Soft Comput (2016) 20:2329–2339
DOI 10.1007/s00500-015-1643-3

METHODOLOGIES AND APPLICATION

Automatic constraints generation for semisupervised clustering: experiences with documents classification

Irene Diaz-Valenzuela · Vincenzo Loia · Maria J. Martin-Bautista · Sabrina Senatore · M. Amparo Vila

Published online: 17 March 2015

Abstract In the last times, semi-supervised clustering has been an area that has received a lot of attention. It is distinguished from more traditional unsupervised approaches on the use of a small amount of supervision to "steer" clustering. Unfortunately in the real world, the supervision is not always available: data to process are often too large and so the cost (in terms of time and human resources) for user-provided information is not conceivable. To address this issue, this work presents an automatic generation of the supervision, by the analysis of the data structure itself. This analysis is performed using a partitional clustering algorithm that discovers relationships between pairs of instances that may be used as a semi-supervision in the clustering process. The methodology has been studied in the document clustering domain, an area where novel approaches for accurate documents classifications are strongly required. Experimental result shows the validity of this approach.

Communicated by V. Loia.

I. Diaz-Valenzuela · M. J. Martin-Bautista · M. A. Vila
Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain
e-mail: idiazval@decsai.ugr.es

M. J. Martin-Bautista
e-mail: mbautis@decsai.ugr.es

M. A. Vila
e-mail: vila@decsai.ugr.es

V. Loia · S. Senatore (✉)
Dipartimento di Informatica, Università degli Studi di Salerno, 84084 Fisciano, SA, Italy
e-mail: ssenatore@unisa.it

V. Loia
e-mail: loia@unisa.it

1 Introduction

Semi-supervised clustering has become a topic of great interest to data mining and machine learning communities. It improves classic unsupervised approaches by including side information which helps the clustering process to find a better solution. This supervision, coming from external or expert knowledge, could be given in two different ways: as a small amount of labeled instances or as pairwise instance level constraints. A good semisupervision guides the clustering algorithm into an adequate partitioning of the data and, often, improves the clustering performance significantly.

Existing methods for semi-supervised clustering fall into three categories: constraint-based, distance-based and hybrid. The first category uses user-provided labels or constraints that are enclosed in the algorithm. They modify the objective function to include the information from the pairwise instance constraints (Wagstaff et al. 2001) or to generate seed clusters using the labeled data (Basu et al. 2002).

In distance-based approaches, metric learning techniques define adaptive distance measures that are used for training the clustering to satisfy the labels or constraints in the supervised data (Xing et al. 2002). Finally, the hybrid methods propose some unification of the first two previous approaches. For instance, in (Basu et al 2004) a general probabilistic framework unifies both ideas and in (Li et al. 2010) provide a complete experimentation, testing the performance of an hybrid proposal in contrast with the use of labels and constraints separately.

The most popular type of supervision used in clustering algorithms is pairwise feedback. It indicates whether two points belong to same cluster or to different clusters. Pairwise relations arise naturally from knowledge in many domains, such as gene classification (two co-occurring proteins), Information Retrieval (documents regarding the same topic), etc.

The metrics used to describe those pairwise relations come from the knowledge of domain experts, generally as similarity (or dissimilarity) measures. Moreover, these pairwise relations are often determined in a subjective way.

Particularly, in the Information Retrieval domain, using semi-supervised clustering with human expertise could be very helpful in comparison with traditional supervised approaches. To support the natural grouping of documents, they can "inject" a certain amount of external information in the process. Nevertheless, the role of the expert in semi-supervised clustering is still costly and time consuming. Finding pairwise relations in a document corpus containing a few thousand of documents would require to read all of them carefully, which could be a tedious and arduous task.

To address this issue, this paper introduces a methodology aimed at the automatic generation of pairwise feedback as instance-level constraints. To accomplish this task, the inherent structure of the data is studied by means of a partitional clustering algorithm. From the output of this initial clustering, it is possible to establish relationships between pairs of elements that led to instance level constraints. On a second stage, these constraints are used as semi-supervision in a hierarchical clustering algorithm.

The paper is organized as follows: Sect. 2 sketches a literature review of some related works. Section 3 introduces the theoretical bases of the methodology. After that, in Sect. 4, the whole process for the constraint generation and the document clustering is described in detail. Section 5 covers experimental results followed by some conclusions and future works in Sect. 6.

2 Related work

Semisupervised clustering appeared as an alternative to traditional unsupervised approaches where a small quantity of side information is introduced in the clustering process to improve its performance.

Some good reviews about this techniques can be found in Grira et al. (2004) and Basu et al. (2008). The paper focuses on a type of semisupervision that come from *pairwise instance-level* constraints. They were first introduced by Wagstaff et al. (2000) and they have been widely used and reformulated since then Xiong et al. (2014) and Tang et al. (2007). Other approaches like Loia et al. (2003) and Pedrycz et al. (2010) have used the concept of pairwise external information in combination with fuzzy clustering.

Document clustering is a well-known research domain that has been studied from very different perspectives (Aggarwal and Zhai 2012). Numerous studies in the Information Retrieval domain have evidenced that document clustering represents a good way of organizing the retrieval results (Leuski 2001), browsing collection of documents [also through clusters hierarchy (Sahoo et al. 2006)] and matching the user's query (Cutting et al. 1992). Particularly, the use of semisupervised document clustering has been previously studied in Basu et al. (2002), Rigutini and Maggini (2005), Zhao et al. (2012) and Hu et al. (2012) among others.

Traditionally, *instance-level* constraints have been generated by a human expert with some knowledge about the specific topic under consideration. Some approaches like (Zhao et al. 2012; Xiong et al. 2014; Barr et al. 2014) basically exploit that assumptions, and automatically select which kind of instances could provide a specific important information for the clustering process and ask the expert about them. In that way, the role of the expert has been always taken into account. Our proposal differs from others in the sense that the human expertise has been removed from the process.

Diaz-Valenzuela et al. (2013, 2014), a hierarchical version of semisupervised clustering has been introduced. Our method is based on this approach, by adding the automatic supervision. Particularly, the approach has not been applied to document clustering before. We believe that its application in that domain can provide a positive insight as it can take advantage of the characteristics of this kind of clustering. As hierarchical clustering does not return a partition with a fixed number of groups, it is possible to get a more specific solution that other kind of methods could not offer.

3 Methodology

The goal of this paper is to generate pairwise constraints that could be used in a semi-supervised clustering algorithm. Using this kind of methods, it is our intention to find a more accurate clustering assignment than the one from unsupervised algorithms. As stated in the introduction, these pairwise constraints are automatically generated by means of a partitional clustering algorithm, without external or human-provided expertise (except for its initialization parameters). Specific details about that process are provided in Sect. 4.2.1. Our proposal has been tailored for the document classification domain, where the massive volume of text from Web, e-mails, feeds, blogs, etc. requires enhanced approaches to automatize the time-consuming labeling process.

3.1 Problem formulation

This proposal is described in the document clustering and classification context. Under this problem, let us assume that a set of data instances, precisely a set of documents, is given (and in case, the corresponding class labels for validation). Document clustering can be defined as the process followed to find the best partition in the document corpus according to a certain criteria. Formally, let us define:

Definition 1 Let $D = \{d_1, \ldots, d_m\}$ be a documents collection, $T = \{t_1, \ldots, t_l\}$ a set of terms from the documents set, then $P = \{C_1, \ldots, C_n\}$ is a partition of the document corpus, where each C_i ($i = 1 \ldots, n$) is a subset of documents, s.t. $\cup_{i=1}^{n} C_i = D$ and $\cap_{i=1}^{n} C_i = \emptyset$.

In our approach, the partitioning of the input document set is obtained by means of semi-supervised hierarchical clustering technique. The main advantage of these techniques is the possibility of obtaining the optimal number of groups in data. In the document clustering context, the hierarchical clustering allows to find a more specific partitioning than partitional clustering, which needs an a priori fixed number of clusters. Moreover, its inherent nature provides an extensive hierarchy of clusters that better fits the representation of subtopics for a more detailed document classification.

Our methodology, based on the method described in (Diaz-Valenzuela et al. 2013), inserts some supervision (as side information) in the hierarchical clustering process to find the best partition of the input data. It applies the semi-supervision using *Instance level constraints* in the sense of (Wagstaff and Cardie 2000). These constraints are based on two types of information: *must-link* and *cannot-link*. The first type describes the relationship between two documents in the same cluster, whilst the latter indicates that the documents are in different clusters. Formally:

Definition 2 (*MUST-LINK*): Given two documents d_i and $d_j \in D$, there is a must-link $ML(d_i, d_j)$, if d_i and d_j are in the same cluster. The set of all *must-link* constraints defined for D is called ML.

Definition 3 (*CANNOT-LINK*): Given d_i and $d_j \in D$, if there is a cannot-link $CL(d_i, d_j)$, then d_i and d_j cannot not be in the same cluster. The set of all *cannot-link* constraints defined for D is called CL.

Moreover, we assume that there is an underlying class structure that assigns each document to one of c classes C. The aim of this method is to find a mapping F between the calculated partition and the given classification, such that $F : P \rightarrow C$.

4 Semisupervised document clustering: the main steps

Figure 1 shows the global process that has been followed to achieve the document clustering. The collection of documents is given as an input to the *Preprocessing* step. This phase cleans the data, discarding "noisy" words, i.e., those words that are very infrequent or uninformative and do not provide important information for the clustering process. Once the text has been refined, it is translated into a document-term matrix used as input to a *Semisupervised Clustering* step. Pairwise constraints are generated as side information and used as input to the hierarchical clustering algorithm, producing the final partition P. Automatically generated constraints are used instead of expert provided information. Next subsections provide details about the data flow through the introduced steps.

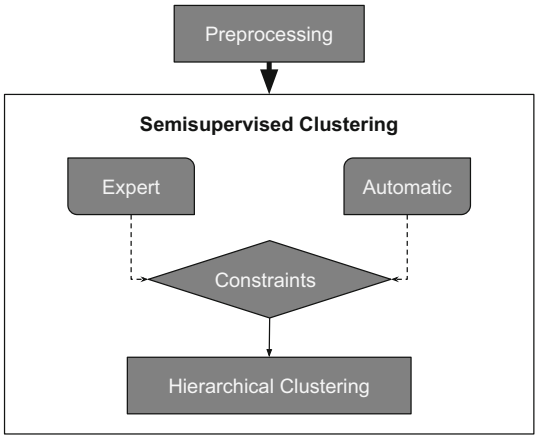


Fig. 1 Summary of the methodology

4.1 Preprocessing

Text preprocessing is an important activity in the context of Information Retrieval. It cleans textual information to enhance the quality of data and, at the same time, to make further processing easier by removing non-essential information.

The document collection must be generally converted into a digest representation, according to some mining process. In our approach, the collection is transformed into a document-term matrix, where each document is represented using the Vector Space model. Typical text preprocessing tasks are sketched as follows:

- *Tokenization* The aim of this task is the exploration of the words in a sentence. Initially, textual data are seen as a block of characters. This task replaces sequences of characters with sensitive data, i.e., identifies meaningful words.
- *Stop-word removal* Removes common words which do not provide meaning of the documents, such as prepositions, articles, and pro-nouns, etc. Those words are treated as stopwords and so removed.
- *Stemming or lemmatization* Reduces the words into their root. Many words in the English language can be reduced to their base form or stem, e.g., *improve, improvement, improved*, etc., belong to *improve*.

After preprocessing, each document d_i is represented as a term vector under the Vector Space Model, where every

component is the weight w_{ij} associated with a t_j in that document. Values for w_{ij} must calculated using appropriate term measures, like TF, TF-IdF, among others. The specific measure should be chosen according to the specific nature of the dataset and may be determined experimentally.

Document-term matrices are generally very sparse, negatively affecting both the computational performance and the quality of the clustering process itself. The preprocessing step may contribute to reduce that sparsity and dimensionality, even though dimensionality reduction is often dataset dependent.

4.2 Clustering

After the preprocessing (topmost part of Fig. 1), the documents' corpus is given as an input of the *Semi-supervised Clustering* as a document-term matrix.

The groups of document underlying the document clustering matrix are found by means of a hierarchical clustering algorithm with side information. Hierarchical clustering produces hierarchies, viz., structures that are often more informative than the unstructured set of clusters returned by flat clustering. Moreover, it does not need to specify the number of clusters a priori, because it finds the natural number of partitions on the data. Consequently, they often get a more accurate partition of the data (in this case, of the documents corpus).

As stated, this proposal is based on the algorithm introduced by Diaz-Valenzuela et al. (2013), where side information is used to find the best partition of the dendrogram. This side information is provided as *instance level constraints* defined between pairs of documents. They indicate whether two documents are similar or not, i.e., if they are (*must-link*) or not (*cannot-link*) in the same cluster.

Constraints determine the final partition of the dendrogram and allow the introduction of some external observations about data that can condition its structure. They can be obtained in several ways, from using an expert to defining them according to class labels. For this proposal, instance level constraints are generated automatically.

4.2.1 Constraints generation

The human expertise is preferable when supervision is recommended, although user-provided suggestions or hints are often expensive and time consuming to obtain. However, in some approaches, human intervention can be replaced by automatically generated knowledge. The study of the data and their placement in the n-dimensional space evidences some structural relationship that can be a valid support for driving in the constraints attribution.

In this approach, constraints are generated by the study of the inherent nature of the data. It is done using a partitional clustering process that obtains a partition of the data according to some distance criteria. Specifically, our method uses k-means (Jain and Dubes 1988), a well-known flat clustering that finds a partition $P_K = \{S_1, \ldots, S_k\}$ for a given k by minimizing the within-cluster sum of squares, according to the following objective function:

$$J = \sum_{i=1}^{k} \sum_{j=1}^{n} \|x_{i,j} - \mu_i\|^2 \tag{1}$$

where $\|x_{i,j} - \mu_i\|$ is the distance between a data point $x_{i,j}$ and the centroid μ_i of the cluster S_i.

The partition P_K gives an approximate idea about how the input data is organized. Regardless the possible mistakes and inaccuracies that P_K could present, it is possible to use this information to generate constraints. In the context of our document corpus, each $S_a \in P_K = \{d_{k_1}, \ldots, d_{k_t}\}$ is a partition containing similar documents. If we consider that all documents from that partition should be in the same group, must-link constraints are pretty straightforward.

More formally, for each pair of documents (d_i, d_j) that are in the same cluster in the clustering-driven partitioning, there exists a *must-link* constraint $ML(d_i, d_j), \forall (d_i, d_j) \in S_a | S_a \in P_K$.

Under this assertion, the set of must-link constraints, ML, contains all pairs of elements that are in the same cluster (considering all clusters independently): $ML = \cup_{i=1}^{k} ML_i$, where ML_i is a set of must-link constraints from a partition S_i.

Similarly, *cannot-link* constraints are defined between pairs of documents (d_i, d_j) that are in different clusters, i.e., $\forall (d_i, d_j); d_i \in S_a, d_j \in S_b | S_a, S_b \in P_K; S_a \neq S_b$. Under this definition, the set of cannot-link constraints, CL ,contains all pairs of elements that are in the different clusters (considering all clusters independently): $CL = \cup_{\forall a,b:a \neq b} CL_{a,b}$, where $CL_{a,b}$ is a set of cannot-link constraints composed of the pair $(d_i, d_j) \in S_a \times S_b | a \neq b$.

This approach is sensible to the initial configuration of the *k-means* algorithm. However, this can be overtaken by exploiting the random component of the k-means initialization. The partition returned by k-means depends on an initial centroid, μ_i that normally is randomly generated on each execution of the algorithm. It means that every execution may provide slightly different partitions. Under that assumption, by executing the algorithm repeatedly, the original set of constraints is refined defining the constraints. In this sense, if two documents d_i and d_j are placed in the same cluster in all executions of the k-means clustering, then there is a *must-link*, $ML(d_i, d_j)$, constraint between them. In the same way, if two documents d_i and d_j are never placed in the same cluster in all executions, then there is a *cannot-link* constraint $CL(d_i, d_j)$. This process has been summarized in Algorithm 1

Algorithm 1 Constraints generation

```
Get P_K an initial partition returned by k-means
for all K_i ∈ P_K do
    if (d_i, d_j) ∈ K_i then
        Add (d_i, d_j) to ML
    else
        Add (d_i, d_j) to CL
    end if
end for
for each k-means execution do //
    Get P_a = {S_1, ..., S_k} the partition returned by k-means
    for all ML(d_i, d_j) ∈ ML do
        if d_i ∈ S_a and d_j ∈ S_b with S_a ≠ S_b then
            Remove (d_i, d_j) from ML
        end if
    end for
    for all CL(d_i, d_j) ∈ CL do
        if d_i, d_j ∈ S_a then
            Remove (d_i, d_j) from CL
        end if
    end for
end for
```

Considering that we are keeping only that information coherent in all the executions of the k-means clustering algorithm, there are some pairs of documents without an associated constraint. This is because they are placed in the same cluster in some executions and in different clusters in others. Moreover, data structure can affect the performance of k-means clustering, which generally tends to produce clusters of relatively uniform size. In case of bad partitioning, some of the resulting clusters could not be used for the constraint generation, because their data relations are considered not good enough for the constraints (for example if they are too big, compared with the remaining clusters). In that case, the constraints are generated considering only the clusters that fit some criteria, discarding all the remaining information.

4.2.2 Hierarchical clustering

Once the instance level constraints have been generated, they will be used to add some "supervision" to the hierarchical clustering. It takes the document-term matrix and produces the final partition of the document corpus, in the form of a dendrogram, i.e., a tree of nested partitions, $E = \{P_{\alpha_1}, \ldots, P_{\alpha_n}\}$. By cutting this tree at a specific point, α_r, every branch becomes a different cluster $P_{\alpha_r} \forall r \in [1, \ldots, n]$. Obviously, different cuts of the dendrogram produce different partitions.

Diaz-Valenzuela et al. (2013), an approach to obtain the optimal partition of a dendrogram using *instance level constraints* is presented. It compares all partitions from the dendrogram E, with some subsets of both ML and CL constraints finding the partition that better fits them. The best partition P_{α_r} must satisfy more constraints according to a score s_α.

Definition 4 For each partition P_{α_r}, there is a score s_{α_r} indicating how this specific partition fits the constraints, s.t. $s_{\alpha_r} = \sum_{i,j=1}^{m} v_{ij}$ where v_{ij} is defined by (2).

$$v_{ij} = \begin{cases} v_m, & \text{if } ML(i,j) \text{ is satisfied;} \\ v_h, & \text{if } CL(i,j) \text{ is satisfied;} \\ v_{nm}, & \text{if } ML(i,j) \text{ is not satisfied;} \\ v_{nh}, & \text{if } CL(i,j) \text{ is not satisfied;} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

weight values v_m, v_h, v_{nm} and v_{nh} can be determined according to the characteristics of each problem.

Definition 5 (*Constraints satisfaction*) Given two documents d_i and d_j, a must-link constraint $ML(d_i, d_j)$ is said to be *satisfied* if the documents d_i and d_j are in the same cluster in a specific partition. On the other hand, there is a cannot-link constraint $CL(d_i, d_j)$ that is *satisfied* if documents d_i and d_j are in different clusters.

Definition 6 The best partition of the data, P_{α_r}, is the one that maximizes s_{α_r}; $P_{\alpha_r} = \max_r s_{\alpha_r}$.

To find the optimal partition of the dendrogram, this method focuses on the overall performance of all constraints and, consequently, is not sensitive to their possible mistakes. This fact makes this algorithm a good candidate to use with automatically generated constraints. We can suppose indeed that they may have some level of noise or mistakes, so it is important that the semi-supervised clustering algorithm is able to overcome such problems.

5 Experimental results

Since the novelty of our method is the automatic constraints generation, after introducing the overall methodology, in this section, we focus on the performance of the method for generating instance level constraints. Let us recall that the instance level constraints are obtained automatically by the methodology described in Sect. 4.2.1. This study aims to evaluate this methodology and constraints performance by analyzing:

- How the k-means configuration affects the quality of the constraints;
- How the automatically generated constraints using k-means perform in comparison with other methods to generate constraints;
- How the semi-supervised approach performs in comparison with unsupervised methods.

5.1 Data description

Two datasets have been used to validate our methodology. They have gone through the preprocessing process outlined in Sect. 4.1, where some cleaning and dimensionality reduction

tasks have been carried out, specifically tailored with respect to the nature of each dataset:

- *Web Snippets* dataset (Phan et al. 2008) Contains 2280 short texts taken from Google, unevenly divided into eight categories. Each snippet contains from 6 to 20 words approximately. During the preprocessing step, it has been prepared by removing the first words of each snippet, as they seem to be part of the URL of the original document (information that it is not useful for categorization). Additionally, a stemming process has been applied and its stop words have been removed. A dimensionality reduction has been performed by removing those words with a correlation higher than 0.5 using Pearson's method. During the constraint generation process, when executing the k-means algorithm, these clusters with sizes bigger than 25% of the whole dataset size have been ignored. Binary Term Frequency has been used as weighting measure for the terms of this dataset.
- *Reuters-21578* collection It is a subset of the well-known Reuters-21578 collection containing 2014 documents. Ten independent categories have been selected from the dataset: *trade*, *ship*, *wheat-grain*, *gold*, *sugar*, *money-fx*, *interest*, *crude*, *money-supply* and *coffee* with different sizes. This dataset has been preprocessed using a lowercase representation, removing numbers, punctuation and stop words. It has also gone through a stemming process. Dimensionality reduction has been performed using the 500 terms with higher TF-IDF value. As with the previous dataset, during the constraints' generation process, these clusters containing more than 25% of documents have not been considered. TF-IdF has been used as weighting measure for the terms of this dataset.

For each dataset, constraints have been generated by 30 executions of the k-means algorithm. The semisupervised clustering algorithm allowed different hierarchical clustering methods. For these data, we have used hierarchical agglomerative clustering algorithm with the Ward's method. The parameters' setting for the formula in 4 that calculates the best partition according to the score s_α is set to consider only satisfied constraints, so $v_{nm} = v_{nh} = 0$. In addition to that, weights v_m, associated with the importance of the must-link constraint and v_h, associated with the cannot-link, take the value $v_m = 2$, $v_h = 1$ to reinforce the information provided for the must-link in contrast with the cannot-link. The reason to do that is because the automatic generation method, by definition, provides more *must-link* than *cannot-link* constraints. Using this setup, we obtain a more balanced contribution by the two types of constraints.

5.2 Validity measures

To validate the goodness of the clustering process, two different measures have been used: F-measure and Normalized Mutual Information (briefly, NMI). F-measure evaluates the quality of the clusters by comparing the relationship between the retrieved documents on each cluster and the relevant documents according to their given class labels. F-measure (5) is defined as the harmonic mean of Precision (3) and Recall (4).

$$\text{Precision} = \frac{|\text{relevant documents} \cap \text{retrieved documents}|}{|\text{retrieved documents}|} \tag{3}$$

$$\text{Recall} = \frac{|\text{relevant documents} \cap \text{retrieved documents}|}{|\text{relevant documents}|} \tag{4}$$

$$F - \text{measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{5}$$

The Normalized Mutual Information (6) evaluates the elements on each cluster against class labels. It measures and normalizes the mutual information between random variables P_α (the optimal partition of the dendrogram) and C (the ground truth given by class labels).

$$\text{NMI}(P_\alpha, C) = \frac{2I(P_\alpha, C)}{H(P_\alpha) + H(C)} \tag{6}$$

where $I(P_\alpha, C)$ is the mutual information between the two random variables and H is the Shannon entropy of the variable.

Sizes of constraints sets ML and CL generated by this method are different for each dataset. Considering that the number of constraints introduced on the process of finding P_α affects its performance, different random subsets are taken from ML and CL of proportional size to the original set.

To guarantee a complete analysis, different constraints' sets of different sizes have been used, incrementally from 0.1 to 90 % of each set. As these subsets of constraints are chosen randomly, each experiment has been executed 5 times using different random subsets of constraint. Thus, the values of F-measure and NMI on the next graphs are obtained as the average of these 5 executions.

5.3 Effect of k-means configuration on the constraints

As stated, the constraint generation process is based on repeated executions of the k-means algorithm. This algorithm requires a parameter, k, representing the number of groups (clusters) in which the input data are split. The influence of k has been studied by generating several constraints' sets with different values of k. Indeed, for each dataset, we have considered a wide range of k values that go from half of expected groups according to class labels, to the effective expected groups (according to class labels), up to a maximum of $k = 30$ (see Fig. 2). This range of values guarantees a complete view of the behaviour of our methodology: first considering a k smaller than the expected clusters' sizes and then assessing how the increase of k impacts on the results.

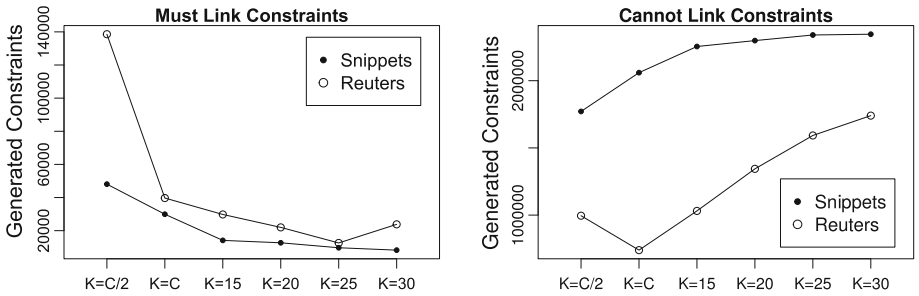


Fig. 2 Influence of the parameter k in the number of constraints

(a) Number of Must Link Constraints **(b)** Number of Cannot Link Constraints

Figure 2 shows the dependency between k (axe x) and the number of generated constraints (axe y). As it is possible to see in the Figure, k affects the size of the constraint sets. Let us notice that there are more *cannot-link* (Fig. 2b) than *must-link* constraints (Fig. 2a). This is due to the number of classes in the dataset: since a must-link constraint is defined between elements in the same class whilst a cannot-link is defined between elements in different classes. It results in more cannot-link constraints than must-link. This is because by definition: when the number of classes is big, an element has more possibilities of being on a cannot-link relationship than on a must-link. Then, cannot-link relationships are defined between every element of a cluster and all elements from the remaining clusters; instead must-link are defined only between elements within their own cluster.

Figure 2a shows how the must-link constraints change as k increases. It is interesting to check how as the number of must-link constraints decrease, the number of cannot-link constraints increase (Fig. 2b). This is related with the number of clusters and their sizes. A bigger value of k means smaller clusters, resulting in few possibilities for the elements to be part of a must-link relationships, as there are few elements inside the cluster. On the other side, there are more clusters, (i.e., many subgroups of documents), so there are more conceivable combinations for the cannot-link information. Similarly, when k is really small, there are more (must-links and cannot-links) constraints, but they are not of good quality, as information is mixed up. This can be seen in Figs. 3 and 4, where the performance of the methodology for different values of k is shown.

Figure 3 shows the F-measure for all considered datasets. Let us notice that using a small k, the performance of our approach is poor, especially when the subset of constraints used is small. This can be seen more clearly for Reuters dataset (Fig. 3b), where it needs up to 60 % of the constraints to obtain an F-measure equal to 0.8. Figure 2a shows that the poor performance is related with the higher number of must-link constraints, coming from fewer clusters composed of mixed data. This behaviour is also observable in Fig. 4 where the Normalized Mutual Information is measured for all analyzed datasets.

In conclusion, from the analysis of the figures, let us assert that the results tend to stabilize when the value of k is close to the number of classes or bigger. Specifically, in the Reuters dataset (Figs. 3b, 4b), it is possible to observe a small improvement using K bigger than 15.

5.4 Comparison with other clustering approaches

Instance level constraints are traditionally provided by an human expert with some knowledge about the specific data. In Tang et al. (2007) for instance, when class labels are available, the instance generation has been mapped to the provided classification (i.e., class labels have been used to get instance level constraints). For that purpose, a must-link constraint $ML(d_i, d_j)$ is defined between two instances d_i and d_j if they are labeled as to be in the same class. On the other hand, there is a cannot-link $CL(d_i, d_j)$ between these elements that do not share the same label. Obviously, under this

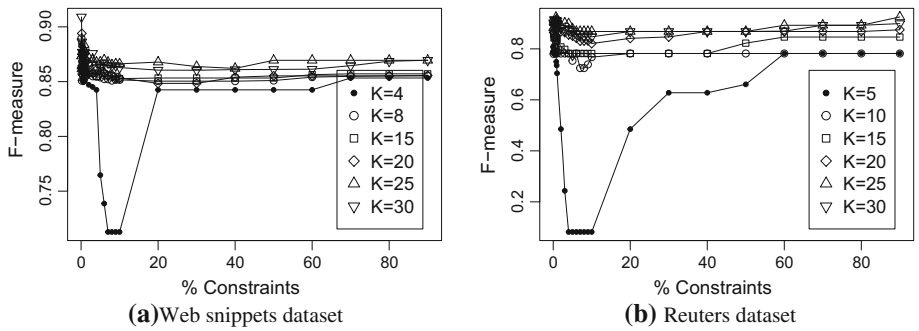


Fig. 3 F-measure with different k-mean configuration

(a) Web snippets dataset **(b)** Reuters dataset

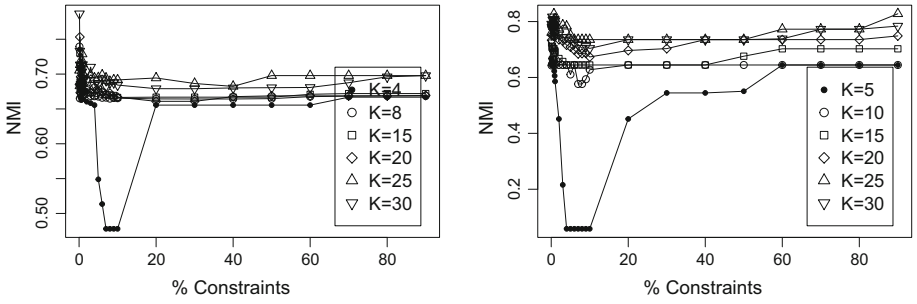


Fig. 4 Normalized mutual information with different k-mean configuration

(a) Web snippets dataset **(b)** Reuters dataset

model, it is possible to reconstruct the original partition using all the constraints generated under this model.

Our method uses semi-supervision to calculate the optimal partition of a dataset by means of hierarchical clustering. The advantage of using hierarchical clustering is that the number of clusters is not fixed, so it is possible to get more specific clusters than using other methods in which the number of clusters is fixed. Figures 7 and 8 show how this method performs in comparison (in terms of F-measure and NMI) with some traditional unsupervised methods such as k-means and Ward's method.

Our method (named *Auto* in the figures) has been evaluated with constraints generated with the best and the worst partitions for the two datasets (see Sect. 5.3), using 30 % of the constraints.

Figure 7 shows that the F-measure value from the semi-supervised method is higher than the F-measure value for the k-means and the Ward's method.

Let us remember that our semi-supervised approach makes use of hierarchical clustering and Ward's method to obtain the dendrogram. So, if we compare the partition obtained by cutting the dendrogram at the expected number of groups and our semi-supervised approach, let us observe that our method returns the more specific partition, since the clusters do not contain mixed data from different categories. Similar behaviour can be observed with the Normalized Mutual Information (Fig. 8).

Moreover, our method outperforms the k-means algorithm, even if k-means was part of the constraints generation process. It means that k-means is not able to find by itself an accurate partition that represents the data, so the provided pairwise information is very useful to find the optimal cut of the dendrogram. As our semi-supervised clustering algorithm has some tolerance for not fulfilled constraints, it is able to overtake the possible mistakes that could be in the constraints by k-means, without affecting the performance, as shown in Figs. 7 and 8.

5.5 Discussion

The main advantage of our automatic semi-supervised approach, compared with other semi-supervised clustering, is that it does not require some human expertise to provide external information but it is able to generate side information autonomously. Side information in the form of instance level constraints is further discussed in the following subsections focusing on the constraints size and, particularly, on the influence of these constraints on the partition of datasets.

5.5.1 Size of constraints

As shown in Fig. 2, the fixed number of cluster k in the k-means algorithm affects the number of constraints generated and, then, the performance of our method when obtaining the partition of the document corpus (Figs. 3 and 4).

The resulting partition of a documents collection by the k-means algorithm often shows some small specific clusters and a big cluster with a lot of mixed data (with respect to the class labels). Particularly, by increasing k, the clusters get more specific and provide better constraints. Indeed, the number of generated must-link constraints decreases as k increases because must links are generated between instances inside the same cluster, so having less elements per cluster provides less constraints. On the contrary, the cannot-link constraints increase, as more clusters mean more cannot-link possibilities. This behaviour is clearly observable in the Reuters dataset (Fig. 2), where with a small k a lot of must-link constraints are generated; but increasing k, those constraints tend to decrease.

However, let us remark that the specific correspondence between the amount of must-link and cannot-link strictly depends on the dataset, its size and its underlying class distribution.

Since constraints are generated automatically, they may have inaccuracies. It means that some of the constraints could be not related with the actual structure of the data. These inaccuracies are mainly solved as k increases. Specifically, since the constraints are being generated with a more specific partition, the resulting must-link/cannot-link information should be of better quality, with a consequent improved performance. Figures 3 and 4 describe this behaviour. As k is getting bigger and the number of cannot-link constraint increases, the performance improves because the clustering algorithm tends to put documents together, so that the cannot-

link constraints help to split up those groups. At the same time, the contribution coming from must-link constraints is also important, because if there are only cannot-link constraints, the results will contain one or two documents per cluster, and that behaviour is not desirable. A good trade-off between the use of must-links and cannot-links justifies assigning a bigger weight to the v_m and v_{nm} parameters (in Eq. 2), to compensate the outnumber of cannot-link constraints and their influence.

Figures 3 and 4 show also how the size of the sets of generated constraints included in our algorithm affects the performance of the method. In general, with $k = 15$ and beyond (values that provide a good partitioning on these datasets), the results in terms of performance tend to stabilize: we can see that the differences in the results, when adding more constraints, are small. Adding more constraints suppose adding more information in the process, which increases the computational complexity. For that reason, and taking into account also the size the dataset, using around 30 % of each must-link and cannot-link would be advisable.

5.5.2 *Partitioning driven by automatic constraint generation*

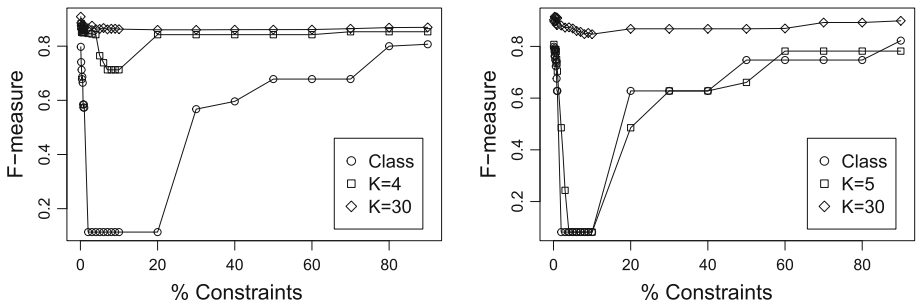
It is interesting to discuss the advantages of using automatically generated constraints instead of *human-provided* constraints.

Figures 5 and 6 also show how automatically generated constraints perform in comparison with constraints generated using class labels. These figures compare automatic vs. label-based generated constraints both for the *best* ($k = 30$) and the *worst* ($k = 4$) number of clusters. Under the same amount of constraints, the partitioning coming from automatic constraints generation performs better than the one obtained using constraints that come from class labels.

Most specific information can be obtained by studying the nature of the data (by means of the k-means-based analysis of data structure) and consequently, most-specific information about the data correlations (in form of constraints). Then, that information could help to find a most specific partition from the hierarchical clustering.

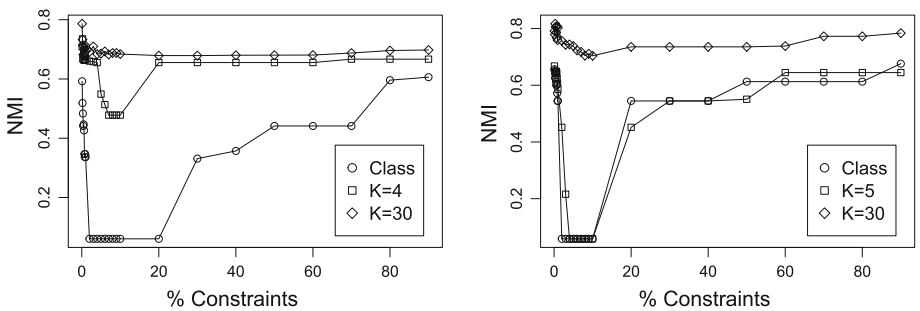
Another important point is the role of the hierarchy-based partitioning in comparison with flat clustering, as they are able to provide a more specific cluster assignment. In particular, Figs. 7 and 8 show how a specific partition outperforms those that come from obtaining the same number of partitions than the expected number of groups in the data. Our method does not modify the clustering algorithm itself, but it helps to find the better partition from the dendrogram D. It means that a partitioning that separates data into groups that "make sense" is already in the dendrogram, but it is not possible to obtain it but cutting the tree at the expected number of groups.

As an example, let us consider a dataset whose data have been split into several categories: where one of them is *sport*. Using the k-means algorithm to find the documents related with sports would not get accurate results. It probably would return some documents related with some specific sport or there would be a lot of documents, with the information regarding sports probably mixed with some unrelated



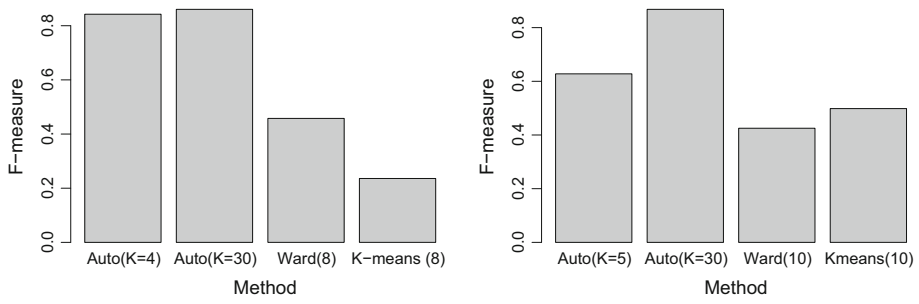
(a) Web snippets dataset **(b)** Reuters dataset

Fig. 5 Comparison of the F-measure of class and k-means automatically generated constraints



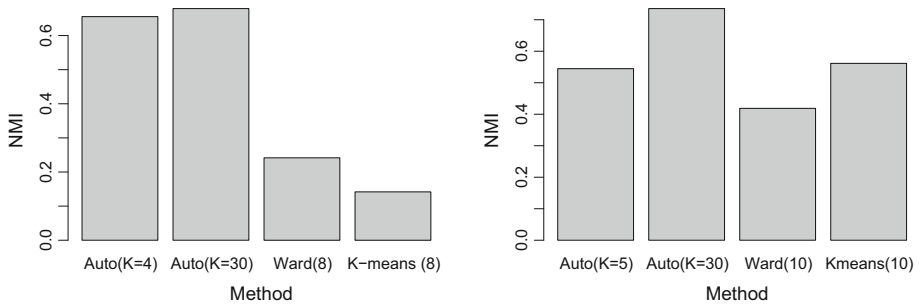
(a) Web snippets dataset **(b)** Reuters dataset

Fig. 6 Comparison of the Normalized Mutual Information of class and k-means automatically generated constraints



(a) Web snippets dataset **(b)** Reuters dataset

Fig. 7 Comparison of the F-measure of semisupervised and unsupervised method



(a) Web snippets dataset **(b)** Reuters dataset

Fig. 8 Comparison of the Normalized Mutual Information of semisupervised and unsupervised method

information. It depends on the "quality" of the document-term matrix: the selection of term features (for instance, co-occurrences of terms) is crucial to get a good partitioning. Anyway, a good term-document matrix does not guarantee an accurate partitioning in flat clustering. It is possible that class labels are quite specific and that there is not enough vocabulary in the documents to identify them.

Instead, in hierarchical clustering, the partitioning generally reveals better cluster specialization, for instance, in the case of sports, related to football, tennis, basketball, etc. and these clusters normally do not contain mixed information. Indeed, thanks to the nature of the dendrogram, it would be possible to use its hierarchy of documents to provide a more in-depth insight into the terms of corpus that could lead to a more specific hierarchical classification.

6 Conclusions

This work presents an approach to semi-supervised clustering of documents, by the automatic generation of instance level constraints.

The re-iterative application of the k-means algorithm allowed us to get a set of must-link/cannot-link constraints that could be used in a subsequent semi-supervised clustering stage. This second step uses the generated constraints along with the data as input of the semi-supervised clustering process. Final result is the partitioning of the document corpus.

The experimental results have shown:

- How the initial configuration of the k-means algorithm affects the quality and performance of the constraints, concluding that it is necessary to fix it at a value similar to the expected number of categories or higher.
- How the number of constraints used in the process affects the performance of the method. This test shows that the results have a tendency to stabilize when using a certain number of constraints, around 30 %, so not all generated information is needed in the process.
- The semi-supervised approach has been compared with both the unsupervised algorithms that intervene in the methodology. As a result, it has been observed that the semi-supervised approach outperforms both methods.

As a future work, we would study how the use of reiterated k-means executions to generate fuzzy constraints that could improve performance in datasets where the categories overlap. Additionally, it would be interesting to study the performance of automatically generated constraints with other semi-supervised clustering approaches.

Acknowledgments This work has been partially funded by the Spanish Ministry of Education under the "Programa de Formación del Profesorado Universitario (FPU)" and the Short Stays Program from CEI-Biotic (University of Granada).

References

Aggarwal C, Zhai C (2012) A survey of text clustering algorithms. Mining text data. Springer, US, pp 77–128

Barr J, Cament L, Bowyer K, Flynn P (2014) Active clustering with ensembles for social structure extraction. In: Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on. pp 969–976

Basu S, Banerjee A, Mooney RJ (2002) Semi-supervised clustering by seeding. In: Proceedings of the Nineteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, pp 27–34 (ICML '02)

Basu S, Bilenko M, Mooney RJ (2004) A probabilistic framework for semi-supervised clustering. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, New York, pp 59–68. doi:10.1145/1014052.1014062 (KDD '04)

Basu S, Davidson I, Wagstaff K (2008) Constrained clustering: advances in algorithms, theory, and applications, 1st edn. Chapman & Hall/CRC

Cutting DR, Karger DR, Pedersen JO, Tukey JW (1992) Scatter/gather: a cluster-based approach to browsing large document collections. In: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, pp 318–329. doi:10.1145/133160.133214 (SIGIR '92)

Diaz-Valenzuela I, Martin-Bautista MJ, Vila MA (2013) Using a semi-supervised fuzzy clustering process for identity identification in digital libraries. In: IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint. pp 831–836

Diaz-Valenzuela I, Martín-Bautista MJ, Vila MA (2014) A fuzzy semi-supervised clustering method: Application to the classification of scientific publications. In: Laurent A, Strauss O, Bouchon-Meunier B, Yager RR (eds) Information Processing and management of uncertainty in knowledge-based systems—15th International Conference, IPMU 2014, Montpellier, France, July 15–19, 2014. Proceedings, Part I, Springer, Communications in Computer and Information Science, vol 442. pp 179–188. doi:10.1007/978-3-319-08795-5

Grira N, Crucianu M, Boujemaa N (2004) Unsupervised and semi-supervised clustering: a brief survey. In: in 'A Review of Machine Learning Techniques for Processing Multimedia Content', Report of the MUSCLE European Network of Excellence FP6

Hu Y, Milios EE, Blustein J (2012) Semi-supervised document clustering with dual supervision through seeding. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing. ACM, New York, pp 144–151. doi:10.1145/2245276.2245306 (SAC '12)

Jain AK, Dubes RC (1988) Algorithms for clustering data. Prentice-Hall Inc, Upper Saddle River

Leuski A (2001) Evaluating document clustering for interactive information retrieval. In: Proceedings of the Tenth International Conference on Information and Knowledge Management. ACM, New York, pp 33–40, doi:10.1145/502585.502592 (CIKM '01)

Li X, Wang L, Song Y, Zhao X (2010) A hybrid constrained semi-supervised clustering algorithm. In: Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on, vol 4. pp 1597–1601

Loia V, Pedrycz W, Senatore S (2003) P-FCM: a proximity-based fuzzy clustering for user-centered web applications. Int J Approx Reason 34(2–3):121–144. doi:10.1016/j.ijar.2003.07.004

Pedrycz W, Loia V, Senatore S (2010) Fuzzy clustering with viewpoints. IEEE Trans Fuzzy Syst 18(2):274–284

Phan XH, Nguyen LM, Horiguchi S (2008) Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: Proceedings of the 17th International Conference on World Wide Web. ACM, New York, pp 91–100, doi:10.1145/1367497.1367510 (WWW '08)

Rigutini L, Maggini M (2005) A semi-supervised document clustering algorithm based on EM. In: Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on. pp 200–206. doi:10.1109/WI.2005.13

Sahoo N, Callan J, Krishnan R, Duncan G, Padman R (2006) Incremental hierarchical clustering of text documents. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management. ACM, New York, pp 357–366. doi:10.1145/1183614.1183667 (CIKM '06)

Tang W, Xiong H, Zhong S, Wu J (2007) Enhancing semi-supervised clustering: a feature projection perspective. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, New York, pp 707–716 (KDD '07)

Wagstaff K, Cardie C (2000) Clustering with instance-level constraints. In: Proceedings of the Seventeenth International Conference on Machine Learning. pp 1103–1110

Wagstaff K, Cardie C, Rogers S, Schrödl S (2001) Constrained k-means clustering with background knowledge. In: Proceedings of the Eighteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, pp 577–584 (ICML '01)

Xing EP, Ng AY, Jordan MI, Russell S (2002) Distance metric learning, with application to clustering with side-information. In: Advances in Neural Information Processing Systems 15, vol 15. pp 505–512. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.3667

Xiong S, Azimi J, Fern X (2014) Active learning of constraints for semi-supervised clustering. Knowl Data Eng IEEE Trans 26(1):43–54

Zhao W, He Q, Ma H, Shi Z (2012) Effective semi-supervised document clustering via active learning with instance-level constraints. Knowl Inf Syst 30(3):569–587. doi:10.1007/s10115-011-0389-1