Partition Level Constrained Clustering

Hongfu Liu, *Student Member, IEEE*, Zhiqiang Tao and Yun Fu *Senior Member, IEEE*

Abstract—Constrained clustering uses pre-given knowledge to improve the clustering performance. Here we use a new constraint called partition level side information and propose the Partition Level Constrained Clustering (PLCC) framework, where only a small proportion of the data is given labels to guide the procedure of clustering. Our goal is to find a partition which captures the intrinsic structure from the data itself, and also agrees with the partition level side information. Then we derive the algorithm of partition level side information based on K-means and give its corresponding solution. Further, we extend it to handle multiple side information and design the algorithm of partition level side information for spectral clustering. Extensive experiments demonstrate the effectiveness and efficiency of our method compared to pairwise constrained clustering and ensemble clustering methods, even in the inconsistent cluster number setting, which verifies the superiority of partition level side information to pairwise constraints. Besides, our method has high robustness to noisy side information, and we also validate the performance of our method with multiple side information. Finally, the image cosegmentation application based on saliency-guided side information demonstrates the effectiveness of PLCC as a flexible framework in different domains, even with the unsupervised side information.

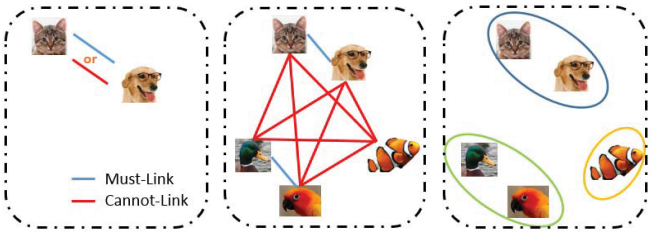
Index Terms—Constrained Clustering, Utility Function, Partition Level, Cosegmentation.

1 Introduction

Cluster analysis is a core technique in machine learning and artificial intelligence [1], [2], [3], 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. It has been widely used in many domains, such as search engines [4], recommend systems [5] and image segmentation [6]. In light of this, many algorithms have been proposed to thrive this area, such as connectivity-based clustering [7], centroid-based clustering [8] and density-based clustering [9]; however, the results of clustering still exist large gaps with the results of classification. To further improve the performance, constrained clustering comes into being, which incorporates pre-known or side information into the process of clustering.

Since clustering has the property of non-order, the most common constraints are pairwise. Specifically, Must-Link and Cannot-Link constraints represent that two instances should lie in the same cluster or not [10], [11]. At the first thought, it is easy to decide Must-Link or Cannot-Link for pairwise comparison. However, in real-world applications, just given one image of a cat and one image of a dog (See Fig. 1), it is difficult to answer whether these two images should be in a cluster or not because no decision rule can be made based on only two images. Without other objects as references, it is highly risky to determine whether the data set is about cat-and-dog or animals-and-non-animals. Besides, as [12] 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 [13]. Moreover, it has been widely recognized that the order of constraints also has great impact on the clustering results [14], therefore sometimes more constraints even make a detrimental effect. Although some methods such as soft constraints [15], [16] are put forward to handle these challenges, the results are still far from satisfaction.

In response to this, we use partition level side information to overcome these limitations of pairwise constraints. Partition level side information also called partial labeling means that only a small portion of data is labeled into different clusters. Compared with pairwise constraints, partition level side information has the following benefits: (1) it is more natural to organize the data in a higher level than pairwise comparisons, (2) when human workers label one instance, other instances provide enough information as reference for a good decision, (3) it is immune to the self-contradiction and the order of pairwise constraints. The concept of partition level side information was proposed by [17], which aims to find better initialization centroids and employs the standard K-means to finish the clustering task; since the partition level side information is only used to initialize the centroids without involving it into the process of clustering, this method does not belong to the constrained clustering area. In this paper, we revisit partition level side information and involve it into the process of clustering to obtain the final solution in a one-step framework. Inspired by the success of ensemble clustering [18], we take the partition level side information as a whole and calculate the similarity between the learnt clustering solution and the given side information. We propose the Partition Level Constrained Clustering (PLCC) framework, which not only captures the intrinsic structure from the data itself, but also agrees with the



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

Fig. 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 well-organised way, such as partition level rather than pairwise constraint.

Manuscript received XXX; revised XXX.

partition level side information as much as possible. Based on K-means clustering, we derive the objective function and give its corresponding solution via derivation. Further, the above solution can be equivalently transformed into a K-means-like optimization problem with only small modification on the distance function and update rule for centroids. Thus, a roughly linear time complexity can be guaranteed. Moreover, we extend it to handle multiple side information and provide the algorithm of partition level side information for spectral clustering. Extensive experiments on several real-world datasets demonstrate the effectiveness and efficiency of our method compared to pairwise constrained clustering and ensemble clustering, even in the inconsistent cluster number setting, which verifies the superiority of partition level side information for the clustering task. Besides, our K-means-based method has high robustness to noisy side information even with 50% noisy side information. And we validate the performance of our method with multiple side information, which makes it a promising candidate for crowdsourcing. Finally, a totally unsupervised framework called Saliency-Guided Constrained Clustering (SG-PLCC) is put forward for the image cosegmentation task, which demonstrates the effectiveness and flexibility of PLCC in different domains. Our main contributions are highlighted as follows.

- We revisit partition level side information and incorporate it to guide the process of clustering and propose the Partition Level Constrained Clustering framework.
- Within the PLCC framework, we propose a K-means-like algorithm to solve the clustering with partition level side information in a high efficient way and extend our model to multiple side information and spectral clustering.
- Extensive experiments demonstrate our algorithm not only has promising performance compared to the state-of-the-art methods, but also performs high robustness to noisy side information.
- A cosegmentation application with saliency prior is employed to further illustrate the flexibility of PLCC. Although only the raw features are extracted and K-means clustering is conducted, we still achieve promising results compared with several cosegmentation algorithms.

This paper is an extension of our conference paper [19] with the following new contents: (1) The complete basic solution for the K-means-based PLCC is provided; (2) we extend the PLCC framework in the multiple partition level side information and evaluate the performance with noisy prior knowledge; (3) the spectral-based PLCC is proposed with the corresponding solution and analysis; (4) two clustering validity metrics and more datasets are used to fully evaluate the performance of the proposed methods; (5) we evaluate the performance of PLCC in the inconsistent cluster number setting and (6) we employ PLCC on cosegmentation to demonstrate the effectiveness in image domain.

This paper is organized as follows. Section 2 introduces the related work; then we illustrate the problem definition, derive objective function and provide the corresponding solutions in Section 3. Several extensions are given in Section 5. Section 6 demonstrates our experimental results on several benchmark datasets and the cosegmentation application is given in Section 7. Finally we draw the conclusion in Section 8.

2 Related Work

In this part, we summarize the work related to our paper. Although partition level side information can also be used for semi-supervised classification, it is unfair to compare the clustering results to classification. Thus, we focus on the clustering scenario. In the following, the works on constrained clustering and ensemble clustering are briefly discussed.

2.1 Constrained 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 [10]. Later, COP-K-means, a K-means-based algorithm kept all the constraints satisfied and attempted to assign each instance to its nearest centroid [11]. [20] 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. These algorithms can be regarded as hard constrained clustering since they do not allow any violation of the constraints in the process of clustering. However, sometimes satisfying all the constraints as well as the order of constraints make the clustering intractable and no solution often can be found.

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 [14]. Further, LCVQE modified CVQE with different computation of violating constraints [15]. Metric Pairwise Constrained K-means (MPCK-means) employed the constraints to learn a best Mahalanobis distance metric for clustering [16]. Among these K-means-based constrained clustering, [21] presented a thoroughly comparative analysis and found that LCVQE presents better accuracy and violates less constraints than CVQE and MPCK-Means. It is worthy to note that an NMF-based method also incorporates the partition level side information for constrained clustering [22], which requires that the data points sharing the same label have the same coordinate in the new representation space.

Another category of constrained clustering is to incorporate constraints into spectral clustering, which can be roughly generalized into two groups. The first group directly modifies the Laplacian graph. Kamvar et al. proposed the spectral learning method which set the entry to 1 or 0 according to Must-link and Cannot-link constraints and employed the traditional spectral clustering to obtain the final solution [23]. Similarly, Xu et al. used the similar way to modify the graph and applied random walk for clustering [24]. Lu et al. propagated the constrains in the affinity matrix [25]. [26] and [27] combined the constraint matrix as a regularizer to modify the affinity matrix. The second group modifies the eigenspace instead. [28] altered the eigenspace according to the hard or soft constraints. Li et al. enforced constraints by regularizing the spectral embedding [29]. Recently, [30] proposed a flexible constrained spectral clustering to encode the constraints as part of a constrained optimization problem.

2.2 Ensemble Clustering

Since we employ the utility function in ensemble clustering area to measure the similarity between the learnt clustering and the partition side information, in the following we briefly introduce some key works in this area. A bunch of basic partitions can be generated by different clustering algorithms or the same algorithm

with multiple runs or different parameters. Ensemble clustering aims to fuse these existing basic partitions into a consensus one, which can be generalized into two categories.

The first category measures the similarity in the instance-level by transforming the ensemble clustering into the traditional graph partition. In such kind of methods, the co-association matrix is used to summarize how many times a pair of instances jointly occur in the same clustering, which can be regarded as a new similarity metric in the instance-level. This leads that the traditional graph partition methods can be directly conducted on the co-association matrix without modification. [31] transformed the set of basic partitions into a hypergraph representation and proposed three graph partition methods CSPA, HGPA and MCLA. [32] used the agglomerative hierarchical clustering on the co-association matrix to find the final clustering. Similarly, Liu et al. [18], [33] employed the spectral clustering on the co-association matrix and solved it via weighted K-means. Other methods include Locally Adaptive Cluster based methods [35], Relabeling and Voting [34], genetic algorithm based methods [36], and still many more.

Another kind of methods employs utility functions to measure the similarity between the consensus clustering and basic ones in the partition-level. The consensus partition can be achieved by maximizing the utility functions. Based on quadratic mutual information, [37] solved the consensus clustering with a K-means clustering. And then they extended their work to using the EM algorithm with a finite mixture of multinomial distributions for consensus clustering [39]. To further explore the K-means-based method, Wu et al. gave the sufficient and necessary condition for KCC utility functions and put forward a theoretic framework for K-means-based Consensus Clustering (KCC) [40], [41], followed by text clustering [42], Entropy-based Consensus Clustering [43] and Infinite Ensemble Clustering [44], [45]. It is worthy to note that [18] built a bridge between these two kinds of methods and showed the similarity in the instance-level and partition-level can be convertible.

Different from the existing work, we consider a new kind of constraint, called partition level side information, which much more obeys the way human being makes decision than pairwise constraints. Besides, partition level side information is not affected by the order of constraints. In this paper, we incorporate such constraints into the process of clustering and propose the Partition Level Constrained Clustering (PLCC) framework. Then several efficient algorithms based on K-means and spectral clustering are derived. Through extensive experiments and the cosegmentation application, PLCC shows significant advantages compared with several state-of-the-art methods.

3 Problem Formulation

In this section, we first give the definition of partition level side information and uncover the relationship between partition level side information, pairwise constraints and ground truth labels. Then based on partition level side information, we give the problem definition, build the model and derive its corresponding solution; further an equivalent solution is designed by modified K-means in an efficient way. Finally, the model is extended to handle multiple side information.

3.1 Partition Level Side Information

Since clustering is an orderless partition, pairwise constraints are put forward to further improve the performance of clustering for long time. Specifically, Must-Link and Cannot-Link constraints represent that two instances should lie in the same cluster or not. Although within the framework of pairwise constraints we avoid to answer the mapping relationship among different clusters and at the first thought it is easy to make the Must-Link or Cannot-Link decision for pairwise constraints, such pairwise constraints are illogic in essence. For example (See Figure 1), given one pair images of a cat and a dog, it cannot be directly determined whether these two images are in the same cluster or not without external information, such as human knowledge or expert suggestion. Here comes the first question that what is the cluster. The goal of cluster analysis is to find cluster structure. Only after clustering, we can summarize the meaning for each cluster. If we already know the meaning of each cluster, the problem becomes the classification problem, rather than clustering. Given that we do not know the meaning of clusters in advance, it is highly risky to make the pairwise constraints. 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. Take Fig. 1 as an example. 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, the decisions we make based on two instances suffer from high risk. More importantly, pairwise constraints disobey the way we make decisions. The data should be organized in a higher level rather than pairwise comparisons. Besides, it is tedious to build a pairwise constraint matrix with only 100 instances. Even though the pairwise constraints matrix is a symmetric matrix and there exists transitivity for Must-Link and Cannot-Link constraints, the size of elements of the pairwise constraints matrix is relatively huge to the number of instances.

To avoid these drawbacks of pairwise constraints, here we leverage a new constraint for clustering, 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 to label from 1 to K, which is the user-predefined cluster number, then the label information for only small portion of the data 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 labels 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 an $np \times np$ 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 also a promising candidate for crowd sourcing labeling.

It is also worth illustrating the difference between partition level side information and ground truth. The partition level side information is still an orderless partition. However if we exchange the labels of ground truth, they become wrong labels. Another point is that partition level side information coming from users

TABLE 1
Contingency Matrix

		S			
		$C_1^{(S)}$	$C_2^{(S)}$	$\cdots$	$C_K^{(S)}$	$\sum$
H_1	C_1	$n_{11}^{(S)}$	$n_{12}^{(S)}$	$\cdots$	$n_{1K}^{(S)}$	n_{1+}
	C_2	$n_{21}^{(S)}$	$n_{22}^{(S)}$	$\cdots$	$n_{2K}^{(S)}$	n_{2+}
	.	.	.	$\cdots$	.	.
	C_K	$n_{K1}^{(S)}$	$n_{K2}^{(S)}$	$\cdots$	$n_{KK}^{(S)}$	n_{K+}
	$\sum$	$n_{+1}^{(S)}$	$n_{+2}^{(S)}$	$\cdots$	$n_{+K}^{(S)}$	np

might have different cluster numbers, even suffer from noisy and wrong decision makings. Besides partition level side information comes from multi-users, which might be different from each other, while the ground truth is unique. Especially in the labeling task, the partial labeled data might have the fewer cluster number than the one of the whole data. In this case, we cannot transform the constrained clustering problem into the traditional classification problem.

3.2 Problem Definition

Based on the Definition 1 of partition level side information, we formalize the problem definition: *How to utilize partition level side information to better conduct clustering?*

This problem is totally new to the clustering area. To solve this problem, we have to handle the following challenges:

- How to fuse partition level side information into the process of clustering?
- What is the best mapping relationship between partition level side information and the cluster structure learned from the data?
- How to handle multi-source partition level side information to guide the generation of clustering?

One intuitive way to solve the above problem is to transform 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 advantages 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 with the given partition level side information as much as possible. Specifically, we calculate disagreement between the clustering result and the given partition level side information from a utility view. Here we take K-means as the basic clustering method and give its corresponding objective function for partition level side information in the following.

3.3 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:

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

TABLE 2
Notations

Notation	Domain	Decription
n	$\mathcal{R}$	Number of instances
m	$\mathcal{R}$	Number of features
K	$\mathcal{R}$	Number of clusters
p	$\mathcal{R}$	Percentage of labeled data
X	$\mathcal{R}^{n\times m}$	Data matrix
S	$\{0,1\}^{np\times K'}$	Partition level side information
H	$\{0,1\}^{n\times K}$	Indicator matrix
C	$\mathcal{R}^{K\times m}$	Centroid matrix
G	$\mathcal{R}^{K\times K'}$	Alignment matrix
W	$\mathcal{R}^{n\times n}$	Affinity matrix
D	$\mathcal{R}^{n\times n}$	Diagonal summation matrix
U	$\mathcal{R}^{n\times K}$	Scaled indicator matrix

where H is the indicator matrix, C is the centroids matrix, $H \otimes S$ is part of H where the instances are also in the side information S, U_c is the well-known categorical utility function [38], λ is a tradeoff parameter to present the confidence degree of the side information and the 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 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.

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:

$$\min_{H_1,H_2} ||X_1 - H_1C||_{\mathrm{F}}^2 + ||X_2 - H_2C||_{\mathrm{F}}^2 - \lambda U_c(H_1, S). \tag{2}$$

To better understand the last term in Eq. 2, we introduce the contingency table. In Table 1, given two partitions S and H_1 containing both K clusters (In practice, the partition level side information might have different cluster number from the true cluster number). Let $n_{kj}^{(S)}$ denote the number of data objects belonging to both cluster $C_j^{(S)}$ in S and cluster C_k in H_1, $n_{k+} = \sum_{j=1}^{K} n_{kj}^{(S)}$, and $n_{+j}^{(S)} = \sum_{k=1}^{K} n_{kj}^{(S)}$, $1 \le j \le K$, $1 \le k \le K$. Let $p_{kj}^{(S)} = n_{kj}^{(S)}/np$, $p_{k+} = n_{k+}/np$, and $p_{+j}^{(S)} = n_{+j}^{(S)}/np$. We then have a normalized contingency matrix (NCM), based on which a wide range of utility functions can be accordingly defined. For instance, the widely used category utility function [38] can be computed as follows:

$$U_c(H_1, S) = \sum_{k=1}^{K} p_{k+} \sum_{j=1}^{K} (\frac{p_{kj}^{(S)}}{p_{k+}})^2 - \sum_{j=1}^{K} (p_{+j}^{(S)})^2. \tag{3}$$

The category utility function measures the similarity of two partitions in partition level, which means that two partitions share more similarity with a higher U_c value. Note that the last term in Eq. 3 is a constant when the side information S is given. Back to our problem, the last term in Eq. 2 has the opposite function to measure the dissimilarity of two partitions in a utility way. We have the following Lemma 1 to illustrate this kind equivalent relationship.

Lemma 1. *Given one partition S, a cluster indicator matrix both containing np instances, we have*

$$\max_{H_1} U_c(H_1, S) \Leftrightarrow \min_{H_1} ||S - H_1 G||_{\mathrm{F}}^2, \tag{4}$$

where $G_k = (\frac{p_{k1}^{(S)}}{p_{k+}}, \cdots, \frac{p_{kK}^{(S)}}{p_{k+}})$ is the $k-$th row of $G, \forall k$.

The proof of Lemma 1 can be found in our previous work [40], [41]. Actually the equivalent relationship between $||S - H_1 G||_{\mathrm{F}}^2$ and $U_c(H_1, S)$ holds not only for the optimal H_1, but also holds for any H_1, since $||S - H_1 G||_{\mathrm{F}}^2 + U_c(H_1, S) =$ constant with given S. Lemma 1 introduces one extra variables G to capture the mapping relationship between S to H_1. After aligning S to H_1 with G, the Frobenius norm is to calculate for the dissimilarity between S and H_1 in an efficient way. This gives us a new insight of the objective function in Eq. 1 as follows.

$$\min_{H_1, H_2, C, G} ||X_1 - H_1 C||_{\mathrm{F}}^2 + ||X_2 - H_2 C||_{\mathrm{F}}^2 + ||S - H_1 G||_{\mathrm{F}}^2. \tag{5}$$

4 Solutions

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

4.1 Algorithm Derivation

To derive the algorithm solving Eq. 2, we rewrite Eq. 2 as

$$\begin{aligned} J = \min_{H_1, H_2, C, G} & \mathrm{tr}((X_1 - H_1 C)(X_1 - H_1 C)^\top \\ & + (X_2 - H_2 C)(X_2 - H_2 C)^\top \\ & + \lambda (S - H_1 G)(S - H_1 G)^\top), \end{aligned} \tag{6}$$

where $\mathrm{tr}(\cdot)$ means the trace of a matrix. By this means, we can update H_1, H_2, C and G in an iterative update procedure.

Fixing H_1, H_2, G, Update C. Let $J_1 = ||X_1 - H_1 C||_{\mathrm{F}}^2 + ||X_2 - H_2 C||_{\mathrm{F}}^2$, we have

$$J_1 = \mathrm{tr}((X_1 - H_1 C)(X_1 - H_1 C)^\top + (X_2 - H_2 C)(X_2 - H_2 C)^\top). \tag{7}$$

Then taking derivative of C and setting it as 0, we get

$$\frac{\partial J_1}{\partial C} = -2 H_1^\top X_1 + 2 H_1^\top H_1 C - 2 H_2^\top X_2 + 2 H_2^\top H_2 C = 0. \tag{8}$$

Therefore, we can update C as follows:

$$C = (H_1^\top H_1 + H_2^\top H_2)^{-1} (H_1^\top X_1 + H_2^\top X_2). \tag{9}$$

Fixing H_1, H_2, C, Update G. The term related to G is $||S - H_1 G||_{\mathrm{F}}^2$, then minimize $J_2 = ||S - H_1 G||_{\mathrm{F}}^2$ over G, we have

$$J_2 = \mathrm{tr}((S - H_1 G)(S - H_1 G)^\top). \tag{10}$$

Next we take the derivative of J_2 over G, and have

$$\frac{\partial J_2}{\partial G} = -2 H_1^\top S + 2 H_1^\top H_1 G = 0. \tag{11}$$

The solution leads to the update rule of G as follows

$$G = (H_1^\top H_1)^{-1} H_1^\top S. \tag{12}$$

Fixing H_2, G, C, Update H_1. The rule of updating H_1 is a little different from the above rules, since H_1 is not a continues variable. Here we use an exhaustive search for the optimal assignment to find the solution of H_1

$$k = \arg\min_j ||X_{1,i} - C_j||_2^2 + \lambda ||z_j - H_{1,i} G||_2^2, \tag{13}$$

where $X_{1,i}$ and $H_{1,i}$ denote the i-th row in X_1 and H_1, C_j is the j-th centroid and z_j is a $1 \times K$ vector with j-th position 1 and others 0.

Fixing H_1, G, C, Update H_2. Similar to the update rule of H_1, we use the same way to update H_2 as follows.

$$k = \arg\min_j ||X_{2,i} - C_j||_2^2. \tag{14}$$

By the above four steps, we alternatively update C, G, H_1 and H_2 and repeat the process until the objective function converges. Here we decompose the problem into 4 subproblems and each of them is a convex problem with one variable. Therefore, by solving the subproblems alternatively, our method will find a solution with the guarantee of convergence.

4.2 K-means-like optimization

Although the above solution is suitable for the clustering with partition level side information, it is not efficient due to some matrix multiplication and inverse. Besides if we have multiple side information, the data is separated to too many fractured pieces, which is hard to operate in real-world applications. This inspires us whether we can solve the above problem in a neat mathematical way with high efficiency. In the following, we equivalently transform 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 concatenated matrix D as follows,

$$D = \begin{bmatrix} X_1 & S \\ X_2 & 0 \end{bmatrix}.$$

Further we decomposed D into two parts $D = [D_1\ D_2]$, where $D_1 = X$ and $D_2 = [S\ 0]^\top$. Here we can see that D is exactly a concatenated matrix with the original data X and partition level side information S, 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 $d_i^{(2)} = (d_{i,m+1}, \cdots, d_{k,m+K})$ denote the side information; 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 filled up.

If we directly apply K-means on the matrix D, it might cause some problems. 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 a certain cluster. Let $m_k = (m_k^{(1)}, m_k^{(2)})$ be the k-th centroid of K-means, which $m_k^{(1)} = (m_{k,1}, \cdots, m_{k,m})$ and $m_k^{(2)} = (m_{k,m+1}, \cdots, m_{k,m+K})$. We modify the computation of the centroids as follows,

$$m_k^{(1)} = \frac{\sum_{x_i \in \mathbf{C}_k} x_i}{|\mathbf{C}_k|},\ m_k^{(2)} = \frac{\sum_{x_i \in \mathbf{C}_k \bigcap S} x_i}{|\mathbf{C}_k \bigcap S|}. \tag{15}$$

Recall that within the standard K-means, the centroids are computed by arithmetic means, whose denominator represents

Algorithm 1 The algorithm of PLCC with K-means

Input: X: data matrix, $n \times m$;
K: number of clusters;
S: $p-$partition level side information, $pn \times K$;
λ: trade-off parameter.
Output: optimal H^*;
1: Build the concatincating matrix D, $n \times (m+K)$;
2: Randomly select K instances as centroids;
3: **repeat**
4: Assign each instance to its closest centroid by the distance function in Eq. 17;
5: Update centroids by Eq. 15;
6: **until** the objective value in Eq. 2 remains unchanged.

the number of instances in its corresponding cluster. Here in Eq. 15, 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 Theorem 1.

Theorem 1. *Given the data matrix X, side information S and augmented matrix $D = \{d_i\}_{1 \le i \le n}$, we have*

$$\begin{aligned} &\min_{H,C,G} ||X - HC||_{\mathrm{F}}^2 + \lambda ||S - (H \otimes S)G||_{\mathrm{F}}^2 \\ &\Leftrightarrow \min \sum_{k=1}^{K} \sum_{d_i \in \mathbf{C}_k} f(d_i, m_k), \end{aligned} \tag{16}$$

where m_k is the k-th centroid calculated by Eq. 15 and the distance function f can be computed by

$$f(d_i, m_k) = ||d_i^{(1)} - m_k^{(1)}||_2^2 + \lambda \boldsymbol{1}(d_i \in S)||d_i^{(2)} - m_k^{(2)}||_2^2. \tag{17}$$

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

Proof. We start from the objective function of K-means.

$$\begin{aligned} &\sum_{k=1}^{K} \sum_{d_i \in \mathbf{C}_k} f(d_i, m_k) \\ &= \sum_{k=1}^{K} \sum_{d_i \in \mathbf{C}_k \cap S} (||d_i^{(1)} - m_k^{(1)}||_2^2 + \lambda ||d_i^{(2)} - m_k^{(2)}||_2^2) \\ &\quad + \sum_{k=1}^{K} \sum_{d_i \in \mathbf{C}_k \cap \overline{S}} ||d_i^{(1)} - m_k^{(1)}||_2^2 \\ &= ||X_1 - H_1 C||_{\mathrm{F}}^2 + \lambda ||S - H_1 G||_{\mathrm{F}}^2 + ||X_2 - H_2 C||_{\mathrm{F}}^2. \end{aligned} \tag{18}$$

According to the definition of the augmented matrix D and Eq. 2, we finish the proof. □

Remark 1. *Theorem 1 exactly maps the problem in Eq. 1 into a K-means clustering problem with modified distance function and centroid updating rules, which has a neat mathematical way and can be solved with high efficiency. Taking a close look at the concatenated 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 respectively and add them together.*

By Theorem 1, we transfer the problem into a K-means-like clustering problem. Since the updating 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 clustering.*

The proof of Theorem 2 is to show that centroid updating rules in Eq. 15 are optimal, which is similar to the proof of Theorem 6 in Ref [44]. Due to the limited space, we omit the proof here. We summarize the proposed algorithm in Algorithm 1. We can see that the proposed algorithm has the similar structure with the standard K-means, and it also enjoys the almost same time complexity with K-means, $O(tKn(m+K))$, where t is the iteration number, K is the cluster number, n and m are the numbers of instance and feature, respectively. Usually $K \ll n$ and $m \ll n$, so the algorithm is roughly linear to the instance number. This indicates that K-means-based PLCC is suitable for large-scale datasets.

5 Discussion

In this part, we discuss the extensions of our model. One is to handle multiple partition level side information, the other is to apply spectral clustering with partition level side information.

5.1 Handling Multiple Side Information

In real-world application, like crowd sourcing, the side information comes from multi-sources. Thus, how to conduct clustering with multiple side information is common in most scenarios. Next, we modify the objective function to extend our method to handle multiple side information.

$$\begin{aligned} &\min_{H,C,G_j} ||X - HC||_{\mathrm{F}}^2 + \sum_{j=1}^{r} \lambda_j ||S_j - (H \otimes S_j)G_j||_{\mathrm{F}}^2 \\ &s.t.\ H_{ik} \in \{0,1\}, \sum_{k=1}^{K} H_{ik} = 1, 1 \le i \le n. \end{aligned} \tag{19}$$

where $S = \{S_1, S_2, \cdots, S_r\}$ is the set of side information and λ_i is the weight of each side information. If we still apply the first solution, the data is separated into so many pieces that it is difficult to handle in practice. Thanks to the K-means-like solution, we concatenate all the side information after the original features and then employ K-means to find the final solution. The centroids consist of r parts, with $m_k = (m_k^{(1)}, m_k^{(2)}, \cdots, m_k^{(r+1)})$, which $m_k^{(j)}, 2 \le j \le r+1$ represents the part of centroids for r side information, and the update rule of centroids and the distance function can be computed as

$$m_k^{(j+1)} = \frac{\sum_{x_i \in \mathbf{C}_k \bigcap S_j} x_i}{|\mathbf{C}_k \bigcap S_j|}, \tag{20}$$

$$\begin{aligned} f(d_i, m_k) &= ||d_i^{(1)} - m_k^{(1)}||_2^2, \\ &\quad + \sum_{j=1}^{r} \lambda_j \mathbf{1}(d_i \in S_j)||d_i^{(j+1)} - m_k^{(j+1)}||_2^2. \end{aligned} \tag{21}$$

Algorithm 2 The algorithm of PLCC with spectral clustering

Input: X: data matrix, $n \times m$;
K: number of clusters;
S: $p-$partition level side information, $pn \times K$;
λ: trade-off parameter.
Output: optimal H^*;
1: Build the similarity matrix W;
2: Calculate the largest K engienvectors of $(D^{-1/2}WD^{-1/2} + \lambda[S\ 0]^\top[S\ 0])$;
3: Run K-means to obtain the final clustering.

5.2 PLCC with Spectral Clustering

K-means and spectral clustering are two widely used clustering methods, which handle the record data and graph data, respectively. Here we also want to incorporate the partition level side information into spectral clustering for broad use. Here we first give a brief introduction to spectral clustering and extend it to handle partition level side information. Let W be a symmetric matrix of given data, where w_{ij} represents a measure of the similarity between x_i and x_j. The objective function of normalized cuts spectral clustering is the following trace maximization problem [46]:

$$\begin{aligned} &\max_U \text{tr}(U^\top D^{-1/2}WD^{-1/2}U) \\ &\text{s.t. } U^\top U = I, \end{aligned} \tag{22}$$

where D is the diagonal matrix whose diagonal entry is the sum of rows of W and U is the scaled cluster membership matrix such that

$$U_{ij} = \begin{cases} 1/\sqrt{n_j}, & \text{if } x_i \in \mathbf{C}_j \\ 0, & \text{otherwise} \end{cases}.$$

We can easily get $U = H(H^\top H)^{-1/2}$ and $U^\top U = I$. The solution is to calculate the largest k eigenvalues of $D^{-1/2}WD^{-1/2}$, and run K-means to get the final partition [46].

Similar to the trick we use for K-means, we also separate U into two parts U_1 and U_2 according to side information. Let U_1 denote the scaled cluster membership matrix for the instances with side information, and U_2 represent the the scaled cluster membership matrix for the instances without side information. Then we can add the side information part and rewrite Eq. 22 as follow.

$$\max_{U_1,U_2} \text{tr}(\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}^\top D^{-1/2}WD^{-1/2} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}) - \lambda||S - H_1G||_{\text{F}}^2. \tag{23}$$

For the second term, through some derivations we can obtain the following equation [47],

$$||S - H_1G||_{\text{F}}^2 = ||S||_{\text{F}}^2 - \text{tr}(U_1^\top SS^\top U_1). \tag{24}$$

Since $||S||_{\text{F}}^2$ is a constant, finally we derive the objective function for spectral clustering with partition level side information.

$$\begin{aligned} &\max_{U_1,U_2} \text{tr}(\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}^\top (D^{-1/2}WD^{-1/2} + \lambda \begin{bmatrix} S \\ 0 \end{bmatrix} \begin{bmatrix} S \\ 0 \end{bmatrix}^\top) \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}) \\ &\Leftrightarrow \max_U \text{tr}(U^\top (D^{-1/2}WD^{-1/2} + \lambda \begin{bmatrix} S \\ 0 \end{bmatrix} \begin{bmatrix} S \\ 0 \end{bmatrix}^\top)U). \end{aligned} \tag{25}$$

TABLE 3
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
$Dogs$	20580	2048	120	0.1354
AWA	30475	4096	50	1.3499
$Pascal$	12695	4096	20	4.6192
$MNIST$	70000	160	10	0.0570

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

To solve the above optimization problem, we have the following theorem.

Theorem 3. *The optimal solution U^* is composed by the largest K eigenvectors of $(D^{-1/2}WD^{-1/2} + \lambda \begin{bmatrix} S \\ 0 \end{bmatrix} \begin{bmatrix} S \\ 0 \end{bmatrix}^\top)$.*

The proof is similar to the one of spectral clustering, we omit it here due to the limited page. And the algorithm is summarized in Alg. 2.

Remark 2. *Similar to Theorem 1, Theorem 3 transforms the spectral clustering with partition level side information into a new spectral clustering problem. So a modified similarity matrix is calculated and followed by the standard spectral clustering. We can see that partition level side information enhances coherence within clusters.*

6 Experimental Results

In this section, we present the experimental results of PLCC nested K-means and spectral clustering compared to pairwise constrained clustering and ensemble clustering methods. Generally speaking, we first demonstrate the advantages of our method in terms of effectiveness and efficiency. Next, we add noises with different ratios to analyse the robustness and finally the experiments with multiple side information and inconsistent cluster number illustrate the validation of our method in real-world application.

6.1 Experimental Setup

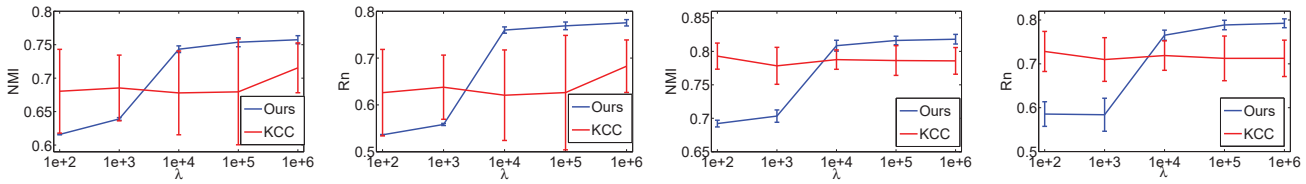
Experimental data. We use a testbed consisting of seven data sets obtained from UCI repositories[1] and four image data sets with deep features[2 3 4 5]. Table 3 shows some important characteristics of these datasets, where CV is the Coefficient of Variation statistic that characterizes the degree of class imbalance. A higher CV value indicates a more severe class imbalance.

Tools. We choose four methods as competitive methods. L-CVEQ [15] is a K-means-based pairwise constraint clustering method; KCC is an ensemble clustering methoud [40], which first generates one basic partition alone from the data and then fuse this partition with incomplete partition level side information; FSC [30] is a spectral-based clustering method with pairwise

1. https://archive.ics.uci.edu/ml/datasets.html
2. http://vision.stanford.edu/aditya86/ImageNetDogs/
3. http://attributes.kyb.tuebingen.mpg.de/
4. https://www.ecse.rpi.edu/homepages/cvrl/database/AttributeDataset.htm
5. http://yann.lecun.com/exdb/mnist/

TABLE 4
Clustering performance on seven real datasets by *NMI*

Data Sets	percent	Ours(K-means)	CNMF	LCVQE	KCC	K-means	Ours(SC)	FSC	SC
	10%	**0.7591±0.0137**	0.7242±0.0262	0.7588±0.0138	0.7574±0.0122		**0.7884±0.0188**	0.1618±0.1368	
	20%	**0.7820±0.0185**	0.7430±0.0204	0.7815±0.0186	0.7759±0.0148		**0.8116±0.0213**	0.1645±0.0537	
breast	30%	**0.8071±0.0214**	0.7691±0.0248	0.8059±0.0212	0.8001±0.0198	0.7361±0.0000	**0.8446±0.0229**	0.2109±0.0778	0.7563±0.0000
	40%	**0.8320±0.0196**	0.7973±0.0278	0.8156±0.1129	0.8246±0.0186		**0.8712±0.0219**	0.2899±0.0602	
	50%	**0.8538±0.0186**	0.8375±0.0217	0.8196±0.1656	0.8458±0.0182		**0.8892±0.0251**	0.3298±0.0922	
	10%	**0.6416±0.0231**	0.6184±0.0508	0.6087±0.0332	0.5957±0.0522		0.4184±0.1391	**0.4902±0.0490**	
	20%	**0.6820±0.0298**	0.6537±0.0430	0.6324±0.0471	0.6056±0.0511		0.4388±0.0954	**0.4677±0.0606**	
ecoli	30%	**0.7321±0.0274**	0.6772±0.0363	0.6782±0.0456	0.6289±0.0621	0.6053±0.0253	0.4487±0.0863	**0.4834±0.0728**	0.5575±0.0086
	40%	**0.7692±0.0284**	0.7119±0.0390	0.7046±0.0454	0.6504±0.0484		0.4634±0.0696	**0.4993±0.0466**	
	50%	**0.8084±0.0272**	0.7410±0.0392	0.7283±0.0533	0.6957±0.0611		0.4990±0.0177	**0.5336±0.0332**	
	10%	0.3749±0.0292	0.1908±0.0887	0.3744±0.0347	**0.3872±0.0333**		**0.3570±0.0724**	0.2466±0.0706	
	20%	**0.3973±0.0270**	0.1908±0.0993	0.3595±0.0373	0.3842±0.0314		**0.4096±0.0636**	0.2950±0.0562	
glass	30%	**0.4251±0.0296**	0.2182±0.1006	0.3466±0.0457	0.3905±0.0306	0.3846±0.0361	**0.4591±0.0383**	0.3208±0.0439	0.4070±0.0042
	40%	**0.4716±0.0337**	0.2534±0.0994	0.3405±0.0345	0.3861±0.0324		**0.5064±0.0275**	0.3833±0.0416	
	50%	**0.5201±0.0282**	0.2770±0.1036	0.3208±0.0527	0.3816±0.0415		**0.5550±0.0234**	0.4258±0.0504	
	10%	**0.7653±0.0177**	0.7135±0.1016	0.7597±0.0341	0.7258±0.0929		**0.7339±0.0678**	0.2662±0.2339	
	20%	**0.7846±0.0241**	0.7298±0.0974	0.7829±0.0271	0.7217±0.1165		**0.7036±0.0540**	0.2915±0.1911	
iris	30%	**0.8105±0.0279**	0.7846±0.1037	0.8096±0.0347	0.7637±0.0961	0.7244±0.0682	**0.7077±0.0489**	0.3562±0.1846	0.7313±0.0290
	40%	**0.8366±0.0283**	0.7855±0.0984	0.8303±0.0608	0.7993±0.0727		**0.6949±0.1139**	0.4571±0.1840	
	50%	**0.8541±0.0303**	0.8067±0.1058	0.8502±0.0388	0.8178±0.0670		**0.7128±0.1104**	0.5943±0.1677	
	10%	**0.6920±0.0149**	0.6801±0.0128	0.6672±0.0120	0.6531±0.0261		**0.5242±0.0441**	0.4183±0.0978	
	20%	**0.7101±0.0188**	0.6961±0.0082	0.6313±0.0231	0.6673±0.0392		**0.4611±0.0454**	0.3916±0.0617	
pendigits	30%	**0.7289±0.0327**	0.7031±0.0304	0.5984±0.0251	0.6858±0.0164	0.6822±0.0148	**0.4631±0.0542**	0.4239±0.0561	0.6522±0.0191
	40%	**0.7645±0.0186**	0.7469±0.0151	0.5786±0.0216	0.7535±0.0306		**0.4690±0.0542**	0.4595±0.0392	
	50%	**0.8054±0.0129**	0.7601±0.0132	0.5406±0.0242	0.7882±0.0306		**0.4986±0.0470**	0.5249±0.0372	
	10%	**0.6140±0.0005**	0.2318±0.0318	0.5456±0.0515	0.5484±0.0724		**0.4456±0.0304**	0.3310±0.0754	
	20%	**0.6143±0.0006**	0.2541±0.0264	0.5263±0.0886	0.6028±0.0498		**0.4466±0.0367**	0.3261±0.0470	
satimage	30%	**0.6149±0.0005**	0.3000±0.0223	0.5133±0.1065	0.5807±0.0679	0.5752±0.0588	**0.4801±0.0280**	0.3364±0.0297	0.5198±0.0306
	40%	0.6153±0.0004	0.3413±0.0184	0.4446±0.1025	**0.6430±0.0447**		**0.4921±0.0316**	0.4056±0.0210	
	50%	0.6161±0.0008	0.4231±0.0346	0.4505±0.1193	**0.6896±0.0521**		**0.5155±0.0665**	0.4570±0.0287	
	10%	**0.2944±0.0532**	0.2426±0.1050	0.2697±0.0592	0.2727±0.0552		**0.4325±0.0771**	0.1865±0.1262	
	20%	**0.3463±0.0505**	0.2321±0.1105	0.2554±0.0771	0.2993±0.0565		**0.4749±0.0574**	0.2470±0.0962	
wine	30%	**0.3774±0.0482**	0.2711±0.0980	0.2339±0.0828	0.3362±0.0527	0.1307±0.0087	**0.5069±0.0751**	0.3137±0.0910	0.4007±0.0271
	40%	**0.4310±0.0345**	0.2887±0.1331	0.1981±0.1076	0.3715±0.0532		**0.5305±0.0762**	0.4019±0.0677	
	50%	**0.4636±0.0355**	0.3267±0.1215	0.1960±0.1334	0.4360±0.0531		**0.5760±0.0690**	0.4904±0.0447	



(a) *satimage* by *NMI* (b) *satimage* by R_n (c) *pendigits* by *NMI* (d) *pendigits* by R_n

Fig. 2. Impact of λ on *satimage* and *pendigits*.

constraint; CNMF [22] is an NMF-based constrained clustering method, which also employs the partition level side information as input. In our method, there is only one parameter λ, here we empirically set it to 100, and we also set the weight of side information as 100 in KCC. 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 and FSC. Although there exist many K-means-based constrained clustering methods, Ref [21] thoroughly studied the K-Means-based algorithms for constrained clustering and recommended LCVQE [15], which presents better performance and violates less constraint than CVQE [14] and MPCK-Means [16]. Therefore, we choose LCVQE as the pairwise constraint comparative algorithm. Note that the number of clusters for three algorithms is set to the number of true clusters.

Validation measure. Since class labels are provided for each data set, Normalized Mutual Information (*NMI*) and Normalized Rand Index (R_n) are used to measure the clustering performance.

Normalized Mutual Information (NMI), measures the mutual information between resulted cluster labels and ground truth labels, followed by a normalization operation to assure NMI ranges from 0 to 1. Mathematically, it is defined as:

$$NMI = \frac{\sum_{i,j} n_{ij} \log \frac{n \cdot n_{ij}}{n_{i+} \cdot n_{+j}}}{\sqrt{(\sum_i n_{i+} \log \frac{n_{i+}}{n})(\sum_j n_{j+} \log \frac{n_{+j}}{n})}}. \quad (26)$$

Normalized Rand Index, denoted as R_n measures the similarity between two partitions in a statistical way, which is defined as:

$$R_n = \frac{\sum_{i,j} \binom{n_{ij}}{2} - \sum_i \binom{n_{i+}}{2} \cdot \sum_j \binom{n_{+j}}{2} / \binom{n}{2}}{\sum_i \binom{n_{i+}}{2}/2 + \sum_j \binom{n_{+j}}{2}/2 - \sum_i \binom{n_{i+}}{2} \cdot \sum_j \binom{n_{+j}}{2} / \binom{n}{2}}. \quad (27)$$

All the variables in Eq. 26 and 27 can be found in Table 1. Note that both NMI and R_n are positive measurements, i.e, a better partition has a larger NMI or R_n value.

Environment. All the experiments were run on a Ubuntu 14.04 platform with Intel Core i7-6900K @ 3.2GHz and 64 GB RAM.

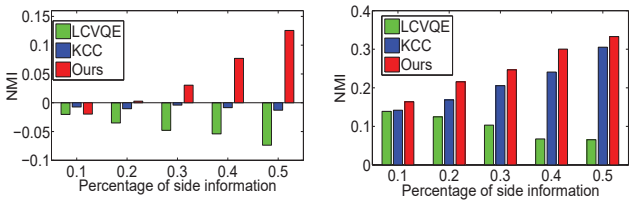
6.2 Effectiveness and Efficiency

Table 4 and 5 show the clustering performance of different algorithms on all the seven data sets with side information of different ratios measured by *NMI* and R_n, respectively. In each scenario,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2763945, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 5
Clustering performance on seven real datasets by R_n

Data Sets	percent	Ours(K-means)	CNMF	LCVQE	KCC	K-means	Ours(SC)	FSC	SC
	10%	**0.8564±0.0103**	0.8271±0.0222	0.8562±0.0104	0.8551±0.0090		**0.8778±0.0125**	0.1112±0.2094	
	20%	**0.8735±0.0136**	0.8420±0.0176	0.8732±0.0137	0.8690±0.0109		**0.8941±0.0139**	0.0687±0.1096	
breast	30%	**0.8912±0.0150**	0.8622±0.0204	0.8904±0.0150	0.8862±0.0139	0.8391±0.0000	**0.9155±0.0140**	0.1137±0.1337	0.8552±0.0000
	40%	**0.9081±0.0131**	0.8827±0.0205	0.8906±0.1212	0.9031±0.0122		**0.9318±0.0129**	0.1555±0.1502	
	50%	**0.9228±0.0118**	0.9113±0.0145	0.8870±0.1745	0.9174±0.0117		**0.9424±0.0149**	0.2474±0.1589	
	10%	**0.5377±0.0587**	0.5783±0.1127	0.5093±0.0849	0.4639±0.0880		0.3570±0.1570	**0.4198±0.0770**	
	20%	**0.6460±0.0831**	0.6200±0.1080	0.5780±0.0884	0.5056±0.1126		0.2996±0.1250	**0.3651±0.0808**	
ecoli	30%	**0.7351±0.0793**	0.6486±0.0894	0.6488±0.0910	0.5336±0.1248	0.4732±0.0772	0.3259±0.1043	**0.3882±0.1102**	0.4434±0.0489
	40%	**0.7957±0.0581**	0.7153±0.0785	0.6901±0.0883	0.5630±0.0992		0.2261±0.1080	**0.3194±0.1068**	
	50%	**0.8458±0.0258**	0.7479±0.0739	0.7304±0.0877	0.6412±0.1042		0.2326±0.0288	**0.3386±0.0902**	
	10%	0.2397±0.0338	0.0969±0.0597	0.2360±0.0284	**0.2442±0.0307**		**0.1879±0.0650**	0.1036 ±0.0847	
	20%	**0.2619±0.0368**	0.1072±0.0651	0.2218±0.0287	0.2426±0.0312		**0.1912±0.0691**	0.1184±0.0737	
glass	30%	**0.2795±0.0393**	0.1345±0.0728	0.2084±0.0355	0.2510±0.0313	0.2552±0.0289	**0.2133±0.0465**	0.0975±0.0652	0.2463±0.0059
	40%	**0.3310±0.0375**	0.1696±0.0817	0.1990±0.0230	0.2436±0.0326		**0.2586±0.0344**	0.1683±0.0595	
	50%	**0.4019±0.0332**	0.1965±0.0804	0.1897±0.0434	0.2377±0.0335		**0.3214±0.0280**	0.2244±0.0705	
	10%	**0.7454±0.0229**	0.6627±0.1534	0.7387±0.0443	0.6801±0.1373		**0.6380±0.1300**	0.1437±0.2247	
	20%	**0.7755±0.0325**	0.6802±0.1491	0.7743±0.0349	0.6770±0.1666		**0.5814±0.0984**	0.1371±0.2025	
iris	30%	**0.8131±0.0371**	0.7664±0.1462	0.8128±0.0442	0.7358±0.1388	0.6690±0.1237	**0.5918±0.0817**	0.1847±0.1976	0.6835±0.0898
	40%	**0.8423±0.0347**	0.7578±0.1517	0.8357±0.0726	0.7813±0.1057		**0.5875±0.1462**	0.2888±0.2154	
	50%	**0.8673±0.0358**	0.7680±0.1776	0.8642±0.0434	0.8079±0.0994		**0.6096±0.1582**	0.4552±0.2038	
	10%	**0.5874±0.0387**	0.5288±0.0317	0.5749±0.0239	0.5204±0.0448		**0.3136±0.0627**	0.1964±0.1036	
	20%	**0.6186±0.0405**	0.5708±0.0110	0.5305±0.0493	0.5375±0.0655		**0.1902±0.0650**	0.1197±0.0661	
pendigits	30%	**0.6475±0.0684**	0.5650±0.0620	0.4884±0.0506	0.5586±0.0237	0.5611±0.0385	**0.1659±0.0808**	0.1216±0.0826	0.5431±0.0272
	40%	**0.6978±0.0419**	0.6406±0.0391	0.4621±0.0355	0.6702±0.0516		**0.1353±0.0807**	0.1161±0.0729	
	50%	**0.7674±0.0237**	0.6411±0.0273	0.3998±0.0328	0.7207±0.0642		**0.1558±0.0782**	0.1992±0.0772	
	10%	**0.5347±0.0006**	0.1458±0.0327	0.4603±0.0754	0.4600±0.0894		**0.2994±0.0407**	0.1807±0.0963	
	20%	**0.5348±0.0007**	0.1553±0.0326	0.4573±0.1214	0.5315±0.0798		**0.2664±0.0258**	0.1021±0.0809	
satimage	30%	**0.5355±0.0003**	0.2159±0.0196	0.4498±0.1369	0.4931±0.0941	0.4804±0.0826	**0.2599±0.0535**	0.0601±0.0280	0.5198±0.0306
	40%	0.5356±0.0005	0.2583±0.0371	0.3603±0.1398	**0.5777±0.0694**		**0.2223±0.0910**	0.1034±0.0331	
	50%	0.5364±0.0007	0.3515±0.0446	0.3431±0.1586	**0.6419±0.0768**		**0.2542±0.1075**	0.1538±0.0353	
	10%	**0.2273±0.0434**	0.2117±0.0930	0.2029±0.0603	0.1947±0.0463		**0.3649±0.1044**	0.0717±0.1385	
	20%	**0.2749±0.0438**	0.1926±0.1086	0.1897±0.0697	0.2161±0.0510		**0.3722±0.0669**	0.0880±0.1370	
wine	30%	**0.3068±0.0406**	0.2203±0.1055	0.1793±0.0786	0.2465±0.0561	0.1275±0.0042	**0.4016±0.1004**	0.1269±0.1268	0.3064±0.0329
	40%	**0.3559±0.0308**	0.2551±0.1275	0.1524±0.1027	0.2844±0.0458		**0.4223±0.1090**	0.2089±0.0900	
	50%	**0.3847±0.0266**	0.2946±0.1167	0.1534±0.1240	0.3332±0.0528		**0.4637±0.0949**	0.3210±0.0620	



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

Fig. 3. Improvement of constrained clustering on *glass* and *wine* compared with K-means.

TABLE 6
Comparison of Execution Time (in seconds)

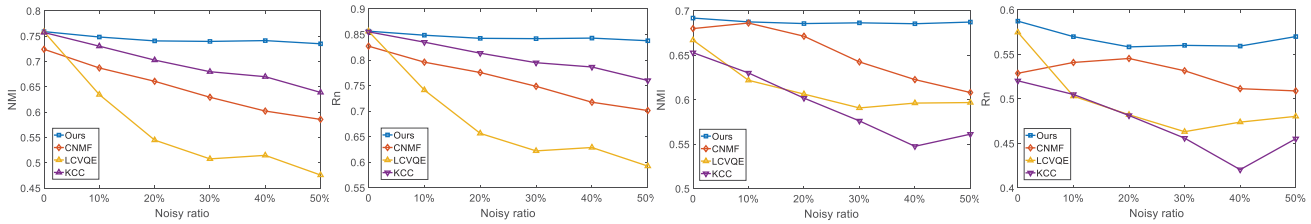
Data Sets	Ours(K-means)	CNMF	LCVQE	KCC	Ours(SC)	FSC
breast	**0.0014**	0.4235	0.0461	0.2638	**0.5429**	4.4632
ecoli	**0.0117**	0.1939	0.0318	0.2175	**0.1591**	1.0187
glass	**0.0052**	0.1936	0.0256	0.1263	**0.1067**	0.3323
iris	**0.0019**	0.1259	0.0097	0.0673	**0.0874**	0.1373
pendigits	**0.4538**	195.3840	76.7346	4.9807	**651.7113**	>4.5hr
satimage	**0.1887**	13.8217	11.5499	1.7020	**56.7173**	1304.2479
wine	**0.0094**	0.0535	0.0126	0.1030	**0.0718**	0.1934

50 runs with different random initializations are conducted and the average performance as well as the standard deviation are reported.

In the K-means-based scenario, our method achieves the best performance in most cases except on *glass*, *pendigits* and *satimage* with 10%, 40% and 50% percent side information (We will tune λ to get better performance on *pendigits* and *satimage* later). If we take a close look at Table 4 and 5, 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* than the basic K-means without any guidance. This might result from the great impact of the order of pairwise constraints, which leads to the deformity of clustering structure. 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. 3 shows the improvement of constrained clustering algorithms over the baseline methods on *glass* and *wine*. It can be seen that for 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. CNMF and our method both take the partition level side information as input. Our method consistently outperforms CNMF, especially on *glass* and *satimages*, which demonstrates the utility function helps to preserve the structure from side information. Although we equivalently transfer the partition level side information into pairwise constraints, our clustering method utilizes the consistency within the side information and achieves better results. In the spectral clustering scenario, our method has also consistent better performance than FSC on all datasets but *ecoli*. Generally speaking, our K-means-based method achieves better performance than the basic K-means, while sometimes our spectral-based method and FSC cannot beat the single spectral clustering.

Next, we evaluate six algorithms in terms of efficiency. Table 6 shows the average of execution time of different algorithms with 10% side information. From the table, we can see that our method shows obvious advantages than other three algorithms. On *pendigits*, our K-means-based method is 10 times faster than KCC, nearly 170 times than LCVQE, 430 times faster than CNMF and our spectral clustering based method run 20 times faster than

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2763945, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) *breast* by *NMI* (b) *breast* by R_n (c) *pendigits* by *NMI* (d) *pendigits* by R_n

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

FSC on large datesets. Taking the effectiveness and efficiency into account, our K-means-based 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 for fair comparisons due to the unsupervised fashion, 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. 2 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 consistently goes up with the increasing of λ with high robustness; besides, our method achieves stability when λ is larger than a threshold, like $1e+4$. Recall that λ plays a key role in controlling the degree that how the learnt partition achieves close to the side information. From this view, λ should be set as large as possible when the given side information is confidence. However, when it comes to noisy side information, we should set λ in an appropriate range (See the application in Section 7).

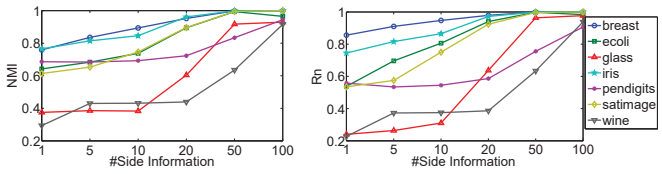
6.3 Handling Side Information with Noises

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 CNMF, 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 weight 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 a rough good partition level side information is good enough (This point can also be verified in Section 7), which validates the effectiveness of our method in practice with noisy side information.

6.4 Handling Multiple Side Information

In crowd sourcing, the side information comes from multi-sources and multi-agents. In the following we show our method handles multiple side information. Here each agent randomly selects 10% instances and provides its corresponding partition level side information. Fig. 5 shows the performance of our method with different numbers of side information. With the increasing of the number of side information, the performance on all data sets goes up with a great improvement, even for the not well-separated data sets, such as *glass* and *wine*. This reveals that our method can easily be applied to crowd sourcing and significantly improve the clustering result with multiple side information.



(a) Measured by *NMI* (b) Measured by R_n

Fig. 5. Impact of the number of side information.

6.5 Inconsistent Cluster Number

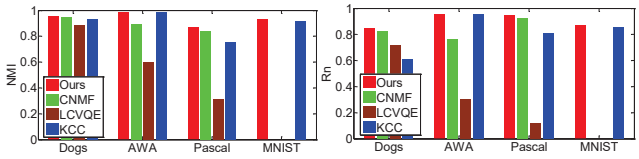
Here we continue to evaluate our proposed method in the scenario that the side information contains inconsistent cluster number with the final cluster number. This obeys the nature of cluster analysis, which aims to uncover the new clusters and cannot be solved by the traditional classification task. Moreover, it is quite suitable for labeling task with only partial data labeled. To simulate such scenario, we label 50% data instances from the first 50% classes on *Dogs*, *AWA*, *Pascal* and *MNIST* as the side information, and then conduct the clustering methods with the true cluster number.

Figure 6 shows the performance of different clustering methods in the setting of inconsistent cluster number. Note that CNMF and LCVQE fail to deliver the partitions on *MNIST* due to the negative input and out-of-memory, respectively. On these four datasets, our method achieves the best performance over other rivals, which demonstrates the effectiveness of our method in real-world applications. Moreover, our method does not need to store cannot-link or must-link constraints, instead employs the partition-level side information. Taking the efficiency and memory into consideration, our method is suitable for large-scale data clustering.

So far the ground truth is employed as the partition level side information for clustering; however, we hardly obtain precious pre-knowledge in practice. In the next section, we illustrate the effectiveness of PLCC in a real-world application. A totally unsupervised saliency-guided side information, which contains noisy and missing labels is incorporated as the side information for the cosegmentation task.

7 Application to Image Cosegmentation

Image clustering, which provides a disjoint image-region partition, has been widely used for the computer vision community,



(a) Measured by *NMI* (b) Measured by R_n

Fig. 6. Performance with inconsistent cluster number on four large scale data sets.

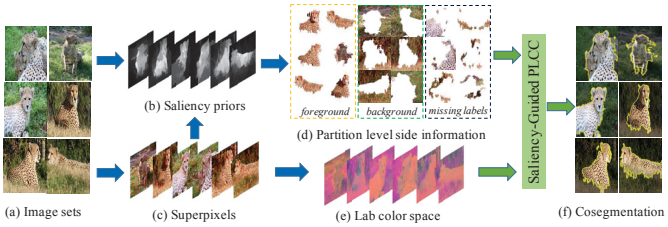


Fig. 7. Illustration of the proposed SG-PLCC model.

especially the multi-image scenario, such as co-saliency detection [48] and cosegmentation [49], [50], [51]. Here, based on our PLCC method, we propose a Saliency-Guided Constraint Clustering (SG-PLCC) model for the task of image cosegmentation, to show PLCC as an efficient and flexible image clustering tool. In details, we employ saliency prior to obtain the partition level side information, and directly use PLCC to cluster image elements (*i.e.*, superpixels) into two classes. In the rest of this section, a brief introduction to the related work comes first, followed by our saliency-guided model, and finally the experimental result is given.

7.1 Cosegmentation

Rother et al. [52] first introduced cosegmentation as to extract the similar objects from an image pair with different background, by minimizing the histogram matching in a Markov Random Filed (MRF). The other two early works could be found in [53] and [54], which also focused on the situation of an image pair sharing with same object. After that, cosegmentation is extended for the multi-image scenario. For example, Joulin et al. [49] employed discriminative clustering to simultaneously segment the foreground from a set of images. For another example, Batra et al. [55] developed an interactive algorithm, intelligently guided by the user scribble information, to achieve cosegmentation for multi-images. Multiple foreground cosegmentation was first proposed by Kim et al. [56] as to jointly segment K different foregrounds from a set of input images. In their work, an iterative optimization process was performed for foreground modeling and region assignment under a greedy manner. Jolin et al. [50] also provided an energy-based model that combines spectral and discriminate clustering to handle multiple foreground and images, and optimized it with Expectation-Minimization (EM) method. Although all theses methods above have achieved significant performance, they may suffer from the requirement of user interaction to guide the cosegmentation [55], or the high computing cost of solving an energy optimization [49], [50], [52], [53].

Compared with these works above, the contributions of using PLCC for cosegmentation are threefold: (1) We provide an alternative cosegmentation approach (SG-PLCC), which is simple yet efficient; (2) Our cosegmentation method could be regarded as a rapid preprocessing for other application, benefiting from the linear optimization in PLCC; (3) We provide a flexible framework to integrate various information, such as user scribble, face detection, and saliency prior, which all can be used as the multiple side information for PLCC.

7.2 Salincy-Guided Model

Existing saliency models mainly focus on detecting the most attractive object within an image [57], whose output is always a probability distribution map (*i.e.*, saliency map) to the foreground. Thus, it could be seen as a "soft" binary segmentation for an image. Moreover, co-saliency detection [48], [58] aims to extract the common salient objects from multiple images, making it as an appropriate prior for cosegmeantion. Generally speaking, there are two main advantages of using saliency prior: 1) most saliency/co-saliency methods are bottom-up and biology inspired, which means they may detect candidate foreground objects in an unsupervised and rapid way; 2) highlighting the salient objects suppresses the common background across images.

However, there still exists two main problems for directly employing saliency prior as the partition level information. First, saliency detection method only provides the probability of each pixel belonging to the foreground, thus we may need to compute the certain label information based on it. Second, one may note that, the "label" we get from saliency is actually a kind of pseudo label, leading to the fact that our method may suffer from the incorrect label information from the saliency prior.

To solve above challenges, we employ a *partial observation* strategy. Given N input images, each of which is represented as a set of superpixels $\mathcal{X}_i = \{x_j\}_{j=1}^n$ by using [59], $1 \leq i \leq N$, and assigned a saliency prior by performed any saliency detection algorithm on it. Without loss of generality, we denote n as the number of superpixels and M the saliency map for each image. For $\forall x \in \mathcal{X}_i$, let $M(x) \in [0, 1]$ be its saliency prior, which is computed as the average saliency value of all the pixels within x. Then, the side information S is defined as:

$$S(x) = \begin{cases} 2\text{: foreground}, & M(x) \geq T_f \\ 1\text{: background}, & M(x) \leq T_b \\ 0\text{: missing}, & \text{otherwise} \end{cases}, \tag{28}$$

where T_f is a threshold for foreground and T_b for background. As suggested by [60], $T_f = \mu + \delta$, where μ and δ are calculated as the mean and standard deviation of M, respectively. Instead of assigning *background* to the remainder directly, $T_b = \mu$ is introduced as a background threshold, that is, we assume the superpixels lower than the average saliency value should belong to the background. By using Eq. 28, we remain the uncertainty of saliency prior as *missing observation*, to avoid wrongly labeling the true foreground. On the other hand, some error detections may exist in the saliency prior. We explain these missing labels and possible errors as the noises in side information S. As we mentioned in Section 6.3 before, PLCC can handle the side information with noises, thus, it alleviates the deficiency of saliency detection.

More details of SG-PLCC are shown by Fig. 7. To exploit the corresponding information among input images (a), we perform the co-saliency model proposed by [58] to achieve the saliency prior. After obtaining co-saliency maps (b) and superpixels (c), the side information (d) is computed by Eq. 28. We then simply extract the mean Lab feature for each superpixel in (e). Finally, the cosegmentaion (f) is achieved by performing PLCC for each

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2763945, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 7
Clustering performance of our method and different priors on iCoseg dataset

Criteria	K-means	Saliency Prior				SG-PLCC
		[61]	[62]	[63]	[58]	
R_n	0.4311	0.5561	0.5378	0.5215	0.5803	**0.6199**
NMI	0.3916	0.4810	0.4762	0.4587	0.5187	**0.5534**

TABLE 8
Comparison of segmentation accuracy on iCoseg dataset

Object class	image subset	[49]	[64]	[65]	SG-PLCC
Alaskan Bear	9/19	74.8	**90.0**	86.4	87.2
Hot Balloon	8/24	85.2	90.1	89.0	**93.8**
Baseball	8/25	73.0	90.1	90.5	**92.7**
Bear	5/5	74.0	**95.3**	80.4	82.3
Elephant	7/15	70.1	43.1	75.0	**90.0**
Ferrari	11/11	85.0	89.9	84.3	**90.0**
Gymnastics	6/6	90.9	91.7	87.1	**96.9**
Kite	8/18	87.0	90.3	89.8	**97.8**
Kite panda	7/7	73.2	**90.2**	78.3	81.2
Liverpool	9/33	76.4	87.5	82.6	**91.1**
Panda	8/25	84.0	**92.7**	60.0	80.0
Skating	7/11	82.1	77.5	76.8	**82.2**
Statue	10/41	90.6	93.8	91.6	**95.7**
Stone	5/5	56.6	63.3	**87.3**	82.0
Stone 2	9/18	86.0	**88.8**	88.4	80.0
Taj Mahai	5/5	73.7	**91.1**	88.7	83.2
Average		78.9	85.4	83.5	**87.9**

image, which jointly combines the feature and label information. It worthy to note that, most missing observations in (d) are segmented as foreground successfully, showing the capability of PLCC to handle the noise in side information.

7.3 Experimental Result

Here, we test the effectiveness of the proposed clustering approach PLCC for a real application task (*i.e.*, image cosegmentation). We perform our cosegmentation model SG-PLCC on the widely used iCoseg dataset [55], which consists of 643 images with 38 object groups and focuses on the *foreground/background* segmentation.

Implementation Details. The saliency prior is obtained by conducting the co-saliency model in [58], which combines the results of three efficient saliency detection methods [61], [62], [63]. For simplicity, our SG-PLCC approach employs the LAB features on a superpixel level, *i.e.*, the mean LAB color values (three-dimensional vector) of a superpixel. Three baseline methods [49], [64], [65] are used to compare with our SG-PLCC, where we directly report the results provided in their papers.

Clustering Performance. As shown by Table 7, we fist validate our result as a $K = 2$ clustering task, under two criteria R_n and *NMI*, respectively. A classic *K-means* algorithm is directly employed with Lab color feature on image superpixels as a baseline. However, it cannot explore the clustering structure effectively. On the other side, we divide each saliency map [58] (including three elementary methods [61], [62], [63]) into 2 classes with T_f thresholding, to demonstrate the effectiveness of our saliency prior. Interestingly, though the discriminative of feature is limited, our SG-PLCC model still improves the performance of saliency prior S by around 4%, showing that the PLCC can combine the feature and side information effectively.

Cosegmentation Performance. Table 8 shows the quantitative comparison between SG-PLCC and other methods by segmentation accuracy (*i.e.*, the percentage of correctly classified pixels to the total). We follow the same experiment setting as [64], where

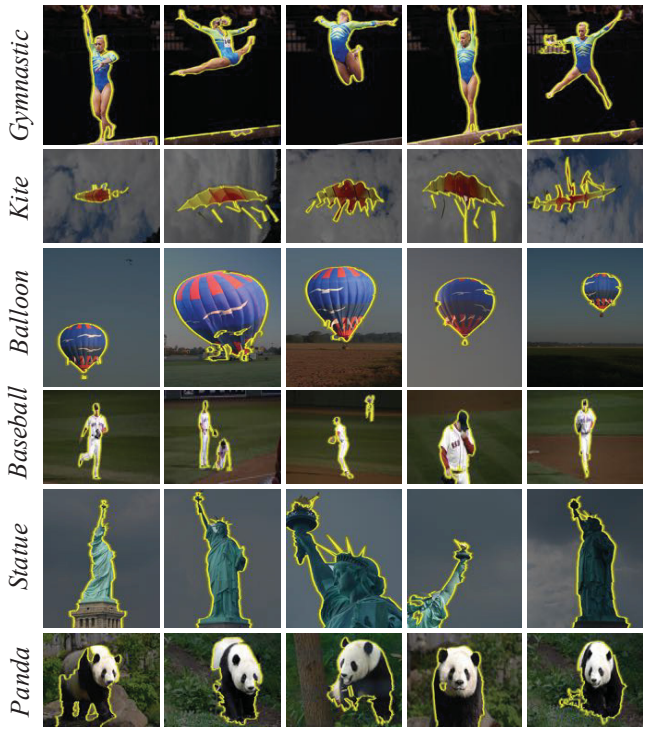


Fig. 8. Cosegmentation results of SG-PLCC on six image groups.

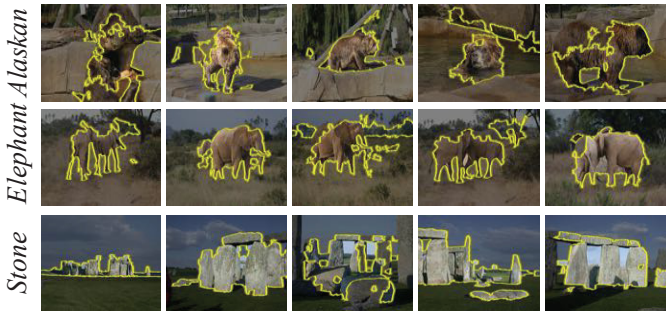


Fig. 9. Some challenging examples for our SG-PLCC model.

all the methods are tested on a subset of each image group from 16 selected object classes in the iCoseg dataset. For fairness, we average the performance of SG-PLCC over 20 random image subset for each object. It can be seen that, SG-PLCC outperforms others in general, and improves the average accuracy of 2.5% to the second best. Moreover, our method achieves 95.7%, 96.9% and 97.8%, nearly one hundred percentage, on the classes of *Statue*, *Gymnastics*, and *Kite*, respectively, without high computing optimization and label information, which significantly shows the success of using PLCC for real application.

Visually, some examples of our results are shown in Fig. 8, where the foreground is segmented with yellow line while the background darkened for a better view. For these cases, pretty fine segmentations are provided by SG-PLCC. However, our performance may degrade for some more challenging scenarios. As shown by Fig. 9, we fail to segment out the entire foreground, and suffer from the cluttered background. To solve these problems, we could feed the SG-PLCC results into some conventional segmentation frameworks to improve the performance of cosegmenation,

and employ more discriminative feature rather than the raw Lab. In addition, the SG-PLCC model can be easily extended for the multi-class cosegmentation with the increase of clustering number.

To sum up, the proposed SG-PLCC model provides an successful example of using PLCC in the real application task. Although SG-PLCC is directly performed with raw features, and only guided by unsupervised saliency prior, we still achieve a promising result for image cosegmeatation, which demonstrates the power of our PLCC method.

8 Conclusion

In this paper, we proposed a novel framework for clustering with partition level side information, called PLCC. Different from pairwise constraints, partition level side information accords with the labeling from human being with other instances as references. Within the PLCC framework, we formulated the problem via conducting clustering and making the structure agree as much as possible with side information. Then we gave its corresponding solution, equivalently transformed it into K-means clustering and extended it to handle multiple side information and spectral clustering. Extensive experiments demonstrated the effectiveness and efficiency of our method compared to three state-of-the-art algorithms. Besides, our method had high robustness when it comes to noisy side information and finally we validated the performance of our method with multiple side information and inconsistent cluster number setting. The cosegmentation application demonstrated the effectiveness of PLCC as a flexible framework in the image domain.

9 Acknowledgement

This research is supported in part by the NSF IIS award 1651902, ONR Young Investigator Award N00014-14-1-0484, and U.S. Army Research Office Award W911NF-17-1-0367.

References

[1] 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*, vol. 14, no. 1, pp. 1–37, 2008.

[2] A. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

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

[4] D. Beeferman and A. Berger, "Agglomerative clustering of a search engine query log," in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 2000, pp. 407–416.

[5] A. Shepitsen, J. Gemmell, B. Mobasher, and R. Burke, "Personalized recommendation in social tagging systems using hierarchical clustering," in *Proceedings of ACM Conference on Recommender Systems*, 2008, pp. 259–266.

[6] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 200O.

[7] E. Fowlkes and C. Mallows, "A method for comparing two hierarchical clusterings," *Journal of the American Statistical Association*, vol. 78, no. 383, pp. 553–569, 1983.

[8] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[9] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[10] K. Wagstaff and C. Cardie, "Clustering with instance-level constraints," in *AAAI/IAAI*, 2000, p. 109.

[11] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrdl, "Constrained k-means clustering with background knowledge," in *Proceedings of International Conference on Machine Learning*, 2001, pp. 577–584.

[12] J. Yi, R. Jin, S. Jain, T. Yang, and A. Jain, "Semi-crowdsourced clustering: Generalizing crowd labeling by robust distance metric learning," in *Advances in Neural Information Processing Systems*, 2012, pp. 1772–1780.

[13] F. Li and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Proceedings of Computer Vision and Pattern Recognition*, 2005, pp. 524–531.

[14] I. Davidson and S. Ravi, "Clustering with constraints: Feasibility issues and the k-means algorithm," in *Proceedings of SIAM International Conference on Data Mining*, 2005, pp. 201–211.

[15] D. Pelleg and D.Baras, "K-means with large and noisy constraint sets," in *Proceedings of European Conference on Machine Learning*, 2007, pp. 674–682.

[16] M. Bilenko, S. Basu, and R. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Proceedings of International Conference on Machine Learning*, 2004, pp. 201–211.

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

[18] H. Liu, T. Liu, J. Wu, and D. T. amd Y. Fu, "Spectral ensemble clustering," in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 2015, pp. 715–724.

[19] H. Liu and Y. Fu, "Clustering with partition level side information," in *Proceedings of International Conference on Data Mining*, 2015.

[20] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall, "Computing gaussian mixture models with em using equivalence constraints," in *Advances in Neural Information Processing Systems*, 2004, pp. 465–472.

[21] T. Covoes, E. Hruschka, and J. Ghosh, "A study of k-means-based algorithms for constrained clustering," *Intelligent Data Analysis*, vol. 17, no. 3, pp. 485–505, 2013.

[22] H. Wu and Z. Liu., "Non-negative matrix factorization with constraints," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2010.

[23] K. Kamvar, S. Sepandar, K. Klein, D. Dan, M. Manning, and C. Christopher, "Spectral learning," in *Proceedings of International Joint Conference of Artificial Intelligence*, 2003.

[24] Q. Xu, M. Desjardins, and K. Wagstaff, "Constrained spectral clustering under a local proximity structure assumption," in *Proceedings of International Florida Artificial Inte Research Society Conference*, 2005.

[25] Z. Lu and M. Carreira-Perpinan, "Constrained spectral clustering through affinity propagation," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2008.

[26] X. Ji and W. Xu, "Document clustering with prior knowledge," in *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006.

[27] F. Wang, C. Ding, and T. Li, "Integrated kl (k-means-laplacian) clustering: A new clustering approach by combining attribute data and pairwise relations," in *Proceedings of SIAM International Conference on Data Mining*, 2009.

[28] T. Coleman, J. Saunderson, and A. Wirth, "Spectral clustering with inconsistent advice," in *Proceedings of International Conference on Machine learning*, 2008.

[29] Z. Li, J. Liu, and X. Tang, "Constrained clustering via spectral regularization," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2009.

[30] X. Wang, B. Qian, and I. Davidson, "On constrained spectral clustering and its applications," *Data Mining and Knowledge Discovery*, vol. 28, no. 1, pp. 1–30, 2014.

[31] A. Strehl and J. Ghosh, "Cluster ensembles — a knowledge reuse framework for combining partitions," *Journal of Machine Learning Research*, 2002.

[32] A. Fred and A. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 835–850, 2005.

[33] H. Liu, J. Wu, T. Liu, D. Tao, and Y. Fu, "Spectral ensemble clustering via weighted k-means: Theoretical and practical evidence," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 5, pp. 1129–1143, 2017.

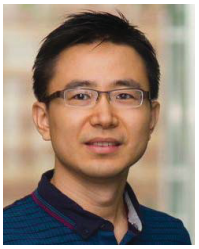
[34] H. Ayad and M. Kamel, "Cumulative voting consensus method for partitions with variable number of clusters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 1, pp. 160–173, 2008.

[35] C. Domeniconi and M. Al-Razgan, "Weighted cluster ensembles: Methods and analysis," *ACM Transactions on Knowledge Discovery from Data*, vol. 2, no. 4, 2009.

[36] H. Yoon, S. Ahn, S. Lee, S. Cho, and J. Kim, "Heterogeneous clustering ensemble method for combining different cluster results," *Data Mining for Biomedical Applications*, 2006.
[37] A. Topchy, A. Jain, and W. Punch, "Combining multiple weak clusterings," in *Proceedings of International Conference on Data Mining*, 2003.
[38] B. Mirkin, "Reinterpreting the category utility function," *Machine Learning*, 2001.
[39] A. Topchy, A. Jain, and W. Punch, "A mixture model for clustering ensembles," in *Proceedings of SIAM International Conference on Data Mining*, 2004.
[40] 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*, vol. 27, no. 1, pp. 155–169, 2015.
[41] J. Wu, H. Liu, H. Xiong, and J. Cao, "A theoretic framework of k-means-based consensus clustering," in *Proceedings of International Joint Conference on Artificial Intelligence*, 2013.
[42] H. Liu, J. Wu, D. Tao, Y. Zhang, and Y. Fu, "Dias: A disassemble-assemble framework for highly sparse text clustering," in *Proceedings of SIAM International Conference on Data Mining*, 2015.
[43] H. Liu, R. Zhao, H. Fang, F. Cheng, Y. Fu, and Y. Liu, "Entropy-based consensus clustering for patient stratification," *Bioinformatics*, vol. 33, no. 17, p. 26912698, 2017.
[44] H. Liu, T. Liu, J. Wu, D. Tao, and Y. Fu, "Spectral ensemble clustering," in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 2016.
[45] H. Liu, M. Shao, S. Li, and Y. Fu, "Infinite ensemble clustering," *Data Mining and Knowledge Discovery*, pp. 1–32, 2017.
[46] S. Yu and J. Shi, "Multiclass spectral clustering," in *Proceedings of IEEE International Conference on Computer Vision*, 2003.
[47] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. Wiley-Interscience, 2000.
[48] H. Fu, X. Cao, and Z. Tu, "Cluster-based co-saliency detection," *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 3766–3778, 2013.
[49] A. Joulin, F. Bach, and J. Ponce, "Discriminative clustering for image co-segmentation," in *Proceedings of IEEE Conference on Computer Vision and Patter Recognition*, 2010.
[50] ——, "Multi-class cosegmentation," in *Proceedings of IEEE Conference on Computer Vision and Patter Recognition*, 2012.
[51] Z. Tao, H. Liu, H. Fu, and Y. Fu, "Image cosegmentation via saliency-guided constrained clustering with cosine similarity," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2017.
[52] C. Rother, T. Minka, A. Blake, and V. Kolmogorov, "Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs," in *Proceedings of IEEE Conference on Computer Vision and Patter Recognition*, 2006.
[53] L. Mukherjee, V. Singh, and C. R. Dyer, "Half-integrality based algorithms for cosegmentation of images," in *Proceedings of IEEE Conference on Computer Vision and Patter Recognition*, 2009.
[54] D. S. Hochbaum and V. Singh, "An efficient algorithm for co-segmentation." in *Proceedings of IEEE Conference on Computer Vision*, 2009.
[55] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, "Interactively co-segmentating topically related images with intelligent scribble guidance," *Internationl Journal of Computer Vision*, vol. 93, no. 3, pp. 273–292, 2011.
[56] G. Kim and E. P. Xing, "On multiple foreground cosegmentation." in *Proceedings of IEEE Conference on Computer Vision and Patter Recognition*, 2012.
[57] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *ArXiv e-prints*, 2015.
[58] X. Cao, Z. Tao, B. Zhang, H. Fu, and W. Feng, "Self-adaptively weighted co-saliency detection via rank constraint." *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 4175–4186, 2014.
[59] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Su, "S-lic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.
[60] Y. Jia and M. Han, "Category-independent object-level saliency detection," in *Proceedings of IEEE Conference on Computer Vision*, 2013.
[61] H. Jiang, Z. Yuan, M.-M. Cheng, Y. Gong, N. Zheng, and J. Wang, "Salient object detection: A discriminative regional feature integration approach." *CoRR*, vol. abs/1410.5926, 2014.
[62] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2013.
[63] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection." in *Proceedings of IEEE Conference on Computer Vision and Patter Recognition*, 2013.
[64] S. Vicente, C. Rother, and V. Kolmogorov, "Object cosegmentation." in *Proceedings of IEEE Conference on Computer Vision and Patter Recognition*, 2011.
[65] J. C. Rubio, J. Serrat, A. M. Lpez, and N. Paragios, "Unsupervised co-segmentation through region matching." in *Proceedings of IEEE Conference on Computer Vision and Patter Recognition*, 2012.

Hongfu Liu received his bachelor and master degree in Management Information Systems from the School of Economics and Management, Beihang University, in 2011 and 2014 respectively. He is currently pursuing the Ph.D. degree in Northeastern University, Boston. His research interests generally focus on data mining and machine learning, with special interests in ensemble learning.

Zhiqiang Tao received the B.E. degree in software engineering from the School of Computer Software, and the M.S. degree in computer science from the School of Computer Science and Technology, Tianjin University, Tianjin, China, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree in Northeastern University, Boston. His current research interests include machine learning, computer vision, and multimedia analysis.



Yun Fu (S'07-M'08-SM'11) received the B.Eng. degree in information engineering and the M.Eng. degree in pattern recognition and intelligence systems from Xi'an Jiaotong University, China, respectively, and the M.S. degree in statistics and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, respectively. He is an interdisciplinary faculty member affiliated with College of Engineering and the College of Computer and Information Science at Northeastern University since 2012. His research interests are Machine Learning, Computational Intelligence, Big Data Mining, Computer Vision, Pattern Recognition, and Cyber-Physical Systems. He has extensive publications in leading journals, books/book chapters and international conferences/workshops. He serves as associate editor, chairs, PC member and reviewer of many top journals and international conferences/workshops. He received seven Prestigious Young Investigator Awards from NAE, ONR, ARO, IEEE, INNS, UIUC, Grainger Foundation; seven Best Paper Awards from IEEE, IAPR, SPIE, SIAM; three major Industrial Research Awards from Google, Samsung, and Adobe, etc. He is currently an Associate Editor of the IEEE Transactions on Neural Networks and Leaning Systems (TNNLS). He is fellow of IAPR, a Lifetime Senior Member of ACM and SPIE, Lifetime Member of AAAI, OSA, and Institute of Mathematical Statistics, member of ACM Future of Computing Academy, Global Young Academy (GYA), INNS and Beckman Graduate Fellow during 2007-2008.