Towards a Framework for Learning from Networked Data

Jan Ramon

Department of Computer Science, KU Leuven
Celestijnenlaan 200A, 3001 Heverlee, Belgium
Jan.Ramon@cs.kuleuven.be

1 Introduction

Over the past decades, one has seen databases of ever increasing size and complexity. While the increasing size is easy to measure in bytes, kilobytes or terabytes, the increase in complexity is more difficult to quantify, however, it has a very deep effect on the theory we use to reason about the data. While in earlier days many researchers reasoned in terms of sets of similarly structured and independent objects, today we are facing large networks of data where everything is connected directly or indirectly to everything else. Examples include social networks, traffic networks, biological networks, administrative networks and economic networks.

These developments have spurred a renewed interest in data storage and knowledge extraction (answers to queries, patterns, models, ...). Three key underlying challenges are the representation of the data and knowledge, managing the computational cost of the problems which we need to solve and the statistical challenge related to the complexity of the data.

In this contribution, I will survey these challenges from a data mining point of view. I will argue that in order to address the current challenges it is valuable to gain a better understanding of fundamental statistical and algorithmic properties of large data networks and to integrate ideas from the many fields of research that are concerned with such networks.

2 Knowledge Representation

More than 15 years ago, many datasets were transactional. They consisted of a set of independent and separate transactions. Examples are a database of transactions describing what an anonymous customer bought in a shop, a database of molecules available to some chemical company, a set of responses to a survey, a database of patient records of a medical doctor, etc.

However, in reality pieces of information are rarely independent. Even for the classical example of transactions containing items bought together in a shop, transactions are related because they featured the same customer, the same shop staff member, the same calendar day, the same discount offer or other more subtle relationships which may have let one transaction influence the behavior

N. Hernandez et al. (Eds.): ICCS 2014, LNAI 8577, pp. 25–30, 2014.
DOI: 10.1007/978-3-319-08389-6_3,

of a customer in another transaction. To the other extreme, in many current big databases including social networks, economic networks, biological regulatory networks and traffic networks the relationships are considered the most important part of the information. This shift in the complexity of data has two major implications on the level of knowledge representation, in particular on representing data and representing dependencies.

Representing Data. In the early nineties, one became aware that sets of vectors (the so-called propositional or attribute-value representation) was not anymore sufficient to represent datasets. The field of inductive logic programming (ILP) [11] was the first to be successful in machine learning describing data with the more powerful language of first order logic. In fact, first order logic is so powerful that many problems including deduction are undecidable, and to mitigate the computational intractability, several settings between the propositional one and the first order logic one were explored [4]. After a while, a lot of researchers shifted to graphs to represent data, as they hit a nice balance between expressivity (being as powerful as relational databases or datalog) and computational tractability (in the sense that most things are decidable and the computational complexity of many tasks have been studied in the field of algorithmic graph theory). While transactional graph mining is closely related to the learning from interpretations setting in ILP, the learning from entailment setting was the first one to see instances as elements of a large connected knowledgebase [3]. When the graph mining community moved away from transactional datasets, this idea of a global knowledgebase was revisited under the name of network analysis, a term which is also used in the branch of statistical physics studying graphs representing complex systems [10]. In this representation framework, networks are graphs whose vertices represent objects (or parts thereof) and edges represent relations between them. Depending on what is most convenient, often for theoretical purposes more simpler and abstract settings and for practical applications richer settings, one can use directed or undirected graphs, graphs or hypergraphs, labeled or unlabeled graphs, but usually there are straightforward transformations from the one type to the other type of graph (as illustrated e.g. in [2]).

Representing Dependencies. A second implication of recognizing that instances are part of a single large world is of a more statistical nature. Indeed, instances sharing relations to the same objects in the world may not be independent from a statistical point of view. For instance, friends may share interests, well connected cities may share economic activity and interacting molecules may participate in a shared biological process. In many statistical approaches it is important to represent these dependencies, and the field of statistical relational learning (SRL) [6] has investigated many ways to extend probabilistic models to graph-structured databases. Such models represent explicitly dependence or independence relationships between variables using (hyper)edges, such that again a graph is obtained. Most SRL approaches somehow assume that one can model these statistical dependencies, or at least learn them in a reasonable way from

data. This holds in a number of applications, to a large extent also the one discussed below, but as Section 4 will argue sometimes things are more difficult.

A Case Study in Experimental Research. Experimental research in the field of computational biology is a typical domain where the knowledge base integrating domain knowledge, experimental setups and experimental results may get very complex. Here, only a simplified illustration from the domain of protein mass spectrometry [9] is provided.

Mass spectrometry is a technique to detect what molecules are present in a sample (e.g. a blood sample), in the case of protein mass spectrometry one aims at detecting the proteins in such sample. A typical experimental setup involve a pipeline of several treatments (e.g. a typical sequence is digestion, chromatography, ionization, fragmentation, detection). The way each step in such pipeline transforms the sample depends on the characteristics on the instrument and its parameter settings. Recently, there is a growing interest in modeling more precisely each of these transformations (see [5] for an illustration on the digest step). The more accurate are such models, the more accurately one can reason about what was in the sample at the beginning based on the output of the detection step at the end of the pipeline.

In experimental biomedical research, one often uses mass spectrometry to detect whether a particular protein is present in a sample or not. However, the results of such experiments are not independent. Several proteins may be part of a common pathway (chain of chemical reactions in the cell), and hence detecting one protein may be correlated with detecting another protein. The graph representing the interactions between proteins is called the regulatory network. The closer the regulatory network relates two observed proteins, the less statistical evidence it provides. The more unrelated the proteins are, the more we can see them as independent evidence / indications of a particular phenomenon of interest (e.g. a disease we want to diagnose). The better our knowledge of this regulatory network is, the better we can assess how much independent evidence a set of observations provide.

3 The Question of Computational Tractability

A second implication of the increasing complexity of available data is that many tasks which were almost trivial for transactional databases get intractable for networked data. One prototypical example is the problem of pattern matching. The pattern matching operator which is most widely studied in the field of graph mining is subgraph isomorphism. Unfortunately, it is an NP-complete problem to decide whether a pattern is subgraph isomorphic to a database graph. For transactional graph databases this was not very problematic as transactions are usually limited in size. Moreover, for many specific applications, e.g., molecule databases, optimized pattern mining solutions have been developed [7] exploiting the structure of the database graphs. Unfortunately, large data networks don't have an easy to exploit structure and are at the same time orders of magnitude

larger. The NP-hardness of the problem suggests that the increase in computational complexity caused by the increasing database sizes is not expected to be compensated by the increasing computing power predicted by the law of Moore.

We therefore face the fundamental problem of making data analysis algorithms scale well with the growing databases. Fortunately, a lot of useful inspiration is provided by recently emerged research lines in theoretical computer science. In particular, despite the fact that the work in theoretical computer science is not necessarily intended for immediate applicability, we recently demonstrated that the above mentioned pattern matching problem can be addressed to a large extent by the use of fixed parameter tractable algorithms [8].

4 Towards a Statistical Framework

A third challenge raised by considering data in networks is of a statistical nature. As explained above, a first step is to explicitly model the dependencies between variables. However, one can argue that even when these dependencies are modeled, the problem of learning a predictive model is still not completely well defined, as we didn't specify complete statistical assumptions.

To see this, let us first recall the most common statistical assumption. In classical statistics, when learning a model on training data and then performing predictions on unseen data, one usually makes the assumption that both the seen and the unseen instances are drawn (independently) from a fixed (but unknown) distribution. If one rolls a dice 1000 times and gets a 2 in 50 of these 1000 cases, then when rolling the dice again, one expects that the result will be a 2 with probability 0.05 because it concerns the same (clearly biased) dice. If the dice would have been replaced by another one, there is no reason to have the same belief.

A similar mechanism is needed for network statistics. However, it is unclear what it means "to be drawn from the same distribution". As an example, consider the following fictitious world. Suppose that 90% of computer users use an operating system called windows and like sunny weather over rainy weather, while 10% of computer users use an operating system called linux and they are nerds not caring about the weather because they are always inside. Designers of operating systems randomly choose some delimiter to separate directory names in paths. linux designers preferred forward slashes while the CEO of the company making windows preferred backslashes. Moreover, assume the CEO of the company making the windows system likes sunny weather. Now, if we take a random computer user, there is a 90% probability that he uses backslashes in pathnames and it is equally probable he likes sunny weather. Now suppose that a tiny change happens: the CEO of the company making windows is replaced by a nerd not interested in the weather, and he prefers to use hash signs as delimiter. Note that everything is still drawn from exactly the same distribution. Now, if we take a random user, will he like sunny weather and/or will he use backslashes? The users using windows didn't change, so they still like sunny weather. However, they are forced now to use hash signs rather than backslashes.

From this example we can see that in situations which are structurally identical from the point of view of data, the result can be quite different due to a difference in the underlying process. In this specific case, we can't distinguish between variables which are functions of the features of individuals and variables which are functions of their connectivity. Of course, this problem would not arise when there would be a large number of providers of operating systems. Unfortunately in the real world many networks have been shown to follow a powerlaw distribution [1], implying there are often a few dominating "hubs" with an exceptionally high connectivity.

Essential to any solution to this problem is to perform a more systematic analysis of learning theory and to specify clearly under which assumptions some prediction (generalization) effort is valid. For instance, in [12] we showed preliminary results providing learning guarantees under the assumption that the connectivity of the objects involved in an observation and the function mapping the features of these objects on the target value are independent.

5 Conclusions

In this contribution, I argued that due to the increasing amount and more importantly the increasing complexity of data, in order to store, process and analyze data we face challenges on the level of knowledge representation, computational costs and statistical inference. Over the past few years, several ideas to address these challenges have been developed, but several open problems remain.

Of particular interest is the statistical challenge. In the past, the majority of efforts were directed at extending data mining algorithms towards graph-based representations and making them computationally feasible. but less attention has been payed to developing a consistent theory of statistics on graphs. It seems plausible that integrating ideas from statistical relational learning theory and random graph theory can further progress the field.

Acknowledgements. This research has been supported by the European Research Council, grant ERC-StG 240186 "MiGraNT: Mining Graphs and Networks, a Theory-based approach".

References

1. Barabási, A.L.: Scale-free networks: A decade and beyond. Science 325(5939), 412–413 (2009)
2. Calders, T., Ramon, J., Van Dyck, D.: All normalized anti-monotonic overlap graph measures are bounded. Data Mining and Knowledge Discovery 23, 503–548 (2011)
3. De Raedt, L.: Logical settings for concept learning. Artificial Intelligence 95, 187–201 (1997)
4. De Raedt, L.: Attribute-value learning versus inductive logic programming: The missing links (extended abstract). In: Page, D. (ed.) ILP 1998. LNCS (LNAI), vol. 1446, pp. 1–8. Springer, Heidelberg (1998)

5. Fannes, T., Vandermarliere, E., Schietgat, L., Degroeve, S., Martens, L., Ramon, J.: Predicting tryptic cleavage from proteomics data using decision tree ensembles. Journal of Proteome Research 12, 2253–2259 (2013)
6. Getoor, L., Taskar, B.: An Introduction to Statistical Relational Learning. MIT Press (2007)
7. Horváth, T., Ramon, J., Wrobel, S.: Frequent subgraph mining in outerplanar graphs. Knowledge Discovery and Data Mining 21(3), 472–508 (2010)
8. Kibriya, A., Ramon, J.: Nearly exact mining of frequent trees in large networks. Data Mining and Knowledge Discovery 27, 478–504 (2013)
9. Martens, L., Laukens, K., Ramon, J., Valkenborg, D.: Inspector: An integrated informatics platform for mass-spectrometry protein assays
10. Newman, M.: Networks: An introduction. Oxford University Press (2010)
11. Nienhuys-Cheng, S.-H., de Wolf, R.: Foundations of Inductive Logic Programming. LNCS (LNAI), vol. 1228. Springer, Heidelberg (1997)
12. Wang, Y., Ramon, J., Guo, Z.-C.: Learning from networked examples in a k-partite graph. In: Proceedings of la Confrence sur l'Apprentissage Automatique, Lille, France, pp. 1–8 (July 2013)