Clustering with Partition Level Side Information

Hongfu Liu[1] and Yun Fu[1,2]
[1]Department of Electrical and Computer Engineering, Northeastern University, Boston
[2]College of Computer and Information Science, Northeastern University, Boston
liu.hongf@husky.neu.edu, yunfu@ece.neu.edu

***Abstract*—Constrained clustering uses pre-given knowledge to improve the clustering performance. Among existing literature, researchers usually focus on Must-Link and Cannot-Link pairwise constraints. However, pairwise constraints not only disobey the way we make decisions, but also suffer from the vulnerability of noisy constraints and the order of constraints. In light of this, we use partition level side information instead of pairwise constraints to guide the process of clustering. Compared with pairwise constraints, partition level side information keeps the consistency within partial structure and avoids self-contradictory and the impact of constraints order. Generally speaking, only small part of the data instances are given labels by human workers, which are used to supervise the procedure of clustering. Inspired by the success of ensemble clustering, we aim to find a clustering solution which captures the intrinsic structure from the data itself, and agrees with the partition level side information as much as possible. Then we derive the objective function and equivalently transfer it into a K-mean-like optimization problem. Extensive experiments on several real-world datasets demonstrate the effectiveness and efficiency of our method compared to pairwise constrained clustering and consensus clustering, which verifies the superiority of partition level side information to pairwise constraints. Besides, our method has high robustness to noisy side information.**

***Keywords*-Clustering; Partition level side information; K-means; Utility function**

I. Introduction

Cluster analysis is a core technique in machine learning and artificial intelligence [1], [8], [14], [23], [10], [25], which aims to partition the objects into different groups that objects in the same group are more similar to each other than to those in other groups. To further improve the performance, semi-supervised clustering or constrained clustering comes into being, which incorporates pre-known information or side information into the process of clustering.

Since clustering has the property of non-order, the most usual constraints are pairwise constraints. Specifically Must-Link and Cannot-Link constraints represent that two instances should lie in the same cluster or not. At the first thought, it is easy to decide yes or no for pairwise constraints. However, in real-world applications, just given one image of a cat and one image of a dog, it is difficult to answer whether these two images should be in a cluster or not because no decision rule can be summarized only based on two images. Besides, as [24] reported, large disagreements are often observed among human workers in specifying pairwise constraints; for instance, more than 80% of the pairwise labels obtained from human workers are inconsistent with the ground truth for the *Scenes* data set [9]. Moreover, it has been widely recognized [3], [5], [13] that the order of constraints also has great impact on the clustering results, therefore sometimes more constraints even make a detrimental effect.

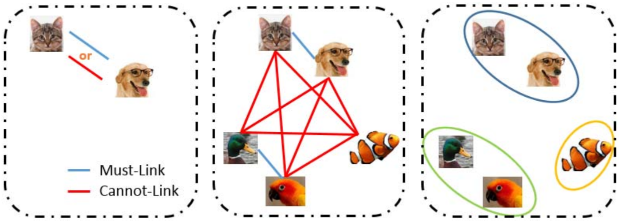
In response to this, we use another constraint, called partition level side information to replace pairwise constraints. Partition level side information also called partial labeling means that only a small portion of data is selected to label from 1 to K. This concept was proposed by [2], which used partition level side information to initialize the centroids for K-means and employed the standard K-means to finish the clustering task; however, it did not involve the side information into the process of clustering. In this paper, we revisit partition level side information and involve it into the process of clustering to obtain the final solution. Inspired by the success of ensemble clustering, we take the partition level side information as a whole and calculate the utility between the learnt clustering solution and partition level side information. We aim to find a clustering result which captures the intrinsic structure from the data itself, and agrees with the partition level side information as much as possible. Based on this, the objective function is derived and we give its corresponding solution by a K-means-like optimization problem with only small modification on the distance function and update rule for centroids. Extensive experiments on several real-world datasets demonstrate the effectiveness and efficiency of our method compared to pairwise constrained clustering and ensemble clustering, which verifies the superiority of partition level side information to pairwise constraints. Besides, our method has high robustness to noisy side information even with 50% noisy side information.

II. Related Work

In this part, we summarize the related work on constrained clustering and ensemble clustering.

K. Wagstaff and C. Cardie first put forward the concept of constrained clustering via incorporating pairwise constraints (Must-Link and Cannot-Link) into a clustering algorithm and modified COBWEB to finish the partition [18]. Later, COP-K-means, a K-means-based algorithm kept all the

1550-4786/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDM.2015.18



(a) One pairwise (b) Multi pairwise (c) Partition level

Figure 1. The comparison between pairwise constraints and partition level side information. In (a), we cannot decide a Must-Link or Cannot-link only based on two instances; compared (b) with (c), it is more natural to label the instances in the partition level way rather than pairwise constraints.

constraints satisfied and attempted to assign each instance to its nearest centroid [19]. [15] developed a framework to involve pre-given knowledge into density estimation with Gaussian Mixture Model and presented a closed form EM procedure and generalized EM procedure for Must-Link and Cannot-Link respectively. However, sometimes satisfying all the constraints as well as the order of constraints makes the clustering intractable and no solution often can be found by returning an empty partition. To overcome such limitation, soft constrained clustering algorithms have been developed to minimize the number of violated constraints. Constrained Vector Quantization Error (CVQE) considered the cost of violating constraints and optimized the cost within the objective function of K-means [5]. Further, LCVQE modified CVQE with different computation of violating constraints [13]. Metric Pairwise Constrained K-means (MPCK-means) employed the constraints to learn a best Mahalanobis distance metric for clustering [3]. Among these K-means-based constrained clustering, [4] presented a thoroughly comparative analysis and found that LCVQE presents better accuracy and violates fewer constraints than CVQE and MPCK-Means.

Another related area is ensemble clustering, which fuses several basic partitions. [16] was the first to propose the ensemble clustering problem, and some graph-based methods [6], co-association matrix based [7] and K-means-based methods [17], [20], [21], [11], [12] are followed to fuse these basic partitions in an efficient way. Here ensemble clustering is applied to fuse the basic partition generated from the data alone and partition level side information. Although there are much work in this area, few of them can handle incomplete partition level side information.

Different from the existing work, we consider a new kind of constraint, called partition level side information. Besides, partition level side information is not affected by the order of constraints. Through extensive experiments, partition level side information shows stronger robustness to noisy constraints than pairwise constraints.

III. Problem Formulation

In this section, we first introduce the definition of partition level side information and discuss the relationship between partition level side information and pairwise constraints. Then based on partition level side information, we give the problem definition and derive the objective function.

A. Partition Level Side Information

Since clustering is a orderless partition, pairwise constraints have been put forward to further improve the performance of clustering for a long time. Specifically Must-Link and Cannot-Link constraints represent that two instances should lie in the same cluster or not. Although in the framework of pairwise constraints we avoid answering the mapping relationship among different clusters and at the first thought it is easy to decide yes or no for pairwise constraints, such pairwise constraints are illogic in essence. For example (See Figure 1), given one image containing a cat and another image containing a dog, the pairwise constraint needs external information, such as human knowledge or expert suggestion, to determine whether these two images are in the same cluster or not. Here comes the first question that what is the cluster. The goal of cluster analysis is to find cluster structure. If we do not know the meaning of clusters, how can we decide the given two images are located in the same cluster or not? That means pairwise constraints request the cluster structure in advance. Someone might argue that experts have their own pre-defined cluster structure, but the matching between pre-defined and true cluster structure also begs questions. The second drawback of pairwise constraints is that we cannot simply say yes or no only based on two instances in practice. Also for the cat and dog images, users might have different decision rules based on different pre-defined cluster structures, such as animal or non-animal, land, water or flying animal and just cat or dog categories. That is to say, without seeing other instances as references, we cannot make any decision to build pairwise constraints. The third drawback is that pairwise constraints disobey the way we make decisions. It is tedious to build an only 100×100 matrix of pairwise constraints. Even though the pairwise constraints matrix is a symmetric matrix and there exists transitivity for must-link and cannot-link constraints, the elements of the pairwise constraints matrix is huge to the number of instances.

To avoid these drawbacks of pairwise constraints, here we propose a new constraint, called partition level side information as follows.

Definition 1. ***Partition Level Side Information.*** *Given a data set containing n instances, randomly select a small portion $p \in (0, 1)$ of the data for a user to label from 1 to K, which is the user-predefined cluster number, then the label information of these np instances is called $p-$partition level side information.*

Different from pairwise constraints, partition level side information groups the given np instances as a whole process. Taking other instances as references, it makes more sense to decide the group label than pairwise constraints. Another benefit is that partition level side information has high consistency, while sometimes pairwise constraints from users might be self-contradictory by transitivity. That is to say, given a $p-$partition level side information, we can build a $np \times np$ pairwise constraints matrix with containing the same information. On the contrary, a $p-$partition level side information cannot be derived by several pairwise constraints. In addition, for human beings it is much easier to separate an amount of instances into different groups, which accords with the way of labeling. As above mentioned, partition level side information has obvious advantages over pairwise constraints, which is a promising candidate for crowd sourcing labeling.

Based on the Def. 1 of partition level side information, we formalize the problem definition: *How to incorporate partition level side information in the process of clustering?*

One naive way to solve the above problem is to transfer the partition level side information into pairwise constraints, then any traditional semi-supervised clustering method can be used to obtain final clustering. However, such solution does not make full use of the consistency of partition level side information. Inspired by the huge success of ensemble clustering, we treat the partition level side information as an integrated one and make the clustering result agree the given partition level side information as much as possible. Specifically speaking, we calculate disagreement between the clustering result and the given partition level side information in a utility view as a penalty term in the objective function of some clustering method. Here we take K-means as the basic clustering method and give its corresponding objective function for partition level side information.

B. Objective Function

Let X be the data matrix with n instances and m features and S be a $np \times K$ side information matrix containing np instances and K clusters, where each row only has one element with value 1 representing the label information and others are all zeros. The objective function of our model is as follows:

$$\begin{aligned} &\min_{H} ||X - HC||_{\text{F}}^2 + \lambda||S - (H \otimes S)G||_{\text{F}}^2, \\ &\text{s.t. } H_{ik} \in \{0, 1\}, \sum_{k=1}^{K} H_{ik} = 1, 1 \leq i \leq n. \end{aligned} \tag{1}$$

where H is the final label matrix, C is the corresponding centroids matrix, $H \otimes S$ is part of H which the instances are also in the side information S, G is a $K \times K$ alignment matrix, λ is a tradeoff parameter to present the confidence degree of the side information and the 1-of-K coding constraints make the final solution a hard partition, which means one instance only belongs to one cluster.

The objective function consists of two parts. One is the standard K-means with squared Euclidean distance, the other is a term measuring the disagreement between the part of H and the side information S. We aim to find a solution H, which not only captures the intrinsic structural information from the original data, but also has as little disagreement as possible with the side information S. Here we introduce G, which plays a role in shuffling the order of clusters in S. It is crucial to align two partitions due to the non-ordering of cluster labels. For instance, the distance between two exact same partitions with different label orders cannot be zero without alignment.

To solve the optimization problem in Eq. 1, we separate the data X and indicator matrix H into two parts, X_1 and X_2, H_1 and H_2, according to side information S, therefore the objective function can be written as:

$$\min_{H_1, H_2} ||X_1 - H_1C||_{\text{F}}^2 + ||X_2 - H_2C||_{\text{F}}^2 + \lambda||S - H_1G||_{\text{F}}^2. \tag{2}$$

IV. Solutions

In this part, we give the corresponding solution to Eq. 2 by equivalently transferring the problem into a K-means-like optimization problem in an efficient way.

A. K-means-like optimization

Although we can use ALM to obtain the solution by taking derivation of each unknown variables, it is not efficient due to some matrix multiply and inverse. Besides if we have multi side information, the data is separated to too many fractured pieces, which is hard to operate in real-world applications. This pushes us to think that can we solve the above problem in a neat mathematical way with high efficiency. In the following, we equivalently transfer the problem into a K-means-like optimization problem via just concatenating the partition level side information with the original data.

First, we introduce the concatenating matrix D,

$$D = \begin{bmatrix} X_1 & S \\ X_2 & 0 \end{bmatrix} \text{ with } D = [D_1 \ D_2], \ D_1 = X \text{ and } D_2 = [S \ 0]^\top,$$

where d_i consists of two parts, one is the original features $d_i^{(1)} =< d_{i,1}, \cdots, d_{i,m} >$, i.e., the first m columns; the other last K columns $d_i^{(2)} =< d_{i,m+1}, \cdots, d_{i,m+K} >$ denotes the side information. Here we can see that D is nothing but a concatenating matrix with the original data X and partition level side information S; for those instances with side information, we just put the side information behind the original features, and for those instances without side information, zeros are used to fill up.

If we just apply K-means on the matrix D, there will be some problems, such as for those instances without side information, all zero values contribute to the computation of the centroids, which inevitably interferes the final cluster structure. Since we make the partition level side information guide the clustering process in a utility way, those all zeros values should not provide any utility to measure the similarity of two partitions. That is to say, the centroids of K-means is no longer the mean of the data instances belonging to certain cluster. Let $m_k = \langle m_k^{(1)}, m_k^{(2)} \rangle$ be the

Algorithm 1 The algorithm of clustering with partition level side information for K-means

```
Require: $X$: data matrix, $n \times m$;
         $K$: number of clusters;
         $S$: $p-$partition level side information, $pn \times K$;
         $\lambda$: trade-off parameter.
Ensure: optimal $H^*$;
Build the concatincating matrix $D$, $n \times (m+K)$;
Randomly select $K$ instances as centroids;
repeat
    Assign each instance to its closest centroid by the distance function in Eq. 5;
    Update centroids by Eq. 3;
until the objective value in Eq. 2 remains unchanged.
```

k-th centroid of K-means, which $m_k^{(1)} = \langle m_{k,1}, \cdots, m_{k,m} \rangle$ and $m_k^{(2)} = \langle m_{k,m+1}, \cdots, m_{k,m+K} \rangle$. We modify the computation of the centroids as follows,

$$m_k^{(1)} = \frac{\sum_{x_i \in \mathbf{C}_k} d_i^{(1)}}{|\mathbf{C}_k|},\ m_k^{(2)} = \frac{\sum_{x_i \in \mathbf{C}_k \bigcap S} d_i^{(2)}}{|\mathbf{C}_k \bigcap S|}. \tag{3}$$

Recall that the standard K-means, the centroids are computed by arithmetic means, whose denominator represents the number of instances in its corresponding cluster. Here in Eq. 3, our centroids have two parts $m_k^{(1)}$ and $m_k^{(2)}$. For $m_{k,1}$, the denominator is also $|\mathbf{C}_k|$; but for $m_{k,2}$, the denominator is $|\mathbf{C}_k \cap S|$. After modifying the computation of centroids, we have the following the Theorem 1.

Theorem 1. *Given the data matrix X, side information S and augmented matrix D, we have*

$$\begin{aligned} &\min_H ||X - HC||_{\text{F}}^2 + \lambda ||S - (H \otimes S)G||_{\text{F}}^2 \\ &\Leftrightarrow \min \sum_{k=1}^{K} \sum_{d_i \in \mathbf{C}_k} f(d_i, m_k), \end{aligned} \tag{4}$$

where d_i is a $1 \times (m+K)$ vector representing the i-th row of D, m_k is the k-th centroid calculated by Eq. 3 and the distance function f can be computed by

$$f(d_i, m_k) = ||d_i^{(1)} - m_k^{(1)}||_2^2 + \lambda I(d_i \in S)||d_i^{(2)} - m_k^{(2)}||_2^2, \tag{5}$$

where $I(d_i \in S) = 1$ means the side information contains x_i, and 0 otherwise.

Proof: According to the objective function, we have

$$\begin{aligned} &\sum_{k=1}^{K} \sum_{d_i \in \mathbf{C}_k} f(d_i, m_k) \\ &= \sum_{k=1}^{K} \sum_{d_i \in \mathbf{C}_k} (||d_i^{(1)} - m_k^{(1)}||_2^2 + \lambda I(x_i \in S)||d_i^{(2)} - m_k^{(2)}||_2^2) \\ &= \sum_{k=1}^{K} \sum_{d_i \in \mathbf{C}_k \cap S} (||d_i^{(1)} - m_k^{(1)}||_2^2 + \lambda ||d_i^{(2)} - m_k^{(2)}||_2^2) \\ &\quad + \sum_{k=1}^{K} \sum_{d_i \in \mathbf{C}_k \cap \overline{S}} ||d_i^{(1)} - m_k^{(1)}||_2^2 \\ &= ||X_1 - H_1 C||_{\text{F}}^2 + \lambda ||S - H_1 G||_{\text{F}}^2 + ||X_2 - H_2 C||_{\text{F}}^2. \end{aligned} \tag{6}$$

According to Eq. 2, we finish the proof. ■

Table I
EXPERIMENTAL DATA SETS

Data set	#Instances	#Features	#Classes	CV
$breast$	699	9	2	0.4390
$ecoli^*$	332	7	6	0.8986
$glass$	214	9	6	0.8339
$iris$	150	4	3	0.0000
$pendigits$	10992	16	10	0.0422
$satimage$	4435	36	6	0.4255
$wine^+$	178	13	3	0.1939

*: two clusters containing only two objects are deleted as noise.
$^+$: the last attribute is normalized by a scaling factor 1000.

Remark 1. *Taking a close look at the concatenating matrix D, the side information can be regarded as new features with more weights, which is controlled by λ. Besides, Theorem 1 provides a way to clustering with both numeric and categorical features together, which means we calculate the difference between the numeric and categorical part of two instances separately and add them together.*

Remark 2. *Different from standard K-means, the distance function in Theorem 1 is a linear combination of two squared Euclidean distances. Moreover, for some instances the original features and side information jointly contribute to the distance and for some instances only the original features decide which cluster the instance should belong to.*

By Theorem 1, we transfer the problem into a K-means-like clustering problem. Since the update rule and distance function have changed, it is necessary to verify the convergency of the K-means-like algorithm.

Theorem 2. *For the objective function in Theorem 1, the optimization problem is guaranteed to converge in finite two-phase iterations of K-means-like optimization problem.*

Here we omit the proof of Theorem 2 due to the limited pages. The proposed algorithm is summarized in Alg. 1.

V. EXPERIMENTAL RESULTS

In this section, we present the experimental results of the clustering with partition level side information compared to pairwise constrained methods clustering and ensemble clustering method. Generally speaking, we first demonstrate the advantages of our method in terms of effectiveness and efficiency. Then we add noises with different ratios to analyse the robustness of our method.

A. Experimental Setup

Experimental data. 7 datasets from UCI repositories[1] are used for evaluating the performance of the proposed method. The basic statistical information is reported in Table I including the number of instances, features and classes and the coefficient of variation of the classes.

Tools. We select two competitive methods for comparison. One is LCVEQ [13], a K-means-based pairwise constraint clustering method; the second is K-means-based Consensus

[1] https://archive.ics.uci.edu/ml/datasets.html

Table II
CLUSTERING PERFORMANCE ON SEVEN REAL DATASETS BY NMI(%)

Data Sets	percent	Ours	LCVQE	KCC
$breast$ 73.61 ± 0.00	10%	**75.91 ± 1.37**	75.88 ± 1.38	75.74 ± 1.22
	20%	**78.20 ± 1.85**	78.15 ± 1.86	77.59 ± 1.48
	30%	**80.71 ± 2.14**	80.59 ± 2.12	80.01 ± 1.98
	40%	**83.20 ± 1.96**	81.56 ± 11.29	82.46 ± 1.86
	50%	**85.38 ± 1.86**	81.96 ± 16.56	84.58 ± 1.82
$ecoli$ 0.53 ± 2.53	10%	**64.16 ± 2.31**	60.87 ± 3.32	59.57 ± 5.22
	20%	**68.20 ± 2.98**	63.24 ± 4.71	60.56 ± 5.11
	30%	**73.21 ± 2.74**	67.82 ± 4.56	62.89 ± 6.21
	40%	**76.92 ± 2.84**	70.46 ± 4.54	65.04 ± 4.84
	50%	**80.84 ± 2.72**	72.83 ± 5.33	69.57 ± 6.11
$glass$ 38.46 ± 3.61	10%	37.49 ± 2.92	37.44 ± 3.47	**38.72 ± 3.33**
	20%	**39.73 ± 2.70**	35.95 ± 3.73	38.42 ± 3.14
	30%	**42.51 ± 2.96**	34.66 ± 4.57	39.05 ± 3.06
	40%	**47.16 ± 3.37**	34.05 ± 3.45	38.61 ± 3.24
	50%	**52.01 ± 2.82**	32.08 ± 5.27	38.16 ± 4.15
$iris$ 72.44 ± 6.82	10%	**76.53 ± 1.77**	75.97 ± 3.41	72.58 ± 9.29
	20%	**78.46 ± 2.41**	78.29 ± 2.71	72.17 ± 11.65
	30%	**81.05 ± 2.79**	80.96 ± 3.47	76.37 ± 9.61
	40%	**83.66 ± 2.83**	83.03 ± 6.08	79.93 ± 7.27
	50%	**85.41 ± 3.03**	85.02 ± 3.88	81.78 ± 6.70
$pendigits$ 68.22 ± 1.48	10%	**68.61 ± 0.52**	66.72 ± 1.20	65.31 ± 2.61
	20%	**68.93 ± 0.48**	63.13 ± 2.31	66.73 ± 3.92
	30%	**69.99 ± 0.59**	59.84 ± 2.51	68.58 ± 1.64
	40%	68.25 ± 0.02	57.86 ± 2.16	**75.35 ± 3.06**
	50%	68.38 ± 0.34	54.06 ± 2.42	**78.82 ± 3.06**
$satimage$ 57.52 ± 5.88	10%	**61.40 ± 0.05**	54.56 ± 5.15	54.84 ± 7.24
	20%	**61.43 ± 0.06**	52.63 ± 8.86	60.28 ± 4.98
	30%	**61.49 ± 0.05**	51.33 ± 10.65	58.07 ± 6.79
	40%	61.53 ± 0.04	44.46 ± 10.25	**64.30 ± 4.47**
	50%	61.61 ± 0.08	45.05 ± 11.93	**68.96 ± 5.21**
$wine$ 13.07 ± 0.87	10%	**29.44 ± 5.32**	26.97 ± 5.92	27.27 ± 5.52
	20%	**34.63 ± 5.05**	25.54 ± 7.71	29.93 ± 5.65
	30%	**37.74 ± 4.82**	23.39 ± 8.28	33.62 ± 5.27
	40%	**43.10 ± 3.45**	19.81 ± 10.76	37.15 ± 5.32
	50%	**46.36 ± 3.55**	19.60 ± 13.34	43.60 ± 5.31

Clustering (KCC) [21], which first generates one basic partition alone from the data and then fuse this partition with incomplete partition level side information. In our method, we empirically set λ to 100, and we also set the weight between side information and basic partition in KCC as $\lambda : 1$. In the experiments, we randomly select certain percent partition level side information from the ground truth for our method and KCC, then transfer the partition level side information into pairwise constraints for LCVQE. Note that the number of clusters for three algorithms is set to the number of true clusters.

Validation measure. Since the class labels are provided for each data set, the Normalized Mutual Information (NMI) is used to measure the clustering performance [22].

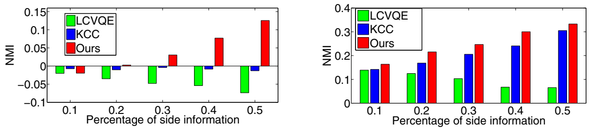
B. Effectiveness and Efficiency

Table II shows the clustering performance of different algorithms on all the seven data sets with side information of different ratios. The first column represents the results without constraints by K-means. In each scenario, 50 runs with different side information are conducted and the average performance as well as the standard deviation are reported.

Our method achieves the best performance in most scenarios except on *pendigits* and *satimage* with 40% and 50% percent side information. If we take a close look at Table II, our method and KCC keep consistently increasing performance as the percent of side information. LCVQE gets reasonable results on the well separated data sets *breast* and *iris*; however, it is surprising that LCVQE gets much worse results with more guidance on *glass*, *pendigits*, *satimage* and *wine*. This might result from the great impact of the order of pairwise constraints, which leads to the deformity of clustering structure and the worse solutions even than the basic K-means without any side information. In addition, our method enjoys better stability than LCVQE and KCC. For instance, LCVQE has up to 17.5% standard deviation on *breast* with 50% side information and the volatility of KCC on *iris* with 20% side information goes up to 16.7%. Fig. 2 shows the improvement of constrained clustering algorithms over the baseline methods on *glass* and *wine*. We can see that in most scenarios, the performance of our method shows a positive relevance with the percentage of side information, which demonstrates the effectiveness of partition level side information. Although we equivalently transfer the partition level side information into pairwise constraints, our clustering method based on partition level side information utilizes the consistency within the side information and achieves better results.

Table III
COMPARISON OF EXECUTION TIME (IN SECONDS)

Data Sets	$breast$	$ecoli$	$glass$	$iris$	$pend.$	$sati.$	$wine$
Ours	**0.0014**	**0.0117**	**0.0052**	**0.0019**	**0.4538**	**0.1887**	**0.0094**
LCVQE	0.0461	0.0318	0.0256	0.0097	76.7346	11.5499	0.0126
KCC	0.2638	0.2175	0.1263	0.0673	4.9807	1.7020	0.1030

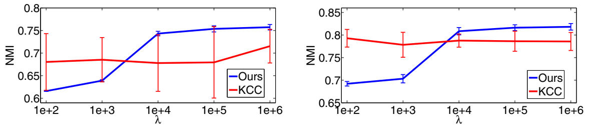


(a) *glass* (b) *wine*

Figure 2. Improvement of constrained clustering on *glass* and *wine*

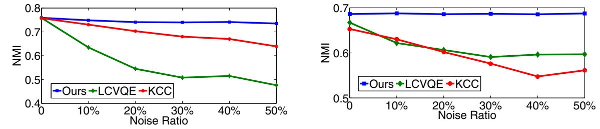
Next, we evaluate the efficiency of three algorithms. Table III shows the average of execution time of different algorithms with 10% side information. We can see that our method shows obvious advantages than other algorithms. On *pendigits*, our method is faster 10 times than KCC, and nearly 170 times than LCVQE. Taking the effectiveness and efficiency into account, our method not only achieves satisfactory result, but also has high efficiency, which verifies that it is suitable for large data set clustering with partition level side information. In the following, we use our K-means-based method as default to further explore its characteristics.

So far, we use a fixed λ to evaluate the clustering performance, and on *pendigits* and *satimage* with 50% side information, our method has a large gap with KCC. In the following, we explore the impact of λ on these two data sets. As can be seen in Fig. 3 with λ varying from $1e + 2$ to $1e + 6$, KCC keeps stable results with the change of λ, but suffers from heavy volatility. The performance of our method continuously goes up as λ increases with high robustness; besides, our method achieves stability when λ is larger than a threshold, like $1e + 4$. Compared to other



(a) *satimage* (b) *pendigits*

Figure 3. Impact of λ on *satimage* and *pendigits*



(a) *breast* (b) *pendigits*

Figure 4. Impact of noisy side information on *breast* and *pendigits*

data sets, *pendigits* and *satimage* have more features so that a larger λ might help to improve the performance.

C. Handling Side Information with Noises

In real-world application, the part of side information might be noisy and misleading, thus we validate our method with noisy side information. Here fixing 10% side information, we randomly select certain instances from the side information and randomly label them as noises. In Fig. 4, we can see that the performance of LCVQE and KCC drops sharply with the increasing of noise ratio; even 10% noise ratio does great harm to LCVQE on *breast*. Misleading pairwise constraints and large weights of the noisy side information lead to corrupted results. On the contrary, our method performs high robustness even when the noise ratio is up to 50%. It demonstrates that we do not need exact side information from the specialists, instead someone only knows a bit also helps to improve the clustering results, which validates the effectiveness of our method in practice with noisy side information .

VI. Conclusion

In this paper, we propose a method for clustering with partition level side information. Different from pairwise constraints, partition level side information accords with the labeling from human being with other instances as references. Based on this, we formulate the problem via conducting clustering and making the structure agree as much as possible with side information. Then we equivalently transfer it into K-means clustering, which can be solved with high efficiency. Extensive experiments demonstrate the effectiveness and efficiency of our methods compared to two state-of-the-art algorithms. Besides, our method has high robustness when it comes to noisy side information.

Acknowledgment

This research is supported in part by the NSF CNS award 1314484, ONR award N00014-12-1-1028, ONR Young Investigator Award N00014-14-1-0484, NPS award N00244-15-1-0041 and U.S. Army Research Office Young Investigator Award W911NF-14-1-0218.

References

[1] C. Aggarwal and C. Reddy. *Data clustering: algorithms and applications*. CRC Press, 2013.

[2] S. Basu. Semi-supervised clustering: Learning with limited user feedback. *Doctoral dissertation*, 2003.

[3] M. Bilenko, S. Basu, and R. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of ICML*, 2004.

[4] T. Covoes, E. Hruschka, and J. Ghosh. A study of k-means-based algorithms for constrained clustering. *Intelligent Data Analysis*, 17(3):485–505, 2013.

[5] I. Davidson and S. Ravi. Clustering with constraints: Feasibility issues and the k-means algorithm. In *Proceedings of SDM*, 2005.

[6] X. Fern and C. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of ICML*, 2004.

[7] A. Fred and A. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005.

[8] A. Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[9] F. Li and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of CVPR*, 2005.

[10] S. Li and Y. Fu. Learning balanced and unbalanced graphs via low-rank coding. *IEEE Transactions Knowledge and Data Engineering*, 2015.

[11] H. Liu, T. Liu, J. Wu, D. Tao, and Y. Fu. Spectral ensemble clustering. In *Proceedings of KDD*, 2015.

[12] H. Liu, J. Wu, D. Tao, Y. Zhang, and Y. Fu. Dias: A disassemble-assemble framework for highly sparse text clustering. In *Proceedings of SDM*, 2015.

[13] D. Pelleg and D.Baras. K-means with large and noisy constraint sets. In *Proceedings of ECML*, 2007.

[14] M. Shao, L. S, Z. Ding, and Y. Fu. Deep linear coding for fast graph clustering. In *Proceedings of IJCAI*, 2015.

[15] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing gaussian mixture models with em using equivalence constraints. In *NIPS*, 2004.

[16] A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:587–617, 2003.

[17] A. Topchy, A. Jain, and W. Punch. Combining multiple weak clusterings. In *Proceedings of ICDM*, 2003.

[18] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *AAAI/IAAI*, page 109, 2000.

[19] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrdl. Constrained k-means clustering with background knowledge. In *Proceedings of ICML*, pages 577–584, 2001.

[20] J. Wu, H. Liu, H. Xiong, and J. Cao. A theoretic framework of k-means-based consensus clustering. In *Proceedings of IJCAI*, pages 1799–1805, 2013.

[21] J. Wu, H. Liu, H. Xiong, J. Cao, and J. Chen. K-means-based consensus clustering: A unified view. *IEEE Transactions on Knowledge and Data Engineering*, 27(1):155–169, 2015.

[22] J. Wu, H. Xiong, and J. Chen. Adapting the right measures for k-means clustering. In *Proceedings of KDD*, 2009.

[23] X. Wu, V. Kumar, J. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. McLachlan, A. Ng, B. Liu, P. Yu, Z. Zhou, M. Steinbach, D. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

[24] J. Yi, R. Jin, S. Jain, T. Yang, and A. Jain. Semi-crowdsourced clustering: Generalizing crowd labeling by robust distance metric learning. In *NIPS*, 2012.

[25] H. Zhao, Z. Ding, and Y. Fu. Block-wise constrained sparse graph for face image representation. In *Proceedings of FG*, 2015.