Clustering with Constrained Similarity Learning

Masayuki Okabe
Toyohashi University of Technology
Tenpaku 1-1, Toyohashi, Aichi, Japan
okabe@imc.tut.ac.jp

Seiji Yamada
National Institute of Infomatics
Chiyoda, Tokyo, Japan
seiji@nii.ac.jp

***Abstract*—This paper proposes a method of learning a similarity matrix from pairwise constraints for interactive clustering. The similarity matrix can be learned by solving an optimization problem as semi-definite programming where we give additional constraints about neighbors of constrained pairwise data besides original constraints. For interactive clustering, since we can get only a few pairwise constraints from a user, we need to extend such constraints to richer ones. Thus this proposed method to extend the pairwise constraints to space-level ones is effective to interactive clustering. First we formalize clustering with constrained similarity learning, and then introduce the extended constraints as linear constraints. We verify the effectiveness of our proposed method by applying it on a simple clustering task. The results of the experiments shows that our method is promising.**

I. Introduction

Interactive clustering is a significant method to realize intelligent Web interaction because it is very useful to interactive visualization, data mining, data analysis of the web[Wu04]. Interactive clustering consists of two main techniques: active learning and constrained clustering. The active learning is for selecting effective data, which are judged by a user and then used as constraints. Also the constrained clustering is used to categorize the data under such constrains from a user. The common requirement in the both technique is to cope with a few constraints form a user. In this paper, we propose constrained clustering to utilize such a few constraints as much as possible by extending a few constraints to richer ones. Semi-Supervised learning that makes classifiers or clusters from a little supervised information and a lot of unlabeled data has been researched vigorously in recent years. Constrained clustering is one of such learning problems. In the typical setting of constrained clustering, several numbers of pairwise data are given as either must-link or cannot-link constraints. The pair of must-link data must be in the same cluster. On the other hand, the pair of cannot-link data cannot be in the same cluster.

One simple use of these constraints is to build a procedure into clustering algorithms to check whether temporal clusters break constraints or not. COP-Kmeans proposed by Wagstaff [Wagstaff 01] is such a representative method. COP-Kmeans is a modified version of the K-means algorithm. It first selects cluster centers, and second allocates each data to one of those centers not to break given constraints, i.e. not to allocate pairs of data with must-link to different clusters and not to allocate pairs of data with cannot-link to the same cluster. This approach is simple but it becomes too difficult to make consistent clusters with constraints when the number of must/cannot-link pairs increases. Since COP-Kmeans does only greedy search, its clustering procedure may stop on the way.

Meanwhile, another approach uses constraints to modify similarities between data as similarities of must-link pairs are forced to be large and similarities of cannot-link pairs are forced to be small. There are several methods to realize this approach [Klein 02], [Shwartz 04], [Davis 07], [Tang 07], [Hoi 07], [Li 08]. For example, Klein et. al proposed a clustering algorithm to deal with space-level constraints in addition to ordinary instance-level constrains with must/cannot-links [Klein 02]. They claimed neighbors around data constrained by must/cannot-links should be constrained in the similar way to the data, and developed concrete methods to propagate space-level constrains by satisfying triangle inequality with all-pairs-shortest-paths and utilizing complete-link hierarchical agglomerative clustering. Li et. al proposed a method to learn a kernel matrix that is obtained by solving an optimization problem as semi-definite programming. They integrate must/cannot-link constraints into the optimization problem to propagate local constraints to the whole kernel matrix. However how the must/cannot-link constraints influence the kernel value of the other pairs is not clear though it ensures the kernel value of constrained pairwise data.

Based on these two methods, we propose a similarity learning method that produces a similarity matrix. Though our method obtains the similarity matrix by solving an optimization problem as the same way Li et. al do, we impose additional constraints about neighbors of constrained pairwise data on it. By imposing the additional constraints, similar data move together when given must/cannot-link to one of them. This approach is very similar to Klein's in terms of propagating space-level constrains under a complete-link hierarchical agglomerative clustering, however our approach does not need a special clustering algorithm and is independent of a clustering algorithm. Also our method can control coverage of the propagation by changing the number of neighbors.

978-0-7695-3801-3/09 $26.00 © 2009 IEEE
DOI 10.1109/WI-IAT.2009.223

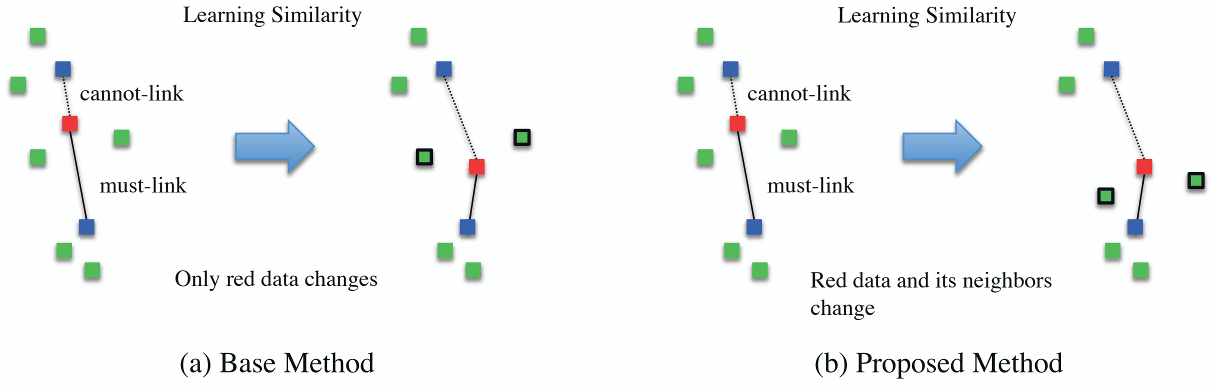

Figure 1. Difference of neighbor's movement between base and proposed method

In the following sections, we first explain about similarity learning and formalize it as semi-definite programming in section 2. Then we describe about our proposed method that impose additional constraints to the original optimization problem in section 3. section 4 shows experimental results where obtained similarity matrix is applied on a simple clustering task. Finally, we conclude our work in section 5.

II. Similarity Learning

In this section, we formalize a similarity learning as a semi-definite programming.

Let P denote a data collection where each data $\vec{p}_i \in R^m (i = 1, ..., n)$ is a vector of length m. Let $S \in R^{n\times n}$ denote its similarity matrix where each $s_{ij} (0 \leq s_{ij} \leq 1)$ is a similarity between $\vec{p}_i$ and $\vec{p}_j$. Pairwise constraints are given as follows.

$$\begin{aligned} M &= \{(i,j) \mid (\vec{p}_i, \vec{p}_j) \text{ is a must-link pair}\} \\ C &= \{(i,j) \mid (\vec{p}_i, \vec{p}_j) \text{ is a cannot-link pair}\} \end{aligned}$$

The objective of similarity learning is to create a new similarity matrix K that satisfies above constraints. We formalize an optimization problem to obtain K as semi-definite programming. Before formalizing, we describe about the graph Laplacian that is used as coefficient for K in the optimization problem. Let D a diagonal matrix where $d_{ii} = \sum_{j=1}^{n} s_{ij}$. We can define the graph Laplacian L as follows.

$$L = D - S$$

Using its normalized version $\bar{L} = I - D^{-\frac{1}{2}} S D^{-\frac{1}{2}}$, we can formalize an optimization problem as follows:

$$\begin{aligned} \min_K : \quad & \bar{L} \bullet K \\ s.t. : \quad & k_{ii} = 1, \quad i = 1, ..., n \\ & k_{ij} = 1, \quad {}^{\forall}(i,j) \in M, \\ & k_{ij} = 0, \quad {}^{\forall}(i,j) \in C, \\ & K \succeq 0 \end{aligned}$$

where $\bar{L} \bullet K$ represents the inner product between $\bar{L}$ and K. $\bar{L} \bullet K$ is calculated as follows.

$$\bar{L} \bullet K = \sum_{i=1}^{n} \sum_{j=1}^{n} \bar{l}_{ij} k_{ij}$$

$\bar{l}_{ij}$ is the element of $\bar{L}$. This objective function measures inconsistency between the original and new similarity matrix. Thus the solver of the optimization problem tends to keep similarities between pairs with no constraints the same value as far as possible. Pairwise constraints are translated into $k_{ij} = 1$ for must-link pairs, $k_{ij} = 0$ for cannot-link pairs. $K \succeq 0$ means K is a semi-definite matrix. This condition is necessary to guarantee the solution is a metric on the Euclidean space. By solving this optimization problem, we can obtain a new similarity matrix K that is used for semi-supervised learning.

III. Extending constrains from pairwise to neighbors

The optimization problem introduced in the previous section returns a similarity matrix. To satisfy the pairwise constraints and the semi-definite condition, returned similarity matrix changes from the original one. Although the purpose of similarity learning is not only to modify similarities between constrained data pairs but also to influence those constraints to their neighbors, it is not clear how the influence of constraints are propagated to non-constrained data. Thus we propose to impose additional constraints about neighbors of constrained data pairs to the optimization problem.

We illustrate this desirable effect in Figure 1, where two pairs of data (one data of red color and two data of blue color) are constrained as must-link and cannot-link respectively. Since the original optimization problem considers only the relation between constrained data pairs, the solution may result the data allocation illustrated in Figure 1(a), where only the data of red color moves. However we expect that not only constrained data pairs but also their neighbors are affected by the constraints illustrated in Figure 1(b), where the data of red color and its neighbors (two data of green color maked with black bold rectangle) move together. To realize the latter phenomena, we impose additional constraints to the optimization problem. More specifically, for a data pair (i, j) with must-link or cannot-link, we impose the following constraints:

$$\begin{array}{ll} k_{jr_t^i} \leq -\bar{l}_{jr_t^i}, & if\ (i,j) \in M \\ k_{jr_t^i} \geq -\bar{l}_{jr_t^i}, & if\ (i,j) \in C \end{array}$$

where $k_{jr_t^i}$ $(t = 1, .., k)$ is neighbors of $\vec{p}_j$, and $-\bar{l}_{ij} = \frac{s_{ij}}{\sqrt{d_{ii}d_{jj}}}$ is a value of regularized s_{ij}. In the same manner, we impose the following constraints about $\vec{p}_j$.

$$\begin{array}{ll} k_{ir_t^j} \leq -\bar{l}_{ir_t^j}, & if\ (i,j) \in M \\ k_{ir_t^j} \geq -\bar{l}_{ir_t^j}, & if\ (i,j) \in C \end{array}$$

These constraints means not only $\vec{p}_j$ but also its neighbors $k_{jr_t^i}$ move closer to $\vec{p}_i$ if (i, j) is must-link, and they move farther if (i, j) is cannot-link. After all, the final optimization problem is formalized as follows.

$$\begin{array}{lll} \min_K : & \bar{L} \bullet K & \\ s.t. : & k_{ii} = 1, & i = 1, ..., n \\ & k_{ij} = 1, & \forall (i,j) \in M, \\ & k_{ij} = 0, & \forall (i,j) \in C, \\ & k_{ir_t^j} \leq -\bar{l}_{ir_t^j},\ k_{jr_t^i} \leq -\bar{l}_{jr_t^i}, & \forall (i,j) \in M \\ & k_{ir_t^j} \geq -\bar{l}_{ir_t^j},\ k_{jr_t^i} \geq -\bar{l}_{jr_t^i}, & \forall (i,j) \in C \\ & K \succeq 0 & \end{array}$$

The fourth and fifth constraints are different from the original problem. By imposing these constraints we intend to ensure that similarities between constrained pairwise data and their neighbors changes in the same way.

IV. Experiments

In this section, we evaluate our proposed method on a clustering task. We use a dataset **tr31** from evaluation datasets for the CLUTO system. The **tr31** is a document dataset derived from TREC collection. It consists of 926 documents and 7 categories. We use cosine distance for the initial similarity measure, and the SDPA package[1] for the optimization tool, and K-medoids algorithm for clustering. The procedure including K-medoids algorithm is described below.

1) Calculate the initial similarity matrix S.
2) Solve the optimization problem described in section 3 and obtain a new similarity matrix K.
3) Select initial cluster centers $c_i (i = 1 \sim N_k)$
4) For each data p_i, sort c_i in descending order of the similarity between p_i and c_i. Assign p_i to the top c_i.
5) After assignment finished, calculate whole similarity Ddescribed below.
$$D = \sum_{t=1}^{N_k} \sum_{p_i \in C_k} (p_i - c_t)^2$$
where C_k is a cluster. Let D as D^{orig}
6) For each p_i, replace p_i and each c_i and calculate D^{tmp}. If $D^{tmp} - D^{orig} < 0$, stock the pair (p_i, c_i). Then find the pair (p_i^*, c_i^*) that produce the smallest $D^{tmp} - D^{orig}$. Replace p_i^* and c_i^*. Let $D^{orig} = D^{tmp}(p_i^*, c_i^*)$, and return procedure 2. If no pair (p_i, c_i) produce $D^{tmp} - D^{orig} < 0$, this algorithm stops and return the temporal clusters.

We compare the following two methods to investigate the effect of additional constraints we impose.

- A method with constraints about must-link and cannot-link that are originally given.(KK-MEANS)
- A method with constraints about neighbors in addition to the original must-link and cannot-link (NKK-MEANS). In the experiments, we set $k = 1$ for the number of neighbors to consider as constraints.

We use *Normalized Mutual Information* (NMI) as the performance measure. This measure is defined as follows.

$$\text{NMI}(C, T) \quad = \quad \frac{I(C, T)}{\sqrt{H(C)H(T)}}$$

where C is the set of clusters returned by the k-means algorithm, and T is the set of true clusters. $I(C, T)$ is the mutual information between C and T, and $H(C)$ and $H(T)$ are the entropies.

We test several numbers of constraints. For each number of constraints, constrained data pairs are randomly generated 10 times. Initial seeds for K-means are also randomly generated 10 times. Thus the evaluated values are the average of total 100 results. Table I shows the results. For the experiments of NKK-MEANS, we change the number of neighbors for the additional constraints (represented by t in the table). The performance of our proposed method NKK-MEANS is better than KK-MEANS at evary number of constraints when $t = 1$. NKK-MEANS needs only 50 constraints to achieve KK-MEANS's performance with 100 constraints. This property is very useful if we do not have enough number of constraints such as interactive clustering. However we need more experiments and analysis to realize the reason that the performance degrades as t increases for every number of constraints.

V. Conclusion

We proposed a method of learning a similarity matrix from pairwise constraints. Our method is based on the same approach proposed by Li et.al, which produces a similarity matrix by solving an optimization problem as semi-definite programming. However we impose additional

[1] http://sdpa.indsys.chuo-u.ac.jp/sdpa/

Table I
RESULTS

Num. of constraints	KK-MEANS	NKK-MEANS				
		t=1	t=2	t=3	t=4	t=5
10	0.076	0.150	0.100	0.08	0.03	0.06
50	0.173	0.225	0.224	0.214	0.189	0.185
100	0.222	0.239	0.235	0.207	0.231	0.211
200	0.254	0.288	0.240	0.235	0.258	0.270
500	0.304	0.331	0.308	0.320	0.342	0.314

constraints that neighbors of constrained pairwise data are also influenced by the constraints, i.e. neighbors of a must-linked pair also become similar to the pair, and neighbors of a cannot-link pair also become not-similar. Experimental results on a simple clustering task shows that our approach is promising though we must test it on other test beds and analyze in detail.

In future work, we have a plan to conduct systematic experiments to evaluate our method in documents clustering using large data bases. Also we will investigate the influence of the number of neighbors and develop how to control it, and find out more effective constrains on the neighbors.

REFERENCES

[Wu04] Wu, W., Yu, C., Doan, A. and Meng, W., "An interactive clustering-based approach to integrating source query interfaces on the deep Web", In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pp.95-106, 2004.

[Wagstaff 01] Wagstaff, K. and Roger, S. : "Constrained K-means Clustering with Background Knowledge", In *Proceedings of the 18th International Conference on Machine Learning* , pp.577-584, 2001.

[Klein 02] Klein, D., Kamvar, S. D. and Manning, C. D., "From Instance-level Constraints to Space-Level Constraints: Making the Most of Prior Knowledge in Data Clustering", In *Proceedings of the 19th International Conference on Machine Learning*, pp.307-314, 2002.

[Shwartz 04] Shwartz, S., Singer, Y. and Ng, A. Y. : "Online and Batch Learning of Pseudo-Metrics", In *Proceedings of the 21st International Conference on Machine Learning* , pp.94-101, 2004.

[Davis 07] Davis, J., Kullis, B. and et.al. : "Information-Theoretic Metric Learning", In *Proceedings of the 24th International Conference on Machine Learning* , pp.209-216, 2007.

[Tang 07] Tang, W., Xiong, H., Zhong, S. and Wu, J. : "Enhancing Semi-Supervised Clustering: A Feature Projection Perspective", In *Proceedings of the 13th International Conference on Knowledge Discovery and Data Mining*, pp.707-716, 2007.

[Hoi 07] Hoi, S. C. H., Jin, R. and Lyu, M. R.: "Learning Nonparametric Kernel Matrices from Pairwise Constraints", In *Proceedings of the 24th International Conference on Machine Learning* , pp.361-368, 2007.

[Li 08] Li, Z., Liu, J. and Tang, X.: "Pairwise Constraint Propagation by Semidefinite Programming for Semi-Supervised Classification", In *Proceedings of the 25th International Conference on Machine Learning* , pp.576-583, 2008.