binary.*

Proof. Since the l_1 norm of a vector is defined as the sum of all the absolute values of its elements, it suffices to consider the l_1 center with respect to every element of the data points. For example, suppose that

$$v_1(1) = v_2(1) = \cdots = v_l(1) = 0, \quad v_{l+1}(1) = \cdots = v_p(1) = 1, \quad 1 \le l < p.$$

[1] Problem (4) can also be reformulated as a clustering problem similarly.

Then at the geometric center of the cluster, we have

$$v_{\bar{c}}(1) = \frac{p-l}{p}.$$

On the other hand, it is straightforward to verify that

$$v_c(1) = \begin{cases} 1 \text{ if } l \geq p/2 \\ 0 \text{ otherwise} \end{cases}.$$

This completes the proof of the theorem. □

We mention that the l_1 center is identical to a restricted binary variant of the geometric cluster center considered in [18].

We conclude this section by presenting a sandwich theorem exploring the relationship between the optimal solutions for problem (5) and BKMP in [22].

Theorem 3.3. *For a given matrix G, let $f_c^*(k)$ and $f_b^*(k)$ denote the values of the objective function at the optimal solutions to problems (5) and BKMP in [22], respectively. Then*

$$f_b^*(k+1) \leq f_c^*(k) \leq f_b^*(k).$$

Proof. The proof follows by observing that the optimal solution (k-centers) to BKMP can be used as the starting centers for problem (5), and the optimal solution (k-centers) of problem (5), together with the origin in the input data space can be used as a starting solution for BKMP with $k+1$ centers. □

4 Two Approximation Algorithms for CBMF

In this section, we present two algorithms for CBMF. In the first subsection, we describe a deterministic 2-approximation algorithm for CBMF whose complexity is exponential in terms of k. The algorithm is effective for small k, but it becomes ineffective when k is large. For the latter case, in the second subsection we present another approximation algorithm for CBMF with randomized centers.

4.1 A Deterministic 2-Approximation Algorithm

There have been many effective algorithms proposed for k-means clustering [10]. In particular, Hasegawa et al. [9] introduced a 2-approximation algorithm for k-means clustering that runs in $\mathcal{O}(n^{k+1})$ time. In what follows we modify the algorithm in [9] for the CBMF problem. To describe the new algorithm, we first cast every column of G as a data point in $\mathbb{R}^m$ and denote the resulting data set by $\mathcal{V}_G$, whose cardinality is n. Then we formulate

another set $\mathcal{S}_V(k)$ that consists of all subsets $\mathcal{V}_G$ with a fixed size k. The cardinality of $\mathcal{S}_V(k)$ is $\binom{n}{k}$. We obtain a clustering algorithm for CBMF (5) in Algorithm 1, which tries every subset in $\mathcal{S}_V(k)$ as an initial U.

Algorithm 1: Clustering for CBMF (5)

```
1  for l ← 1 to (n choose k) do
2      Choose the subset s_l ∈ S_V(k) and form initial U by casting every point in
       s_l as its column vector;
3      for i ← 1 to n do
4          Assign g_i to the nearest centroid among u_0, u_1, ..., u_k;
5          for j ← 1 to k do
6              if g_i is assigned to u_j then
7                  w_i(j) = 1;
8              else
9                  w_i(j) = 0;
10             end
11         end
12     end
13     Compute the new l_1 center for every cluster C_p based on the newly
       assigned data points; if there is no change in the l_1 center for every
       p = 1, ..., k then
14         Output U and the corresponding W as the solution;
15     else
16         Update the l_1 center for every cluster and go to line 3;
17     end
18 end
19 Return U and W with the minimum objective value over all the runs.
```

We next consider the approximation ratio of Algorithm 1 for CBMF (5).

Theorem 4.1. *Suppose that $U^* = [u_1^*, \ldots, u_k^*]$ is the global optimal solution of problem (7) with an objective value f_{opt}, and $U = [u_1, \ldots, u_k]$ is the solution output by Algorithm 1 with an objective value $f(U)$. Then*

$$f(U) \leq 2f_{opt}.$$

Proof. Let $C_p = \{g_{p_1}, \ldots, g_{p_d}\}$ denote the p-th cluster with the binary centroid u_p^* at the optimal solution of CBMF for $1 \leq p \leq k$, and C_0 the optimal cluster aligned with u_0. Then we can rewrite the optimal objective value of (7) as

$$f_{opt} = \sum_{p=1}^{k} \sum_{g_i \in C_p} \|g_i - u_p^*\|_1 + \sum_{g_i \in C_0} \|g_i\|_1.$$

Let

$$g_p^* = \arg\min_{i=1,\ldots,d} \|g_{p_i} - u_p^*\|_1. \tag{9}$$

It follows that

$$\begin{aligned}
\sum_{i=1}^{d} \|g_{p_i} - g_p^*\|_1 &= \sum_{i=1}^{d}\sum_{j=1}^{m} |g_{p_i}(j) - g_p^*(j)| \\
&\leq \sum_{i=1}^{d}\sum_{j=1}^{m} |g_{p_i}(j) - u_p^*(j)| + |u_p^*(j) - g_p^*(j)| \\
&= \sum_{i=1}^{d} \|g_{p_i} - u_p^*\|_1 + d\|g_p^* - u_p^*\|_1 \\
&\leq 2\sum_{i=1}^{d} \|g_{p_i} - u_p^*\|_1,
\end{aligned}$$

where the first inequality follows from the triangular inequality for l_1 distance, and the last inequality follows from (9). Therefore, we have

$$\begin{aligned}
f(U) &\leq \sum_{p=1}^{k}\sum_{g_i \in C_p} \|g_i - g_p^*\|_1 + \sum_{g_i \in C_0} \|g_i\|_1 \\
&\leq 2\sum_{p=1}^{k}\sum_{g_i \in C_p} \|g_i - u_p^*\|_1 + \sum_{g_i \in C_0} \|g_i\|_1 \\
&\leq 2\Big(\sum_{p=1}^{k}\sum_{g_i \in C_p} \|g_i - u_p^*\|_1 + \sum_{g_i \in C_0} \|g_i\|_1\Big) \\
&= 2f_{opt},
\end{aligned}$$

where the first inequality is implied by the optimality of U^* and (9). The second inequality holds due to (10). It is straightforward to verify the third inequality, and the last equality follows from (9).

We remark that as one can see from the proof of Theorem 4.1, a 2-approximation solution can also be obtained even when we do not update the cluster centers. This implies that we can obtain a 2-approximation to problem (7) in $\mathcal{O}(mn^{k+1})$ time. Similarly, we can modify Algorithm 1 slightly to obtain a 2-approximation for CBMF (4) in $\mathcal{O}(nm^{k+1})$ time. This implies that the proposed algorithm can find a 2-approximation to CBMF effectively for small k. Moreover, combining Theorem 4.1 and Proposition 2.1, we can derive the following result for UBMF.

Corollary 4.1. *A 2-approximation to UBMF with $k = 1$ can be obtained in $\mathcal{O}(nm^2 + mn^2)$ time by applying Algorithm 1 to problems (4) and (5), clustering both by columns and by rows, respectively, and taking the best result.*

It is worth mentioning that in [24], Shen et al. proposed to solve rank-1 UBMF via reformulating it as an integer linear program that involves nm variables. By solving the corresponding LP relaxation, a 2-approximate solution to rank-1 UBMF was first reported in [24]. In the next subsection, we shall discuss how to use a random starting strategy to improve the efficiency of the algorithm and to obtain a good approximation to CBMF for large k.

4.2 A Randomized Approximation Algorithm

In this subsection we present a $\mathcal{O}(\log k)$ approximation algorithm for CBMF based on randomized centers. Instead of the exhaustive search procedure in Algorithm 1, here we modify slightly the random seed selection process in *kmeans++* [1] to obtain the starting centers. Let $D(x)$ denote the l_1 distance from a data point x to the closest center we have already chosen. We use the following procedure to select the starting centers. Once the starting centers

Algorithm 2: Random Initialization

1.1 Take the origin of the space of the data set $\mathcal{V}$ to be the first center, u_0;
1.2 Choose the next cluster center u_i by selecting $u_i = v' \in \mathcal{V}$ with probability $D(v')/(\sum_{v\in\mathcal{V}} D(v))$;
1.3 Repeat Step 1.2 until all k centers are selected;

are chosen, we can proceed as steps 3-17 of Algorithm 1. For convenience, we call the weighting used in the above procedure D_1 weighting.

As in [1], we need several technical results to prove Theorem 4.2. For notational convenience, let us denote $\mathcal{C}_{opt} = \{\mathcal{C}_0, \mathcal{C}_1, \cdots, \mathcal{C}_k\}$ where every $\mathcal{C}_i$ is the cluster in the optimal solution associated with cluster center u_i^*.

We first consider the cluster $\mathcal{C}_0$ aligned to $u_0 = u_0^*$. We have

Lemma 4.1. *Let $\mathcal{C}_0$ be the cluster in $\mathcal{C}_{opt}$ associated with $u_0 = u_0^*$, and let $f(\mathcal{C}_0)$ denote the objective function value after the clustering process. Then*

$$f(\mathcal{C}_0) \leq f_{opt}(C_0).$$

Proof. The lemma follows from (7) and the fact $u_0 = u_0^*$. □

The next result considers the cluster $\mathcal{C}_i$ for some $i \geq 1$ whose center is selected at random uniformly from the set itself. Though we do not use such a strategy to select the starting centers, the result is helpful in our later analysis.

Lemma 4.2. *Let A be an arbitrary cluster in the final optimal clusters $\mathcal{C}_{opt}$, and let $\mathcal{C}$ be the clustering with the center selected at random uniformly from A. Then*

$$E(f(A)) \leq 2f_{opt}(A).$$

Proof. The proof follows a similar vein as the proof of Lemma 3.1 in [1] with the exception that the Euclidean distance has been replaced by the l_1 distance. Let $c(A)$ be the l_1 center of the cluster in the optimal solution. It follows that

$$\begin{aligned} E(f(A)) &= \sum_{a_0 \in A} \frac{1}{|A|} \sum_{a \in A} \|a - a_0\|_1 \\ &\leq \frac{1}{|A|} \sum_{a_0 \in A} \left(\sum_{a \in A} \|a - c(A)\|_1 + |A| \cdot \|a_0 - c(A)\|_1 \right) \\ &= 2 \sum_{a \in A} \|a - c(A)\|_1. \end{aligned}$$ □

It should be mentioned that the above lemma holds for the cluster $\mathcal{C}_0 \in \mathcal{C}_{opt}$, where all the data points are aligned with u_0. In such a case, we need only to change the l_1 center $c(A)$ to u_0 in the proof of the lemma. We next extend the above result to the remaining centers chosen with the D_1 weighting.

Lemma 4.3. *Let A be an arbitrary cluster in the final optimal clusters $\mathcal{C}_{opt}$, and let $\mathcal{C}$ be an arbitrary clustering. If we add a random center to $\mathcal{C}$ from A, chosen with D_1 weighting. Then*

$$E(f(A)) \leq 4 f_{opt}(A).$$

Proof. Note that for any $a_0 \in A$, the probability that a_0 is selected as the center is $D(a_0)/(\sum_{a \in A} D(a))$. It follows that

$$\begin{aligned} E(f(A)) &= \sum_{a_0 \in A} \frac{D(a_0)}{\sum_{a \in A} D(a)} \sum_{a \in A} \min(D(a), \|a - a_0\|_1) \\ &\leq \frac{1}{|A|} \sum_{a_0 \in A} \frac{\sum_{a \in A} (D(a) + \|a - a_0\|_1)}{\sum_{a \in A} D(a)} \sum_{a \in A} \min(D(a), \|a - a_0\|_1) \\ &\leq \frac{1}{|A|} \sum_{a_0 \in A} \frac{\sum_{a \in A} D(a)}{\sum_{a \in A} D(a)} \sum_{a \in A} \|a - a_0\|_1 + \frac{1}{|A|} \sum_{a_0 \in A} \frac{\sum_{a \in A} \|a - a_0\|_1}{\sum_{a \in A} D(a)} \sum_{a \in A} D(a) \\ &= \frac{2}{|A|} \sum_{a_0 \in A} \sum_{a \in A} \|a_0 - a\|_1 \leq 4 f_{opt}(A), \end{aligned}$$

where the first inequality follows from the triangle inequality for l_1 distance, and the last inequality follows from Lemma 4.2. □

The following lemma resembles Lemma 3.3 in [1], with a minor difference in the constant used in the estimate. For completeness, we include its proof here.

Lemma 4.4. *Let $\mathcal{C}$ be an arbitrary clustering. Choose $T > 0$ 'uncovered' clusters from $\mathcal{C}_{opt}$, and let $\mathcal{V}_u$ denote the set of points in these clusters, with*

$\mathcal{V}_c = \mathcal{V} - \mathcal{V}_u$. Suppose we add $t \leq T$ random centers to $\mathcal{C}$, chosen with D_1 weighting. Let $\mathcal{C}'$ denote the resulting clustering. Then

$$E(f(\mathcal{C}')) \leq (1 + H_t)(f(\mathcal{V}_c) + 4f_{opt}(\mathcal{V}_u)) + \frac{T-t}{T} f(\mathcal{V}_u),$$

where H_t denotes the harmonic sum, $1 + \frac{1}{2} + \cdots + \frac{1}{t}$.

Proof. We prove this by induction, showing that if the result holds for $(t - 1, T)$ and $(t-1, T-1)$, then it also holds for (t, T). Thus, it suffices to check the base cases $t = 0, T > 0$ and $t = T = 1$.

The case $t = 0$ follows easily from the fact that $1 + H_t = (T - t)/T = 1$. Suppose $T = t = 1$. We choose the new center from the one uncovered center with probability $f(\mathcal{V}_u)/f(\mathcal{V})$. It follows from Lemma 4.3 that

$$E(f(\mathcal{C}')) \leq f(\mathcal{V}_c) + 4f_{opt}(\mathcal{V}_u).$$

Because $f(\mathcal{C}') \leq f(\mathcal{V})$, even if we choose a center from a covered cluster, we thus have

$$E(f(\mathcal{C}')) \leq \frac{f(\mathcal{V}_u)}{f(\mathcal{V})} (f(\mathcal{V}_c) + 4f_{opt}(\mathcal{V}_u)) + \frac{f(\mathcal{V}_c) f(\mathcal{V})}{f(\mathcal{V})} \leq 2f(\mathcal{V}_c) + 4f_{opt}(\mathcal{V}_u).$$

Since $1 + H_t = 2$, the lemma holds for both cases.

We next proceed to the inductive step. We first consider the case where the center is chosen from a covered cluster, which happens with probability $f(\mathcal{V}_c)/f(\mathcal{V})$. Since adding the new center will only decrease the objective value, by applying the inductive hypothesis with the same choice of covered clusters, but with t decreased by 1, we can conclude that the contribution to $E(f(\mathcal{C}'))$ in this case is at most

$$\frac{f(\mathcal{V}_c)}{f(\mathcal{V})} \left((f(\mathcal{V}_c) + 4f_{opt}(\mathcal{V}_u)) (1 + H_t) + \frac{T-t+1}{T} f(\mathcal{V}_u) \right). \tag{10}$$

Suppose that the first center is chosen from some uncovered cluster A, which happens with probability $f(A)/f(\mathcal{V})$). Let p_a be the conditional probability that we choose $a \in A$ as the center given the fact that the center is from A, and $f_a(A)$ denotes the objective value when a is used as the center. Adding A to the covered center (thus decreasing both T and t by 1) and applying the inductive hypothesis again, we have

$$\begin{aligned} E(f(\mathcal{C}')) &\leq \frac{f(A)}{f(\mathcal{V})} \sum_{a \in A} p_a \left((f(\mathcal{V}_c) + f_a(A) + 4f_{opt}(\mathcal{V}_u) - 4f_{opt}(A)) (1 + H_{t-1}) + \frac{T-t}{T-1} (f(\mathcal{V}_u) - f(A)) \right) \\ &\leq \frac{f(A)}{f(\mathcal{V})} \left((f(\mathcal{V}_c) + 4f_{opt}(\mathcal{V}_u)) (1 + H_{t-1}) + \frac{T-t}{T-1} (f(\mathcal{V}_u) - f(A)) \right), \end{aligned}$$

where the last inequality follows from Lemma 4.3. Recalling the power-mean inequality, we have

$$\sum_{A\in\mathcal{V}_u} f(A)^2 \geq \frac{1}{T} f(\mathcal{V}_u)^2.$$

Summing over all uncovered clusters, we obtain

$$\begin{aligned} E(f(\mathcal{C}')) &\leq \frac{f(\mathcal{V}_u)}{f(\mathcal{V})}\left(f(\mathcal{V}_c)+4f_{opt}(\mathcal{V}_u)\right)(1+H_{t-1}) + \frac{T-t}{(T-1)f(\mathcal{V})}\left(f(\mathcal{V}_u)^2 - \frac{f(\mathcal{V}_u)^2)}{T}\right) \\ &= \frac{f(\mathcal{V}_u)}{f(\mathcal{V})}\left(\left(f(\mathcal{V}_c)+4f_{opt}(\mathcal{V}_u)\right)(1+H_{t-1}) + \frac{T-t}{T} f(\mathcal{V}_u)\right). \end{aligned} \tag{11}$$

From (10) and (11) we derive

$$\begin{aligned} E(f(\mathcal{C}')) &\leq \left(f(\mathcal{V}_c)+4f_{opt}(\mathcal{V}_u)\right)(1+H_{t-1}) + \frac{T-t}{T} f(\mathcal{V}_u) + \frac{f(\mathcal{V}_c)f(\mathcal{V}_u)}{Tf(\mathcal{V})} \\ &\leq \left(f(\mathcal{V}_c)+4f_{opt}(\mathcal{V}_u)\right)(1+H_{t-1}+\frac{1}{T}) + \frac{T-t}{T} f(\mathcal{V}_u) \\ &\leq \left(f(\mathcal{V}_c)+4f_{opt}(\mathcal{V}_u)\right)(1+H_{t-1}+\frac{1}{t}) + \frac{T-t}{T} f(\mathcal{V}_u). \end{aligned} \tag{12}$$

This completes the proof of the lemma. □

Now we are ready to state the main result in this subsection.

Theorem 4.2. *If the starting centers are selected by the random initialization Algorithm 2, then the expected objective function value $E(f) = E(f(U))$ satisfies*

$$E(f(U)) \leq 4(\log k + 2) f_{opt}.$$

Proof. Consider the clustering $\mathcal{C}$ after all the starting centers have been selected. Let A denote the cluster in $\mathcal{C}_{opt}$ from which we choose u_1. Applying Lemma 4.4 with $t = T = k-1$, and with $\mathcal{C}_0$ and A the only two possibly covered clusters, we have

$$\begin{aligned} E(f(\mathcal{C})) &\leq \left(f(\mathcal{C}_0) + f(A) + 4f(\mathcal{C}_{opt}) - 4f_{opt}(\mathcal{C}_0) - 4f_{opt}(A)\right)(1+H_{k-1}) \\ &\leq 4(2+\log k) f(\mathcal{C}_{opt}), \end{aligned}$$

where the last inequality follows from Lemma 4.1, Lemma 4.3, and the fact that $H_{k-1} \leq 1 + \log k$. □

It is worth mentioning that compared with Theorem 3.1 in [1], the approximation ratio in the above theorem is sharper, due to the use of the l_1 norm.

5 Extension of CBMF

In the previous sections, we have focused on two specific variants of CBMF, (4) and (5). In this section we introduce several new variants of CBMF and explore their relationships to UBMF. Note that if we use the l_1 norm as the objective function, then the optimization model for UBMF can be written as

$$f(U,W) = \|G - UW\|_1 = \sum_{i=1}^{n} \|g_i - \sum_{j=1}^{k} w_i(j)u_j\|_1. \tag{13}$$

If we temporarily fix one matrix argument, say U, then we obtain the BLP subproblem

$$\begin{aligned} \min_{w_i} \;\; & f(w_i) = \|g_i - \sum_{j=1}^{k} w_i(j)u_j\|_1 \\ s.t. \;\; & w_i \in \{0,1\}^k. \end{aligned} \tag{14}$$

As in our discussion of CBMF in Section 4, we can cast g_i and u_i as points in $\mathbb{R}^m$. Consequently, problem (13) reduces to the problem of assigning each point g_i to the nearest linear combination of $u_1, \ldots, u_k$. Let $S(u_1, \ldots, u_k)$ denote the set of all possible linear combinations, i.e., $S = \{s_1, \cdots, s_{2^k}\}$, with $s_l = \sum_{j=1}^{k} \alpha_l(j)u_j$ and $\alpha_l(j) \in \{0,1\}$. It is easy to see that $|S| = 2^k$. Using the above notation, it is easy to see that the optimal solution to problem (13) can be obtained as follows:

$$w_i(j) = \begin{cases} 1 & \text{if } s_l = \arg\min_{s \in S} \|g_i - s\|_1 \text{ and } \alpha_l(j) = 1 \\ 0 & \text{otherwise} \end{cases}, \tag{15}$$

where $j = 1, \cdots, k$. Based on this relation, we reformulate UBMF (1) as the following clustering problem:

$$\begin{aligned} \min_{u_1,\ldots,u_k} \; & \textstyle\sum_{i=1}^{n} \min_{c \in S(u_1;\ldots;u_k)} \|g_i - c\|_1 \\ s.t. \;\; & u_j \in \{0,1\}^m, \quad \forall j = 1, \ldots, k. \end{aligned} \tag{16}$$

We next establish a sandwich theorem between the optimal objective values of CBMF and UBMF.

Theorem 5.1. *For a given matrix G, let $f_u^*(k)$ and $f_c^*(k)$ denote the values of the objective function at the optimal solutions to problems (1) and (5), respectively, where k is the rank constraint on matrices U and W. Then*

$$f_c^*(2^k - 1) \leq f_u^*(k) \leq f_c^*(k).$$

Proof. The relation $f_u^*(k) \leq f_c^*(k)$ holds because the optimal solution of rank-k CBMF is also a feasible solution for rank-k UBMF.

Now we proceed to prove the relation $f_c^*(2^k - 1) \leq f_u^*(k)$. Denote $U = \{u_1, u_2, \cdots, u_k\}$ the matrix in the optimal solution to rank-k UBMF, and $S(u_1, \cdots, u_k)$ the set of all possible combinations of the columns of U. It follows immediately that the matrix W can be obtained from the assignment process (15). Note that for every element $s_l \in S(u_1, \cdots, u_k), l = 1, \cdots, 2^k$, we can construct another binary vector $\bar{s}_l$ by

$$\bar{s}_l^i = \begin{cases} 1 & \text{if } s_l^i > 1 \\ s_l^i & \text{otherwise} \end{cases}, \quad i = 1, \cdots, n. \tag{17}$$

Accordingly we obtain another set $\bar{S}$ that contains all the elements $\bar{s}_l$. Since the matrix G is binary, for every column g_i of G, we have

$$\|g_i - \bar{s}_l\|_1 \leq \|g_i - s_l\|_1.$$

Note that the set $\bar{S}$ can be used as a starting matrix in CBMF with rank $2^k - 1$. It follows from (7) and (16) that $f_c^*(2^k - 1) \leq f_u^*(k)$. This completes the proof of the theorem. □

To illustrate, we recall Example (6). It is straightforward to verify the following relation

$$f_c^*(3) = 0 < f_u^*(2) = 1 < f_c^*(2) = 2.$$

We remark that from Theorem 5.1, one can see that there might be a large gap between UBMF and the two variants of CBMF, in particular when k is reasonably large. This is not surprising due to the extra constraint in CBMF. For example, in problem (5) we imposed the constraint $W^T e_k \leq e_n$. Note that because W is binary, the relation $W^T e_k \leq k e_n$ always holds. In other words, we can view UBMF as a special variant of CBMF where the constraint $W^T e_k \leq k e_n$ is redundant. Based on this observation, we can also replace the constraint in problem (5) by

$$W^T e_k \leq t e_n, \quad 1 < t < k.$$

Let us denote the corresponding optimization model by CBMF(t), and the optimal objective value by $f_{CBMF(t)}^*$. Then one can easily show that

$$f_{CBMF(1)}^* \geq f_{CBMF(2)}^* \cdots \geq f_{CBMF(k)}^* = f_{UBMF(k)}^*.$$

This shows that UBMF can be approached via a series of CBMF models.

On the other hand, though problem (13) can be solved via the assignment (15) for a fixed U, the procedure has complexity 2^k, which is still very high for large k. In what follows we present a simple iterative procedure for problem (13) that reduces the objective function value step by step. For this, we first rewrite the objective in (13) as

$$f(U,W) = \|G - UW\|_1 = \|G - UW + u_{:i}w_{i:} - u_{:i}w_{i:}\|_1 = \|\tilde{G} - u_{:i}w_{i:}\|_1 = f_1(w_{i:}),$$

where $w_{i:}$ denotes the i-th row of W, and $u_{:i}$ the i-th column of U. Note that the matrix $\tilde{G}$ is independent of $w_{i:}$, since the terms involving $w_{i:}$ cancel. Now let us temporarily fix $\tilde{G}$ and consider the problem

$$\min_{w_{i:} \in \{0,1\}^n} f_1(w_{i:}). \tag{18}$$

Define $\tilde{w}_{i:} = u_{:i}^T \tilde{G} / e_n^T u_{:i}$. From Theorem 3.2 we can obtain the optimal solution (of problem (18)) as follows

$$w_{ij}^* = \begin{cases} 0 \text{ if } \tilde{w}_{ij} < \frac{1}{2} \\ 1 \text{ otherwise} \end{cases}. \tag{19}$$

Algorithm 3: Iterative Algorithm for UBMF (13)

1 For $i = 1, \ldots, k$;
2 Update $w_{i:}$ via (19);
3 Repeat above process until no improvement is obtained.

It should be pointed out that though the above procedure can reduce the objective value of problem (13) and is easy to implement, the solution provided might not be optimal.

6 Numerical Results

In this section we report numerical results for our proposed algorithms for both CBMF and UBMF on some test data sets, and compare them with other existing algorithms for UBMF. For efficiency considerations, we implemented only the randomized algorithm for CBMF analyzed in Section 4.2. Since the solution from CBMF is also feasible for UBMF, the output from CBMF can be used as initial matrices for UBMF. Then we apply Algorithm 3 to obtain a solution for UBMF. Accordingly, we call such an algorithm hybrid UBMF. We also compare the solutions of UBMF and CBMF. All numerical tests were conducted using MATLAB R2012 and performed on a 64-bit Windows 7 system with Intel Core2 Quad 2.66 GHz CPU and 4 GB RAM.

For numerical comparison, we apply PROXIMUS to UBMF [11], which splits a data set based on the entries of a binary vector and performs recursive partitioning in the direction of such vectors. When the rank is fixed, we apply the rule proposed in [18] to find the best solution among all possible solutions of the desired rank. We also coded the 2-approximation algorithm [24], denoted by ILP, for rank-1 UBMF. ILP reformulates UBMF as a 0-1 integer programming program and finds an approximate solution by using its linear programming relaxation. We also implemented a penalty function algorithm given in [29]. We chose the penalty function algorithm over the thresholding algorithm in [29] because initial testing showed the thresholding algorithm to be very time-consuming.

Data sets from three different categories were tested. Synthetic data sets are first used to test the effectiveness and efficiency of the proposed algorithms. We also use gene expression data sets to find bicluster structures. In the last part of this section we apply the proposed algorithms in this work

to document clustering and compare results with those from the standard k-means algorithm and the PROXIMUS algorithm in [11].

6.1 Synthetic Dataset

First, we generated some synthetic data sets to test the effectiveness of the proposed algorithms. We randomly generate two binary matrices U and V and round all nonzero entries of the product matrix ($G = UV$) to 1. Here we use the l_1 norm to measure the approximation error. Since the proposed algorithm is randomized, to be fair we repeat the algorithm 20 times and average the outputs.

Table 1 Numerical Results for Synthetic Data Set

Algorithm			CBMF		Hybrid UBMF		PROXIMUS		ILP	
n	m	k	Time	Obj	Time	Obj	Time	Obj	Time	Obj
100	50	5	0.0999	16	0.2675	16	0.6617	42	3.0954	62
			0.0918	18	0.1223	13	0.2454	42	1.8560	58
			0.0587	24	0.1568	20.5	0.6031	43	1.3335	61
500	200	10	0.7285	27	4.5933	23	1.2405	200	3.7188	300
			0.7379	32	5.4058	30	1.1135	74	4.4731	74
			0.7592	36	4.6835	36	1.2614	89	4.4259	89
800	400	10	2.5793	100	26.8796	87	12.1947	317	24.1523	499
			2.2930	122	29.6951	97	11.1525	317	37.0757	486
			2.3735	100	30.1860	92	10.3123	141	22.0120	141

As can be observed in Table 1, the hybrid UBMF algorithm always produced the best recovery matrix among all algorithms tested. Also, the CBMF algorithm significantly outperforms the PROXIMUS and ILP algorithms.

6.2 Metagene Pattern Discovery

Here we use the proposed algorithms to reduce the dimension of gene expression data and find metagene patterns. The goal is to find a small number of metagenes such that the gene expression pattern of samples can be approximated as a linear combination of these metagenes. We consider a data set consisting of gene expression levels of N genes in M samples (where normally $N \gg M$), represented by matrix G of size $N \times M$. The rows of G contain the expression levels of the N genes in the M samples. We seek a rank k approximation $G = UV$ where $U \in \mathbb{R}^{N \times k}$ and $V \in \mathbb{R}^{k \times M}$. Each column of matrix U represents a metagene from N genes, and each column of V stands for the metagene expression pattern of the corresponding sample. We are mostly interested in the biclustering case, $k = 2$.

The *match score* is used to evaluate clustering performance. If M_1, M_2 are biclustering sets, then the match score in attribute dimension (gene dimension) of M_1 with respect to M_2 is given by

$$S(M_1, M_2) = \frac{1}{|M_1|} \sum_{(G_1,C_1)\in M_1} \max_{(G_2,C_2)\in M_2} \frac{|G_1 \cap G_2|}{|G_1 \cup G_2|},$$

where $\{(G_1, C_1) \in M_1\}$ is a gene-sample partition in M_1. In the experiment, we use the implanted/optimal biclustering structure as M_1, and the computed biclustering structure as M_2. The data sets we employ are commonly used in the bioinformatics community [3, 23]. Specifically, they are ALL/AML, leukemia, and lung cancer data sets. A brief summary of the data sets is given in Table 2. We follow the procedure described in [23] to generate the data. Based on different discretization schemes, we have two different data sets and will report their results separately. The match scores of the various

Table 2 Gene Expression Data Set

Data Set	Gene	Sample
ALL_AML	3051	38
Leukemia No. 1	15060	30
Leukemia No. 2	15060	170
Lung Cancer	9036	82

algorithms are reported in Tables 3 and 4. As we can see from both tables, the hybrid UBMF reports the highest match score among all algorithms tested. PROXIMUS is consistently worse than CBMF. Sometimes the penalty function algorithm is able to produce similar results as the Hybrid UBMF does, but with significantly longer computational time. For all gene expression data sets, ILP fails to report any reasonable outputs.

Table 3 Numerical Results for Gene Expression Data Set 1

Algorithm	CBMF		Hybrid UBMF		PROXIMUS		Penalty Function	
Data Set	Time	Score	Time	Score	Time	Score	Time	Score
AML_ALL	0.4906	0.7675	0.4272	0.7698	43.4760	0.7525	1.2970	0.8019
Leukemia-1	1.2762	0.8376	1.3501	0.8411	62.2387	0.8338	7.3132	0.8070
Leukemia-2	1.2521	0.8297	1.4421	0.8301	62.2819	0.8165	47.9095	0.8071
Lung Cancer	2.4525	0.8343	2.7295	0.8343	57.5884	0.7877	14.1992	0.7197

Table 4 Numerical Results for Gene Expression Data Set 2

Algorithm	CBMF		Hybrid UBMF		PROXIMUS		Penalty Function	
Data Set	Time	Score	Time	Score	Time	Score	Time	Score
AML_ALL	0.3709	0.7050	0.3653	0.7258	44.0312	0.6986	1.4123	0.7350
Leukemia-1	1.3717	0.7478	1.5330	0.7550	43.9606	0.7397	11.3092	0.6801
Leukemia-2	10.015	0.7373	12.795	0.7400	150.189	0.7162	36.9800	0.6766
Lung Cancer	2.3814	0.7643	2.4908	0.7643	71.0308	0.7315	17.2165	0.6599

6.3 Document Clustering

In this experiment, we apply the proposed algorithms to some text mining applications. We first choose the most frequent terms in the documents to form the term space and project each document into the term space, which will generate a binary matrix. In this way we represent the documents using a binary vector space model where each document is a binary vector in the term space. The terms are grouped into k different classes by definition. We then apply our CBMF algorithm to obtain k clusters. The idea is to use the resulting clusters to approximate those classes and predict new ones. A well-known measure of the clustering performance is *accuracy*. For each cluster, the accuracy is defined as the similarity between it and its closest class and sum over all clusters:

$$accuracy = \sum_k \max_m T(C_k, L_m)/N,$$

where the $\{C_k\}$ is the set of clusters we obtain and $\{L_m\}$ the set of labels, N is the total number of documents, and $T(\cdot)$ is the number of entities belonging to class m that are assigned to cluster k. All the data sets are from [6], as summarized below.

- **20 newsgroups.** The 20 Newsgroups data set is a collection of approximately 20000 newsgroup documents, partitioned evenly across 20 different newsgroups. It was originally collected by Ken Lang. The 20 newsgroups collection has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering. We adopt two subsets of this data set for our experiment.
- **CNAE-9.** This is a data set containing 1080 documents of free text business descriptions of Brazilian companies categorized into a subset of nine categories cataloged in a table called National Classification of Economic Activities (CNAE). The number of attributes is 857. This data set is highly sparse (99.22% of the matrix is filled with zeros).
- **Internet Ads.** This data set represents a set of possible advertisements on Internet pages. The features encode the geometry of the image (if available) as well as phrases occuring in the URL, the URL and alt text of the image,

the anchor text, and words occuring near the anchor text. The task is to predict whether an image is an advertisement or not.

A brief summary of the data sets is given in Table 5 and the results are reported in Table 6. Each entry is the clustering accuracy of the column method on the corresponding row data set and a result of averaging 10 runs. The ILP and penalty function algorithms failed to report on these data sets. Again, the hybrid UBMF beats other algorithms in terms of accuracy. Also it is clear that CBMF and Hybrid UBMF algorithms work well on highly sparse data sets.

Table 5 Text Mining Data Set

Name	# documents	# attributes	# classes
20 newsgroups-1	11269	1000	20
20 newsgroups-2	7505	1000	20
CNAE-9	1080	856	9
Internet Ads	3279	1555	2

Table 6 Text Mining Results: Accuracy

Name	CBMF	Hybrid UBMF	PROXIMUS	k-means
20 newsgroups-1	0.3605	0.4257	0.0823	0.3371
20 newsgroups-2	0.3693	0.3947	0.0809	0.2945
CNAE-9	0.2888	0.3010	0.3000	0.3076
Internet Ads	0.2104	0.2252	0.1061	0.1876

7 Conclusions

There are several ways to extend the results in this paper. One possible direction is to develop more effective algorithms for both CBMF and UBMF, in particular for reasonably large k. For example, in Algorithm 3 we present a simple iterative procedure to reduce the objective function in UBMF. Since such a procedure might not provide an optimal solution to problem (13), it is of interest to incorporate some local search heuristics to further reduce the objective function. Another possible direction is to consider the scenario of two different types of mismatched entries: 0-to-1 and 1-to-0. In the current CBMF model, we minimize the sum of the two types of mismatched entries without any preference between them. However, in many practical applications, it might be helpful to include such a preference in the optimization model. In such a case, we can extend the current CBMF model by using different weights for each type of error and then design effective algorithms for the new model. More study is needed to address these issues.

Acknowledgements. We would like to thank an anonymous referee who provided many useful suggestions to improve the presentation of this paper. We also thank Prof. Yaxiang Yuan for suggesting use of the l_1 norm to simplify the analysis in the paper. The research for this work was supported by AFOSR grant FA9550-09-1-0098, NSF grants DMS 09-15240 ARRA and CMMI-1131690, and the endowment for the Fulton Watson Copp Chair in Computer Science at the University of Illinois at Urbana-Champaign.

References

[1] Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. In: Proc. Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027–1035 (2007)
[2] Bruckstein, A.M., Donoho, D.L., Elad, M.: From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Review 51(1), 34–81 (2009)
[3] Brunet, J., Tamayo, P., Golub, T.R., Mesirov, J.P., Lander, E.S.: Metagenes and molecular pattern discovery using matrix factorization. Proc. National Academy Sciences (2004)
[4] Chaovalitwongse, W., Androulakis, I.P., Pardalos, P.M.: Quadratic integer programming: Complexity and equivalent forms. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization (2007)
[5] Crama, Y., Hansen, P., Jaumard, B.: The basic algorithm for pseudo-Boolean programming revisited. Discrete Appl. Math. 29, 171–185 (1990)
[6] Frank, A., Asuncion, A.: UCI Machine Learning Repository, School of Information and Computer Science, University of California, Irvine, CA (2010), `http://archive.ics.uci.edu/ml`
[7] Gillis, N., Glineur, F.: Using underapproximations for sparse nonnegative matrix factorization. Pattern Recognition 43(4), 1676–1687 (2010)
[8] Hammer, P.L., Rudeanu, S.: Boolean Methods in Operations Research and Related Areas. Springer, New York (1968)
[9] Hasegawa, S., Imai, H., Inaba, M., Katoh, N., Nakano, J.: Efficient algorithms for variance-based k-clustering. In: Proc. First Pacific Conf. Comput. Graphics Appl., Seoul, Korea, pp. 75–89. World Scientific, Singapore (1993)
[10] Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. 31(3), 264–323 (1999)
[11] Koyutürk, M., Grama, A.: PROXIMUS: a framework for analyzing very high dimensional discrete-attributed datasets. In: ACM SIGKDD, pp. 147–156 (2003)
[12] Koyutürk, M., Grama, A., Ramakrishnan, N.: Compression, clustering, and pattern discovery in very high-dimensional discrete-attribute data sets. IEEE TKDE 17(4), 447–461 (2005)
[13] Koyutürk, M., Grama, A., Ramakrishnan, N.: Nonorthogonal decomposition of binary matrices for bounded-error data compression and analysis. ACM Trans. Math. Softw. 32(1), 33–69 (2006)
[14] Lee, D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)

[15] Lee, D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Neural Information Processing Systems, NIPS (2001)
[16] Li, T.: A general model for clustering binary data. In: ACM SIGKDD, pp. 188–197 (2005)
[17] Li, T., Ding, C.: The relationships among various nonnegative matrix factorization methods for clustering. In: ICDM, pp. 362–371 (2006)
[18] Lin, M.M., Dong, B., Chu, M.T.: Integer Matrix Factorization and Its Application (2009) (preprint)
[19] Lloyd, S.: Least squares quantization in PCM. IEEE Trans. Inform. Theory, 129–137 (1982)
[20] McQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. University of California Press, Berkeley (1967)
[21] Meeds, E., Ghahramani, Z., Neal, R.M., Roweis, S.T.: Modeling dyadic data with binary latent factors. In: Neural Information Processing Systems 19 (NIPS 2006), pp. 977–984 (2006)
[22] Miettinen, P., Mielikäinen, T., Gionis, A., Das, G., Mannila, H.: The discrete basis problem. IEEE Trans. Knowledge Data Engineering 20(10), 1348–1362 (2008)
[23] Prelić, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., Hennig, L., Thiele, L., Zitzler, E.: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 22(9), 1122–1129 (2006)
[24] Shen, B.H., Ji, S., Ye, J.: Mining discrete patterns via binary matrix factorization. In: ACM SIGKDD, pp. 757–766 (2009)
[25] Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
[26] van Uitert, M., Meuleman, W., Wessels, L.: Biclustering sparse binary genomic data. J. Comput. Biol. 15(10), 1329–1345 (2008)
[27] Zass, R., Shashua, A.: Non-negative sparse PCA. In: Advances in Neural Information Processing Systems (NIPS), vol. 19, pp. 1561–1568 (2007)
[28] Zhang, Z.Y., Li, T., Ding, C., Ren, X.W., Zhang, X.S.: Binary matrix factorization for analyzing gene expression data. Data Min. Knowl. Discov. 20(1), 28–52 (2010)
[29] Zhang, Z.Y., Li, T., Ding, C., Zhang, X.S.: Binary matrix factorization with applications. In: ICDM, pp. 391–400 (2007)
[30] Zdunek, R.: Data clustering with semi-binary nonnegative matrix factorization. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 705–716. Springer, Heidelberg (2008)

Erratum: Data Mining and Knowledge Discovery for Big Data

Wesley W. Chu

Department of Computer Science,
University of California,
Los Angeles,
USA

W.W. Chu (ed.), *Data Mining and Knowledge Discovery for Big Data,*
Studies in Big Data 1,
DOI: 10.1007/978-3-642-40837-3, © Springer-Verlag Berlin Heidelberg 2014

DOI 10.1007/978-3-642-40837-3_10

In the original online version of this volume, the foreword is missing. It is given on the next page.

Foreword

Modern science is characterized not only by a rapid development pace but a necessity to transcend boundaries of traditional fields and a necessary bridging gaps between fields. This is a result of an increasing complication of the present world in which all systems operate in a highly complex and interwoven environment so that results from various fields of science, sometimes very distant from one another, must be used for analysis and solution. An inherent part of such a new reality in which science, and also technology, must operate is that it has to discover new challenges and be able to respond to them both quickly, and effectively and efficiently to stay competitive in the difficult environment in which various human activities, including research, must fight for recognition and financing which are implied to a large extend by the fact if they can to solve real problems of a crucial and growing relevance to the society.

One of such problems we are facing in recent years, may be a decade, is a so called *Big Data*. *Big Data* can be found more and more both in serious scientific publications and presentations and in the media. Basically, though various definitions can be found, Big Data is an emerging paradigm that applies to what can and should be done with sets of data which are beyond not only the human cognitive capabilities but also beyond what the commonly employed software tools and packages can do in the sense of capturing, managing, processing, displaying, etc. the data within a time that would be acceptable for a practical use, for instance by the human user. Needless to say that this problem has been triggered by an unprecedented growth of data sets produced in any human activity as the cost of memory becomes negligible, everything is stored, and sets of data stored become larger and larger. Moreover, one should take into account that the complexity of those data constantly increases as more and more the data stored contain in addition to traditional numeric data, also texts, pictures, videos, voice, etc. etc. Most are unstructured which make the problem even more difficult. All that data comes from various sources exemplified by social media, sensors, scientific experiments, surveillance data, video and image archives, texts from the

Internet, medical records, business transactions, web logs; etc. etc. It is quite obvious that to effectively and efficiently handle problems with such kinds of data, various solutions should be applied exemplified by broadly perceived distributed computing, massively parallel processing databases, scalable storage systems, cloud computing platforms, etc. which should preferably operate in a synergistic way.

Scientific publishers are always trying to be a good "mirror" of the scientific community and to keep themselves up-to-date with the main new research developments. Therefore they have to respond timely and in an appropriate way to any major new research directions and challenges that strongly emerge and quickly become subjects of intensive research. The new field of Big Data is a perfect example of such a new challenge that is taking by storm the interest of scientific communities all over the world.

We have therefore decided to launch the new book series *Studies in Big Data* in the Springer scientific program. This new series aims to serve the needs of both prospective authors and readers by providing up to date account and coverage of the newest developments in the broadly perceived "Big Data" area, both in a foundational and theoretical and applied dimension. The new book series will include both a state of the art, even textbook like text, and highly advanced and specialized books and volumes. With such a broad we hope to best serve the scientific and technological communities, and fulfill needs of many readers, of different needs, backgrounds and credentials.

We are very happy to start this new series with the present volume " Data Mining and Knowledge Discovery for Big Data: Methodologies, Challenges, and Opportunities" edited by Professor Wesley Chu from the University of California at Los Angeles, USA. Professor Chu's illustrious career spanned over a couple of decades and it is difficult to even list all of his novel and pioneering contributions. To just name a few, in the beginning of his career he worked on the design of large-scale computers at IBM; on computer communication and distributed databases at Bell Labs, and then he continued work on computer communication and networks, distributed databases, memory management, real-time distributed processing systems, and statistical multiplexing, the development of ATM networks, etc. Among his pioneering works, one should mention those on file allocation and directory design for distributed databases that helped develop of domain name servers for the web and current cloud computing systems. Moreover, he has obtained many original results in the area of broadly perceived intelligent systems exemplified by intelligent (knowledge-based) information systems and knowledge acquisition for large information systems, relaxation of query constraints that led to the development of CoBase, a cooperative database system for structured data, and KMed, a knowledge-based multimedia medical image system, etc. The list of his original achievements is much longer, indeed.

This first volume, *Data Mining and Knowledge Discovery for Big Data: Methodologies, Challenges and Opportunities*, is very proper for the launching of the new book series. First, it deals with the field of data mining and

knowledge discovery, a field of science that has enjoyed a huge popularity among researchers, scholars and practitioners due to its sheer usefulness for virtually all areas of science and technology, even – as one can say – virtually all human activities as it tries to discover some relations that cannot be seen by the humans but can be of use while solving problems. Clearly, such a problem – which is simple to verbally state but difficult to formalize and solve – has implied many foundational, analytic, and implementation challenges which have triggered new ideas and results by researchers and scholars, followed by engineers and other practitioners.

As the field of data mining and knowledge discovery has matured, people have started to apply their tools and techniques to more and more complex, and less and less structured information exemplified by textual or multimedia data, nonlinear dynamics of even spatiotemporal form, and to sets of data and problems in more and more "soft sciences" in which such a complex and unstructured information prevails. Moreover, due to the progress in IT/ICT, better and better capabilities and specifications of hardware and software solutions are available, and increased and general use of all kinds of distributed computing systems, the "data sets" in question have become different that they used to be. That is, big in size, complex, distributed, etc. This all has triggered the emergence of the *Big Data* as a new discipline.

These aspects are well reflected in the contributions in this volume. Basically, they all are concerned with innovative tools and techniques for broadly perceived data mining and knowledge discovery that would be effective and efficient for solving problems a big data context as explained above. The authors of the chapters address subjects ranging from mining data from opinion, spatiotemporal databases, discriminative subgraph patterns, path knowledge discovery, social media, and privacy issues. Therefore, the contributions cover a comprehensive set of areas that are relevant, and in which the big data aspect is relevant.

We wish to congratulate Professor Chu for his vision to notice that the time is right for such a relevant volume, and for his excellent job to select relevant and challenging topics and bring together prominent contributors. The contributors should be greatly appreciated for their papers which provide both a coverage of the area and a presentation of their new results.

We sincerely hope that this great volume will be a very good start of the new *Studies in Big Data* book series, and that the series will develop rapidly in line with our other big and highly successful book series at Springer.

August 2013 Thomas Ditzinger
Heidelberg and Warsaw Janusz Kacprzyk

The original online version for this volume can be found at
http://dx.doi.org/10.1007/978-3-642-40837-3

Author Index