Relative Constraints as Features

Piotr Lasek[1] and Krzysztof Lasek[2]

[1] Chair of Computer Science, University of Rzeszow,
ul. Prof. Pigonia 1, 35-510 Rzeszow, Poland,
lasek@ur.edu.pl
[2] Institute of Computer Science, Warsaw University of Technology,
ul. Nowowiejska 15/19, 00-665 Warszawa, Poland,
klasek@elka.pw.edu.pl

Extended abstract

Abstract. One of most commonly used methods of data mining is clustering. Its goal is to identify unknown yet interesting and useful patters in datasets. Clustering is considered as unsupervised, however recent years have shown a tendency toward incorporation external knowledge into clustering methods making them semi-supervised methods. Generally, all known types of clustering methods such as partitioning, hierarchical, grid and density-based, have been adapted to use so-called constraints. By means constraints, the background knowledge can be easily used with clustering algorithms which usually leads to better performance and accuracy of clustering results. In spite of growing interest in constraint based clustering this domain still needs attention. For example, a promising relative constraints have not been widely investigated and seem to be very promising since they can be easily represent domain knowledge. Our work is another step in the research on relative constraints. We have used and simplified the approach presented in [1] so that we created a mechanism of using relative constraints as features.

Keywords: clustering, density-based clustering, constrainted clustering, relative constraints

1 Introduction

Normally, clustering proceeds in an unsupervised manner. However, in some cases, it is desirable to incorporate external knowledge into a clustering algorihtm. Let us consider the following example. A system designed for analysing customers' data of a GSM company is going to be built. The purpose of this system is to give managers the ability to detect new groups of customers for which new and customised individual plans could be offered. The number of customers is large and each of them can be described by the great number (thousands) of attributes. In order to analyse and cluster the dataset any clustering algorithm that is capable of dealing with large and high dimensional datasets can be applied. However, the company employs a number of experinced analysts which

know the market and can give invaluable insights into the system. The question is how to easily incorporate analysts' knowledge into clusteiring algorithms so that it could enrich the clustering results.

One of the ways of incorporating external knowledge into clustering algorithms is to use so-called constraints [2]. Several types of constraints are known. In [6] authors introduced simple yet very popular instance-level constraints, namely: the *must-link* and *cannot-link* constraints. If we say that two points p_0 and p_1 are in a *must-link* relationship (or are connected by a *must-link* constraint) then, by means of a clustering process, these points will be assigned to the same cluster c. On the other hand, if we say that two points p_0 and p_1 are in a *cannot-link* relationship (or are connected by a cannot-link constraint) then these points will not be assigned to the same cluster c.

Incorporation of of just few constraints of above type can increase clustering accuracy as well as decrease runtime [3].

In [1], Asafi and Cohen-Or presented an interesting method of incorporating instance constraints into any clustering algorithm. They proposed to treat constraints as additional features of a given object. In order to incorporate these contraints, they alter the original distance matrix so that they set the distances between objects in a *must-link* relationship to shortest distance between any two objects in the input dataset. Then, the triangle inequality is restored. Additionaly, in order to take *cannot-link* constraints into account, they employed the idea of the *diffusion distance* which was used to compute modified distances between objects satisfying *cannot-link* constraints [7]. In other words, the process of using *instance-level* constraints as features is composed of two steps. First, *must-link* constraints are used to modify a distance matrix. Then, by using the *Diffusion Map* to compute *diffusion distances*, *cannot-link* constraints are taken into account. The above reminded computations are performed using the following formulas:

$$\widetilde{D}_{i,j} = \widehat{D}_{i,j} + \sum_{c=1...N} \alpha D^{(c)}_{i,j},$$

where $\widehat{D}_{i,j}$ is the matrix created by adding *must-link* constraints to the original distance matrix $D_{i,j}$, α is used to scale distance space from $(-1, 1)$ interval to $(-\alpha, \alpha)$ (α is the longest distance in $D_{i,j}$), $\widetilde{D}_{i,j}$ is the matrix constructed by adding $D^{(c)}_{i,j}$ matrices to it, where $c = 1, ..., N$ (N is a number of *cannot-link* constraints) and each $D^{(c)}_{i,j}$ matrix represents one *cannot-link* constraint. $D^{(c)}_{i,j}$ is called the diffusion distance matrix [8] and is computed using the following formulas:

$$D^{(c)}_{i,j} = |v_i - v_j|,$$

$$v_i = \frac{\varphi(i, c_2) - \varphi(i, c_1)}{\varphi(i, c_2) + \varphi(i, c_1)},$$

where c_1 and c_2 are *cannot-link* objects in a *cannot-link* relationship and i is the index of the object in the dataset. φ is the function given by the following formula:

$$\varphi(x, y) = |\Psi_t(x) - \Psi_t(y)|,$$

where

$$\Psi_t(x) = (\lambda_1^t \psi_1(x), \lambda_2^t \psi_2(x), \ldots, \lambda_1^t \psi_n(x))^T,$$

where λ_i^t are the eigen values and ψ_i are the eigen vectors of the input dataset's *Diffusion Map*, n is the number of objects in a dataset and t is a time parameter which can be determined experimentally [7].

Diffusion maps are based on the observation that when walking from one point to another in a dataset, it is more probable to traverse points that are located nearby than far away. This observation leads to defining the probability of traversing from point a to b. As described above, in order to use a Diffusion Map, several computation has to be done. First, the distance matrix needs to be built. Then, the computed distance matrix has to be normalized by using the sums of its rows. Next, the spectral decomposition is performed to using eigenvalues and eigenvectors of the previously normalized matrix. The dimensionality reduction is performed by omitting the smallest eigenvalues. For this reason diffusion maps can be used to deal with high dimensional datasets by discovering an underlying manifold from data and to give a simpler global description of the dataset [6].

2 Relative constraints as features

In recent studies relative constraints gain more and more attention due to the fact that they can easily represent domain knowledge [4] [5]. Relative constraints are usually defined as object triples and are presented in the following way: $ab|c$, where a, b and c are objects from the dataset and a is closer to b than to c. The formula $ab|c$ is equivalent to the following comparision: $d(a, b) < d(a, c)$ where d is the distance function. The intuition behind relative constraints (comparing to *instance-level* constraints) is that it may be easier to define relative similarity between objects and use this knowledge in a process of clustering than strictly define which objects should or should not be assigned to specific clusters.

Our proposal of using relative constraints comprises the ideas presented by [1] and [2]. In our approach we employ relative constraints into the clustering process so that we similarly construct modified diffusion distance matrix. However, our method may be considered simpler because of the fact that we do not have to restore triangle inequality property. Because of that, the resulting distance matrix is given by the following formula:

$$\widetilde{D}_{i,j} = D_{i,j} + \sum_{r=1\ldots N} \alpha D_{i,j}^{(r)},$$

where r is a set of relative constarints (a set of objects triples).

Further, in our method in order to compute the diffusion distance matrix $D_{i,j}^{(r)}$ of diffusion distances between objects, the following formulas are used:

$$D_{i,j}^{(r)} = |v_i - v_j|,$$

where

$$v_i = \frac{min(\varphi(i,a) + \varphi(i,b)) - \varphi(i,c)}{min(\varphi(i,a) + \varphi(i,b)) + \varphi(i,c)},$$

where a, b and c are points which are in a relative relationship $ab|c$.

3 Experiments and results

In order to test our method we implemented a framework for performing data clustering experiments. We have implemented appropriate functions for computing diffusion distance matrices which were later used in a neighborhood-based clustering algorithm (NBC) [9] to determine nearest neighbors. We have performed a number of experiments using several well known benchmarch datasets [10]. Due to the fact that in order to determine diffusion distance matrix a eigen vectors end eigen values must be computed, the overall time efficiency of the method is low. However, the qualitative results are very promising. Moreover, in comprasion to instance-level, relative constraints can be specified by experts more easily since an a priori knowledge about assignement of the object to the same cluster is not requried. The only information necessary to obtain from a domain expert is the specification of the relation between two objects.

4 Conclusion and further research

In the nearest future we are going to focus to make our method more efficient. Moreover we want to focus on examination of the influence of the t parameter on the quality and efficiency of our method. Additionally we would like to test different core functions used for when determining the *Diffusion Map* ane check their influence on the results of the clustering.

References

1. S. Asafi and D. Cohen-Or, "Constraints as Features," in IEEE Conference on Computer Vision and Pattern Recognition (2013), 2013, pp. 1634–1641.
2. S. Basu, I. Davidson, and K. Wagstaff, Constrained Clustering: Advances in Algorithms, Theory, and Applications, 1 edition, vol. 45. 2008, pp. 961–970.
3. I. Davidson and S. Ravi, "Clustering With Constraints: Feasibility Issues and the k-Means Algorithm," in Proceedings of the 2005 SIAM Int. Conf. on Data Mining, pp. 138–149.
4. M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In NIPS 04, 2004.
5. C. Semple and M. Steel. A supertree method for rooted trees. Discrete Applied Mathematics, 105(1-31-3):147158, 2000.
6. Nadler, Boaz, et al. Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators. arXiv preprint math/0506090 (2005).
7. De la Porte, J., et al. An introduction to diffusion maps. Applied Mathematics Division, Department of Mathematical Sciences, University of Stellenbosch, South Africa and Colorado School of Mines, United States of America (2008).
8. Lafon, Stephane, and Ann B. Lee. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. Pattern Analysis and Machine Intelligence, IEEE Transactions on 28.9 (2006): 1393-1403.

9. Zhou, Shuigeng, et al. "A neighborhood-based clustering algorithm." Advances in Knowledge Discovery and Data Mining. Springer Berlin Heidelberg, 2005. 361-371.
10. Asuncion, Arthur, and David Newman. "UCI machine learning repository." (2007).