A Method to extend Existing Document Clustering Procedures in order to include Relational Information

Tijn Witsenburg TIJN@LIACS.NL
Leiden Institute of Advanced Computer Science, Universiteit Leiden, Niels Bohrweg 1, Leiden, The Netherlands

Hendrik Blockeel BLOCKEEL@LIACS.NL
Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium
Leiden Institute of Advanced Computer Science, Universiteit Leiden, Niels Bohrweg 1, Leiden, The Netherlands

Abstract

We consider the problem of clustering nodes in a graph, where each node has also internal content (e.g., the Web, where nodes are web pages). In this context we can distinguish two kinds of information: content information and structural information. Standard clustering methods use content information only, while graph clustering methods are usually based on the graph structure. Relatively recently, researchers have proposed to combine both types of information. In this paper we propose a very simple, yet hitherto unexplored, method for doing this by extending existing clustering procedures that use content information.

1. Introduction

Clustering is an important task in machine learning. We can distinguish "standard clustering" algorithms and "graph clustering" algorithms. In **standard clustering**, items are clustered according to their similarity whilst not taking into account any relational information. A distance or similarity function is given which for any pair of items tells us how similar they are. In this way a $N \times N$ matrix can be created with all these values. In **graph clustering**, unlabeled graphs are considered, where clustering (or partitioning) the graph typically means finding subgraphs of the graph such that the number of links connecting different subgraphs is as small as possible whilst not taking into account any information about the content of the node. Any $N \times N$ matrix can be converted to a graph and vice versa. This raises the question whether, in those cases where both node content and graph structure are available (such as the Web), one could find a clustering method that combines both types of information.

Neville et al. (2003) discuss this problem, and discuss a number of possible solutions. In the combined method they propose, the structure of the graph remains the same; the edges of the graph are given weights that correspond to the similarity between the nodes they connect, then a graph clustering algorithm is applied to them. Neville et al. compare different graph clustering algorithms.

In this work, we take an opposite direction: instead of introducing the content information in the graph (in the form of edge weights that indicate the similarity between nodes), we will inject the structure information into the similarity function, after which a standard clustering algorithm is used. One could say that Neville et al. map the hybrid clustering task onto graph clustering, whereas we will map it onto standard clustering.

2. The Method

2.1. General Principle

A number of clustering procedures that work on independent data sets do this by first creating a distance or similarity matrix M and then use M to cluster the items. To create this matrix it is needed to define a distance or similarity measurement that works on the features of the items in the data set. This measurement is then used to calculate the value of all elements in M where m_{ij} is the distance or similarity between each pair of items i and j in the data set.

It could be that items in the data set are not entirely independent and that there are also relations between items. To incorporate this relational information, a new matrix, the adjacency matrix A, is created besides matrix M. These two matrices are combined with a method that is based on matrix multiplication. This results in a new matrix M' which then can be used for clustering the items in the data set, assuming all constraints on M for the clustering procedure are still met in M'. This puts some constraints on the procedure to create M'.

When the values of M' need to be in a certain range, the correct constants need to be chosen. Another important constraint is the fact that a distance or similarity matrix used for clustering needs to be symmetric. Clustering will only go well when the distance from i to j is the same as the distance from j to i. For both constraints it will become clearer later on how they are met. The last important constraint is more intuitive. When two nodes i and j look alike, they have a small distance or a high similarity. The values in matrix M' should follow the same principle, although their meaning is not completely the same as the meaning of the values in M.

2.2. Definitions

Consider a data set with N items. Matrix M is a symmetric $N \times N$ matrix where each element m_{ij} is the distance or similarity between nodes i and j. How these elements are calculated depends on the clustering procedure we would like to adapt and how this is done in this experiment will be discussed in section 3.2. Matrix A is an adjacency matrix created from the graph describing the used relation. In order to ensure the symmetry of matrix M', matrix A also needs to be symmetric. Therefore an element a_{ij} in A is 1 when there is a relation from i to j or from j to i.

Once M and A are created, any element in M' can be created with equation 1. Notice that when M' is constructed using equation 1 this can also be written as $M' = M \times A + A \times M$.

$$m'_{ij} = \sum_{n=0}^{N}(m_{in} \cdot a_{nj}) + \sum_{n=0}^{N}(a_{in} \cdot m_{nj}) \qquad (1)$$

The practical meaning of the values in M' can easily be understood when taking a closer look at equation 1. Considering the first part ($\sum_{n=0}^{N}(m_{in} \cdot a_{nj})$) for every a_{nj} it holds that it is 0 when there is no relation between node n and node j and 1 otherwise. Thus, this first part will sum all values m_{in} for which a_{nj} is equal to 1. This can be described by saying that the first part gives the sum of all distances from node i to all neighbours of j in the graph describing the relations. Analogously the second part is the sum of all distances or similarities from node j to all neighbours of i.

In some cases the constraints in section 2.1 are not all met yet. When examining this more closely, it is best to keep in mind that whatever holds for the first part of equation 1 also holds for the second part, but then with i and j reversed. The first part of equation 1 gives the sum of the distances or similarities between node i and the neighbours of node j. When j has a lot of neighbours that look like i, intuitively, it would be preferred that the value for m'_{ij} would be such that the clustering procedure would consider i and j to look alike.

When M is a distance matrix, nodes that look alike have a small distance and thus, their value in M is low. The more small distances are added in equation 1, the higher its result in M'. This is the opposite of what is preferred. Therefore, instead of using the sum of the distances between i and the neighbours of j it would be better to use their average value. To ensure this, equation 1 needs to be extended to equation 2.

$$m'_{ij} = \frac{1}{2} \cdot \left(\frac{\sum_{n=0}^{N}(m_{in} \cdot a_{nj})}{\sum_{n=0}^{N} a_{nj}} + \frac{\sum_{n=0}^{N}(a_{in} \cdot m_{nj})}{\sum_{n=0}^{N} a_{in}} \right) \qquad (2)$$

The constant '1/2' in equation 2 ensures that all values in M' are in the same range as the values in M. That was not possible in equation 1.

3. First Results

3.1. Cora Data Set

For our first experiments we used the Cora data set (McCallum et al., 2000). This is a big data set with scientific papers divided in 70 classifications. Of 37,000 of these papers the abstracts are available for keyword extraction and the citations between papers are also available. The disadvantage of this data set is that it is created automatically. This has resulted in the fact that several papers have more than one abstract and some have more than one classification. Therefore errors in the data set can be expected.

3.2. Setup

First, matrix M was created by creating for every paper the list of words that are in that paper, the so-called bag-of-words. When b_i is defined as the bag-of-words for paper i, every value in M can be calculated with equation 3.

$$m_{ij} = \frac{1}{2} \cdot \left(\frac{|b_i \cap b_j|}{|b_i|} + \frac{|b_i \cap b_j|}{|b_j|} \right) \qquad (3)$$

It can easily be seen that this is the average ratio of words that are in common between two papers. Second, matrix A was created by taking into account the citation relation where a_{ij} is 1 when paper i cites paper j or is cited by it and 0 otherwise.

Table 1. Best found F-score on several partitions of the Cora data set for three clustering methods.

SET	PAPERS	CLASSES	M	M'_{sum}	$M'_{average}$
1	1104	8	0.531	0.545	0.584
2	2294	17	0.354	0.365	0.416
3	3209	24	0.332	0.364	0.382
4	5725	31	0.336	0.342	0.361
5	9055	45	0.310	0.343	0.350

Since we use a similarity measurement, both equation 1 (resulting in M'_{sum}) and equation 2 (resulting in $M'_{average}$) can be used. They are compared with using M for clustering which can be seen as clustering only on content without any relational information.

These three matrices will be clustered using a simple and greedy clustering procedure. It takes the matrix used for clustering and considers every node to be in one cluster. Then it finds the two clusters that are closest according to this matrix. These two clusters are joined together to form a new cluster and all distances to the other clusters will be the average of the distances to the clusters that formed the new cluster. This will be repeated until there is only one cluster left. After every step the quality of the clustering is measured using the known labelling from the Cora data set. The best clustering is chosen as the final score.

As a measurement for the clustering, F-score (Larsen & Aone, 1999) was used. The F-score is calculated by using equation 4 where P is the precision (number of documents labelled of class C in that cluster, divided by the number of documents in that cluster) and R is the recall (number of documents labelled of class C in that cluster, divided by the number of documents of class C in the total database). The value of F-score is between 0 and 1 where 1 is a perfect score.

$$\text{F-score} = \frac{2PR}{P+R} \tag{4}$$

3.3. Results

Only papers with exactly one abstract and one classification were considered. Of this partition we used first only 8 classes to create data set '1' and extended this data set by repeatedly adding more classes as can be seen in Table 1. Any of these sets where clustered in three ways, using M, M'_{sum} and $M'_{average}$. Table 1 shows the best found F-score for any of these matrices.

The first thing to notice is that the results are not very good. This is probably caused by the fact that the Cora data set might contain an error and the fact that the clustering procedure used to incorporate our method in, is not considered to be very good. Still, since all circumstances are the same for any of the three clustering methods, they can be compared to each other. A first cautious conclusion can be that for this case it seems to be that adding relational information does enhance the performance of this clustering procedure on these data sets. This is especially the case when the average M-values are used. This setting even seems to outperform the one where the sum of the values is used. Despite these results, still a lot more research needs to be done.

4. Conclusions and Future Work

In a relational context, one may want to cluster objects based on both their content and the relationships between them, with the latter being indicated by a graph. We have proposed a method for adapting the distance or similarity matrix in such a way that relational information is inserted. An experiment on the Cora data set shows that this more informed distance measure leads to better clustering results when it is plugged into a standard clustering procedure.

This is work in progress and more needs to be done. Its performance should be tested on other data sets and while using different clustering procedures. Besides that, it could be very interesting to explore different variations of this method like, for instance, using only incoming (or outgoing) neighbours and thus creating an asymmetric matrix M'.

Acknowledgments

Hendrik Blockeel is a postdoctoral fellow of the Fund for Scientific Research of Flanders (FWO). This research is funded by the Dutch Science Foundation (NWO) through a VIDI grant.

References

Larsen, B., & Aone, C. (1999). Fast and effective text mining using linear-time document clustering. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 16–22).

McCallum, A., Nigam, K., Rennie, J., & Seymore, K. (2000). Automating the construction of internet portals with machine learning. *Information Retrieval Journal*, *3*, 127–163.

Neville, J., Adler, M., & Jensen, D. (2003). Clustering relational data using attribute and link information. *Proceedings of the Text Mining and Link Analysis Workshop, Eighteenth International Joint Conference on Artificial Intelligence.*