2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Semi-supervised Clustering with Pairwise and Size Constraints

Shaohong Zhang, Hau-San Wong, and Dongqing Xie

***Abstract*— In recent years, semi-supervised clustering receives considerable attention in the pattern recognition and data mining communities. This type of clustering algorithms takes advantage of partial prior knowledge, and significant improved performance beyond traditional unsupervised clustering algorithms is observed. In general, the partial prior knowledge is mainly in the form of pairwise constraints, which specify whether point pairs should be in the same cluster or in different clusters. Moreover, some other forms of constraints also attract research interests, for example, the balance constraint or the size constraint. However, it is also important to consider different types of constraints simultaneously, since different types of prior knowledge might have their own bias when considered separately. In this paper, we propose an improved algorithm to incorporate the pairwise and size constraints into a unified framework. Experiments on several benchmark data sets demonstrate that the proposed unified algorithm outperforms previous approaches under a variety of different conditions, which demonstrates that judicious integration of different types of constraints can result in improved performance than in those cases where only a single kind of constraint is used.**

I. Introduction

Recently, partial knowledge of ground truth label information is integrated into different clustering methods and is widely reported to improve performance greatly [1]–[4]. This new type of semi-supervised clustering methods is usually referred to as constrained clustering. In general, the clustering constraints are often in the form of pairwise constraints between some of the data point pairs. The pairwise constraint is usually of two types, the Must-Link (ML) constraint and the Cannot-Link (CL) constraint [1]: for two data points $\mathbf{x}$ and $\mathbf{y}$, a Must-Link (ML) constraint $ML(\mathbf{x}, \mathbf{y})$ requires that these two data points must be assigned to the same class, while a Cannot-Link (CL) constraint $CL(\mathbf{x}, \mathbf{y})$ requires them to be assigned to two different classes. A number of constrained clustering algorithms have recently been proposed to take advantage of these two kinds of pairwise constraints. Most of these algorithms focus on the Kmeans algorithm, since Kmeans is one of the best-known and most widely used algorithms, and extension work on this algorithm has attracted great interests [5], [6]. For example, COPKmeans [1] tries to satisfy each constraint, in addition to considering the distance of each data point to the cluster centroids. Partial Constrained Kmeans (PCKmeans) [4] and Partial closure-based constrained Kmeans (PCCKmeans) [2] optimize corresponding cost functions which take into consideration not only distances but also constraints. On the other hand, several authors recently introduce another new constraint, the size constraint, which emphasizes the size distribution of the resulting clusters. Here the size of a cluster refers to the total number of data points in the cluster, and class distribution refers to the set of cluster sizes in the clustering solution. The most active issue of this topic is to achieve balanced clustering, which aims to attain a clustering result, with the constraint that all of the clusters have comparable sizes [3]. Various approaches are proposed to obtain balanced clusters, e.g. by including the balance criterion in the clustering formulation via graph partitioning [7], [8], by adding balance constraints within an optimization framework [9], or by using the class of frequency sensitive competitive learning methods which penalizes clusters with large sizes [10]. However, imbalanced data sets are also widely observed in practical applications. Here "imbalanced data sets" refers to those data sets where the number of data points are significantly different across the classes. For example, when performing classification in medical diagnosis, the number of data points in the disease class will be much smaller than that in the normal class. Xiong et al. points out that Kmeans tends to produce clusters with similar sizes, even if the original data have clusters of different sizes [11]. In contrast with the balanced data set case, few works have been performed to impose size constraints for imbalanced data sets. To our best knowledge, only a recent work [12] has addressed this problem by introducing constraints on the number of data points per cluster based on the fuzzy clustering framework.

Although clustering might benefit from either the pairwise constraints or the size constraints, there are no previous works to our knowledge which consider the integration of these two types of constraints into the clustering framework simultaneously. Moreover, different types of constraints might have their own bias to the resulting solution, and in some cases these might cause deterioration of the clustering result when considered separately. Several researches provide empirical results which show that pairwise constraints are not always good for clustering [13], [14]. On the other hand, a practical problem for imposing the size constraint is how to minimize the difference between the class distribution of the ground truth and that of the clustering solution, and there are no ideal solutions to handle this problem. This problem will be more crucial in imbalanced data sets. For example, assume the size constraint for a data set with three classes is $\{20, 40, 80\}$ (i.e., there are three classes whose

Shaohong Zhang and Dongqing Xie are with the Department of Computer Science, Guangzhou University, Guangzhou, P.R. China (email: zimzsh@gmail.com); Hau-San Wong is with the Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Hong Kong, P.R. China (email: cshswong@cityu.edu.hk)

The work was partially supported by a grant from National Natural Science Foundation of China [No. 61202273], a grant from Natural Science Foundation of Guangdong Province [No. S2012040007206], a grant from Department of Education in Guangdong province [No. 2013KJCX0144], a grant from Guangzhou Education Bureau Science Foundation for Yangcheng Scholars (Project No. 10A033D), and a grant from the City University of Hong Kong [No. 7004047].

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

sizes are $20, 40$ and 80 respectively) and a clustering solution has three clusters $\{A, B, C\}$, we need to suitably enforce the size constraint to control the actual size distribution of the clusters. Otherwise, the clustering result will not be satisfactory. This is an important problem in clustering which is difficult to solve without any supervision information. In view of these questions, in this paper we propose to integrate the pairwise constraints and the size constraints simultaneously into clustering within an optimization framework. Specifically, we perform the clustering assignment taking into account not only the distance cost but also the cost of pairwise constraint violations, and we penalize the inconsistency between the size distribution of the clustering solution and the size constraints.

Another related problem we consider is the ordering of the data points in constrained clustering. Although random ordering is adopted in the original works performed in [1] [4], there were some recent observations that suitably ordering the constrained points with a number of heuristics might improve the clustering performance. For example, Davidson et al. propose the q-inductive ordering approach to identify and generate easy constraint sets [15]. Zhang et al. extend this idea [15] to rank the order of the constrained points using the Cannot-Link constraint degrees and the sizes of closures in [2]; Hong et al. propose to rank the data points according to their clustering uncertainties, which are calculated by using ensembles of multiple clustering algorithms in [16]. Different from these works, in this paper we propose a new criterion, in which the size constraint is used, in addition to the pairwise constraints, in determining the priority of assignment of each data point to different clusters. This criterion is more effective since it takes advantage of the extra information, i.e., the size constraint, which is available. In addition, the pairwise constraints will help to adjust the clusters and consequently refine the boundaries according to the size constraint. Benefiting from the two kinds of constraints, a more reliable clustering solution will be found.

To evaluate our algorithm, we also consider the scenario with active (pairwise) constraints. In general, most of the existing constrained clustering approaches use constraints chosen at random, i.e., by randomly selecting point pairs and then querying whether these point pairs come from the same class. This random selection approach is in most cases neither effective nor efficient. As an alternative, some recent researches [4] [17] focus on the adoption of active constraints, which are usually selected using an active mechanism, and corresponding advantages are reported. On the other hand, a common assumption on pairwise constraints is that they are intrinsically correct. However, this assumption is not always possible to be guaranteed in practice.

This paper therefore makes two main contributions. First, our proposed algorithm, PCKmeans with Size constraint (PCS), provides empirical evidence that combination of different kinds of partial information might improve the performance in constrained clustering. Second, to our best knowledge, PCS is the first attempt to incorporate the pairwise constraints and the size constraints simultaneously. Experimental results on several benchmark data sets demonstrate the advantages of our algorithm over the two baseline constrained algorithms, KmeansS (Kmeans with the Size constraint) which use the size constraint only, and PCKmeans which use the pairwise constraint only.

II. Proposed algorithms

In this section, we first briefly describe the two basic algorithms, i.e., Kmeans and PCKmeans algorithms. In what follows, we extend these two basic algorithms with the size constraints.

A. Kmeans and PCKmeans

Formally, given a data set $X = \{\mathbf{x}_i\}_{i=1}^N$, the traditional unsupervised clustering algorithm Kmeans [18] searches for a disjoint k partition $\{X_h\}_{h=1}^k$ (with an associated centroid $\boldsymbol{\mu}_h$) of X such that the following cost function is minimized:

$$J_{KM} = \sum_{h=1}^{k} \sum_{\mathbf{x}_i \in X_h} \|\mathbf{x}_i - \boldsymbol{\mu}_h\|^2 \tag{1}$$

Kmeans aims to find the point assignment such that the total distance between the points and their associated centroids are minimized. In general, the cost function is minimized when

$$\frac{\partial J_{KM}}{\partial \boldsymbol{\mu}_h} = -2 \sum_{\mathbf{x}_i \in X_h} (\mathbf{x}_i - \boldsymbol{\mu}_h) = 0 \tag{2}$$

The centroids are then computed as follows:

$$\boldsymbol{\mu}_h = \frac{\sum_{\mathbf{x}_i \in X_h} \mathbf{x}_i}{|X_h|} \tag{3}$$

where $|X_h|$ is the cardinality of the cluster X_h.

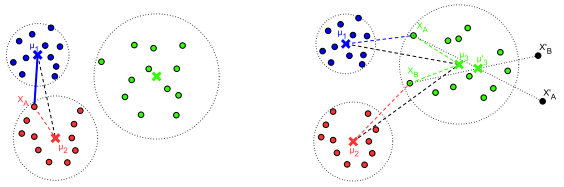
PCKmeans introduces a soft form of optimization to compromise between the degree of satisfaction of the various constraints and the minimization of the distances between the data points and their associated cluster centroids [4]. Specifically, given a data set X, a set of ML constraints S, a set of CL constraints D, the corresponding penalty weights $\phi_{ij}(\varphi_{ij})$ for violating ML(CL) constraints and the number k of clusters, PCKmeans aims to find a disjoint k partition $\{X_h\}_{h=1}^k$ (each with its associated centroid $\boldsymbol{\mu}_h$) so as to minimize the following cost function

$$\begin{aligned} J_{PC} = & \frac{1}{2} \sum_{h=1}^{k} \sum_{\mathbf{x}_i \in X_h} \|\mathbf{x}_i - \boldsymbol{\mu}_h\|^2 \\ & + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in S} \phi_{ij} \delta(L(\mathbf{x}_i) \neq (L(\mathbf{x}_j)) \\ & + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in D} \varphi_{ij} \delta(L(\mathbf{x}_i) = (L(\mathbf{x}_j)) \end{aligned} \tag{4}$$

where $\delta()$ is the indicator function defined as follows:

$$\delta(true) = 1, \delta(false) = 0. \tag{5}$$

and $L(\mathbf{x}_i)$ denotes the estimated cluster label for point $\mathbf{x}_i$. PCKmeans uses a greedy search technique as Kmeans does. Specifically, it minimizes Eq. (4) so as to assign points to



(a) NO points (b) FI and VFI points

Fig. 1. Illustration of nearest out-class (NO) points, farthest in-class (FI) points and virtual paired FI (VFI) points : (a) NO points of the smaller class with cluster centroid $\boldsymbol{\mu}_1$: the NO cost function for $\boldsymbol{\mu}_1$ and point $\mathbf{x}_A$ can be computed as $D_{NO}(\mathbf{x}_A, \boldsymbol{\mu}_1) = \frac{d(\mathbf{x}_A,\boldsymbol{\mu}_1)[d(\mathbf{x}_A,\boldsymbol{\mu}_1)+d(\mathbf{x}_A,\boldsymbol{\mu}_2)]}{d(\mathbf{x}_A,\boldsymbol{\mu}_1)d(\mathbf{x}_A,\boldsymbol{\mu}_2)}$; (b) FI points of the larger class with cluster centroid $\boldsymbol{\mu}_3$: The FI cost function for $\boldsymbol{\mu}_3$ and the point $\mathbf{x}_A$ can be computed as $D_{FI}(\mathbf{x}_A, \boldsymbol{\mu}_1) = \frac{d(\mathbf{x}_A,\boldsymbol{\mu}_1)[d(\mathbf{x}_A,\boldsymbol{\mu}_1)+d(\mathbf{x}_A,\boldsymbol{\mu}_3)]}{d(\mathbf{x}_A,\boldsymbol{\mu}_1)d(\mathbf{x}_A,\boldsymbol{\mu}_3)}$. $\mathbf{x}'_A$ and $\mathbf{x}'_B$ are the virtual paired FI (VFI) points of $\mathbf{x}_A$ and $\mathbf{x}_B$ respectively.

their corresponding clusters, and then computes new cluster centroids for the clusters in each iteration. Note that the final performance of PCKmeans is sensitive to the assignment order of the constrained points, and a random order is adopted in the original paper [4].

B. Kmeans with size constraints (KmeansS)

For classes $\{X_h\}_{h=1}^k$, let E_h denote the corresponding prior knowledge of the class size of the h-th cluster, and F_h denote its class size, The first problem for incorporating the size constraints is to find a suitable alignment of the sizes $\{E_h^{'}\}_{h=1}^k$ from $\{E_h\}_{h=1}^k$ for $\{F_h\}_{h=1}^k$. Intuitively, we would like to align the sizes so as to minimize the following difference

$$J_{align} = \sum_h |F_h - E_h^{'}| \tag{6}$$

This problem can therefore be solved when the two class size distributions are aligned with their sorted sizes. Let $S(p_h)$ be the index of E_h in the ascending sorted list of $\{E_h\}_{h=1}^k$, the corresponding alignment for F_h is

$$E_h^{'} = E_g, \text{where } S(E_g) = S(F_h) \tag{7}$$

The size difference U_h for class X_h is thus

$$U_h = F_h - E_h^{'} \tag{8}$$

We further consider the above size difference as follows: We first measure the size difference from the perspective of an overall size distribution, i.e., between the prior class size distribution $\mathbf{p} = \{p_h : p_h = E_h/N\}_{h=1}^k$ and the current distribution $\mathbf{q} = \{q_h : q_h = F_h/N\}_{h=1}^k$, with the Jensen-Shannon divergence [19]:

$$JSD(\mathbf{p}, \mathbf{q}) = \frac{1}{2}KL(\mathbf{p}, \mathbf{q}) + \frac{1}{2}KL(\mathbf{q}, \mathbf{p}) \tag{9}$$

where $KL(\mathbf{p}, \mathbf{q})$ is the Kullback-Leibler divergence [20] for $\mathbf{p}$ and $\mathbf{q}$

$$KL(\mathbf{p}, \mathbf{q}) = \sum_h p_h \log \frac{p_h}{q_h} = \sum_h \frac{E_h}{N} \log \frac{E_h}{F_h} \tag{10}$$

The overall size divergence cost is defined as

$$J_A = JSD(\mathbf{p}, \mathbf{q}) * N \tag{11}$$

On the other hand, we also consider the size difference within each class pair. A penalty J_S for clusters which are too small is imposed when the sizes of the individual classes are smaller than expected. Similarly, a penalty J_L for clusters which are too large is imposed when their sizes are larger than expected. Therefore, taking into consideration these size costs, the original cost function in Eq.(1) is modified as follows

$$J_{KMS} = J_{KM} + \alpha J_A + \beta J_S + \gamma J_L \tag{12}$$

where α, β, γ are the corresponding non-negative scale parameters which represent different weights for the different penalty functions for the cluster sizes.

When $U_h < 0$, i.e., class X_h is smaller than expected, we must increase the cost function with the penalty for small-sized clusters. We use the set of nearest out-class points from other classes to determine this penalty. The point $\mathbf{x}$ is called a nearest out-class (NO) point for class X_h if it minimizes the following function, which we call the NO cost function

$$D_{NO}(\mathbf{x}, \boldsymbol{\mu}_h) = \frac{d(\mathbf{x}, \boldsymbol{\mu}_h)[d(\mathbf{x}, \boldsymbol{\mu}_h) + d(\mathbf{x}, \boldsymbol{\mu}_j)]}{d(\mathbf{x}, \boldsymbol{\mu}_j)d(\boldsymbol{\mu}_h, \boldsymbol{\mu}_j)}, \quad \forall \mathbf{x} \in X_j, j \neq h \tag{13}$$

where

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| \tag{14}$$

is the Euclidean distance function between two points $\mathbf{x}$ and $\mathbf{y}$.

The nearest out-class point sets for all classes are then used to measure the small-size penalty as follows:

$$J_S = \sum_{h=1}^{k} \sum_{\mathbf{x}_i \in NO(X_h)} \|\mathbf{x}_i - \boldsymbol{\mu}_h\|^2 \tag{15}$$

It is easy to observe that the penalty J_S for class X_h will force the centroid $\boldsymbol{\mu}_h$ to move toward the set of nearest out-class points $NO(X_h)$.

When $U_h > 0$, i.e., X_h is larger than expected, we also need to increase the cost function with the penalty for large-sized clusters. We first find the set of farthest in-class points for each class. The point $\mathbf{x}$ is called a farthest in-class (FI) point if it minimizes the following function

$$D_{FI}(\mathbf{x}, \boldsymbol{\mu}_j) = \frac{d(\mathbf{x}, \boldsymbol{\mu}_j)[d(\mathbf{x}, \boldsymbol{\mu}_h) + d(\mathbf{x}, \boldsymbol{\mu}_j)]}{d(\mathbf{x}, \boldsymbol{\mu}_h)d(\boldsymbol{\mu}_h, \boldsymbol{\mu}_j)}, \quad \forall \mathbf{x} \in X_h, j \neq h \tag{16}$$

We use $FI(X_h)$ to denote the $|U_h|$ farthest in-class points for class X_h. In contrast to the penalty for small-sized clusters, the penalty for large-sized clusters will result in a movement of the centroid away from the set of farthest in-class (FI) points $FI(X_h)$. Considering Eq.(2) and Eq.(3), the penalty for large-sized clusters cannot be directly measured using the FI points $FI(X_h)$ (otherwise, the centroids will be forced to move toward the farthest in-class points, which is

opposite to the original objective). In this case, we use the virtual paired FI (VFI) points. For a FI point $\mathbf{y} \in FI(X_h)$ in class X_h, a VFI point $\mathbf{z}$ is defined as

$$\mathbf{z} = 2\frac{\sum_{x \in \{X_h - FI(X_h)\}} \mathbf{x}}{|\{X_h \setminus FI(X_h)\}|} - \mathbf{y} \tag{17}$$

where $\{X_h \setminus FI(X_h)\}$ is the point set for class X_h when the FI points $FI(X_h)$ are removed. Similarly, we use $VFI(X_h)$ to denote the U_h VFI points for class X_h. The penalty for large-sized clusters can then be defined as

$$J_L = \sum_{h=1}^{k} \sum_{\mathbf{x}_i \in VFI(X_h)} \|\mathbf{x}_i - \boldsymbol{\mu}_h\|^2 \tag{18}$$

Illustrative examples of the NO points, FI points and VFI points are shown in Figure 1: In Figure 1(a), the cluster with centroid $\boldsymbol{\mu}_1$ has a smaller number of points than expected. We might search for NO points to improve the size of the cluster; in Figure 1(b), the cluster with centroid $\boldsymbol{\mu}_3$ has a larger number of points than expected. We might search for VFI points to reduce the size of the cluster.

Similar to Kmeans, the KMeansS centroids can be computed by setting the partial derivative to zero. The centroid $\boldsymbol{\mu}_h$ can be computed as

$$\boldsymbol{\mu}_h = \frac{\sum_{\mathbf{x}_i \in X_h} \mathbf{x}_i + \beta \sum_{\mathbf{x}_i \in NO(X_h)} \mathbf{x}_i + \gamma \sum_{\mathbf{x}_i \in VFI(X_h)} \mathbf{x}_i}{|X_h| + \beta|NO(X_h)| + \gamma|VFI(X_h)|} \tag{19}$$

where $NO(X_h)$ and $VFI(X_h)$ can be computed from Eq.(13) and Eq.(17) respectively.

From Eq.(19), we can see that the centroids will be forced toward the nearest out-class points (e.g., $NO(X_h)$) for smaller classes, or toward the VFI points (e.g., $VFI(X_h)$) and thus away from the FI points (e.g., $FI(X_h)$).

From the above, we can see that an important condition for the KmeansS algorithm is to find a moderately good initialization assignment. Otherwise the alignment between the current assignment and the size constraint will not be satisfactory, which will in turn result in a reduction of its performance. We use a simple filter-refinement scheme to address the problem. Specifically, we apply Kmeans to obtain the initialization assignment, and then perform further iterations to refine the solution. The complete KmeansS algorithm is summarized in Figure 2.

C. PCKmeans with size constraints (PCS)

To make use of both pairwise and size constraints, the natural generalization from KmeansS and PCKmeans is to unify their cost functions Eq. (12) and Eq. (4). The resulting cost function can therefore be formulated as

$$J_{PCS} = J_{PC} + \alpha J_A + \beta J_S + \gamma J_L \tag{20}$$

Also, if we use the filter-refinement scheme, we will first use PCKmeans for clustering the data with pairwise constraints, and then perform clustering with both kinds of constraints.

An interesting benefit from the size constraints is that it is now possible to detect the points which will be transferred from larger clusters to smaller clusters through the movement of the cluster centroids. Therefore, when considered in the context of pairwise constraints, the constrained points within the set of transferred points shall have a smaller priority than the other constrained points, since they are less stable in terms of which cluster they are assigned to. In other words, they are closer to the boundaries of the current set of clusters. In PCS, random values r sampled from a uniform distribution are assigned to individual constrained points as their priority values, and a penalty threshold value t is imposed on those points which belong to the set of transferred points

$$pr(\mathbf{x}_i) = \begin{cases} r & \text{if } \mathbf{x}_i \in X_C \text{ and } \mathbf{x}_i \notin X_T \\ r + t & \text{if } \mathbf{x}_i \in X_C \text{ and } \mathbf{x}_i \in X_T \end{cases} \tag{21}$$

where

$$X_C = \{\mathbf{x}_i : \mathbf{x}_i \in M \text{ or } \mathbf{x}_i \in C\} \tag{22}$$

is the constrained point set, and

$$X_T = \{\mathbf{x}_i : \mathbf{x}_i \in \bigcup_h NO(X_h) \text{ or } \mathbf{x}_i \in \bigcup_h FI(X_h)\} \tag{23}$$

is the transferred point set. The constrained points are thus assigned to their clusters in ascending order of their priority values. The smaller their priority values are, the greater the possibility that they will be considered in advance. The complete PCKmeansS algorithm is summarized in Figure 3.

Algorithm 1: KmeansS
INPUT:
data set X,
cluster number k_s;
prior knowledge of the size distribution $\{E_1, E_2, \cdots, E_k\}$;
maximum iteration T;
OUTPUT:
clustered label result Y of X.
METHOD:
1. initialize cluster centroids $\{\boldsymbol{\mu}_i\}_{i=1}^{k_s}$ at random or otherwise;
2. if the filter-refinement scheme is used, perform Kmeans with $\{\boldsymbol{\mu}_i\}_{i=1}^{k_s}$;
3. **Repeat**
4. assign label Y of points to their closest centroids;
5. update centroids according to Eq. (19);
6. **Until** Y converges or maximum iteration T reaches
7. return Y.

Fig. 2. The KmeansS algorithm

III. Experiments

A. Data sets

We apply our proposed approach to a number of public data sets. The first two are 2-D artificial data sets, Petals and Half-ring, which can be downloaded from the web [1]. We also perform experiments on several public benchmark data sets obtained from the well-known UCI machine learning repository[2], including Iris, Wine, Balance Scale, and Digits. The Digits data set is constructed by extracting the numerals 0, 7 and 8 from the Pen-Based Recognition of Handwritten

[1] http://www.bangor.ac.uk/m̃as00a/activities/artificial_data.htm
[2] http://archive.ics.uci.edu/ml/

Algorithm 2: PCS
INPUT:
data set X,
cluster number k_s;
must-link constraints S and cannot-link constraints D;
prior knowledge of the size distribution $\{E_1, E_2, \cdots, E_k\}$;
maximum iteration T;
OUTPUT:
clustered label result Y of X.
METHOD:
1. initialize clusters centroids $\{\boldsymbol{\mu}_i\}_{i=1}^{k_s}$ at random or otherwise;
2. if the filter-refinement scheme is used, perform PCKmeans with $\{\boldsymbol{\mu}_i\}_{i=1}^{k_s}$, S and D;
3. initialize the priority values of the constrained points pr at random;
4. initialize last cost $J_{last} = +\infty$;
5. **Repeat**
6. order the constrained points in ascending order of pr;
7. assign label Y to the points according to Eq.(4);
8. compute the cost function J according to Eq. (20);
9. **if** $J < J_{last}$
10. update centroids according to Eq. (19);
11. $J_{last} = J$;
12. update pr according to Eq. (21);
13. **else**
14. update pr with random values sampled from a uniform distribution;
15. **Until** Y converges or maximum iteration T reaches
16. return Y.

Fig. 3. The PCS algorithm

Digits as in [21]. These data sets are widely used in evaluating clustering or constrained clustering algorithms, such as in [21]–[25].

B. Evaluation Measures

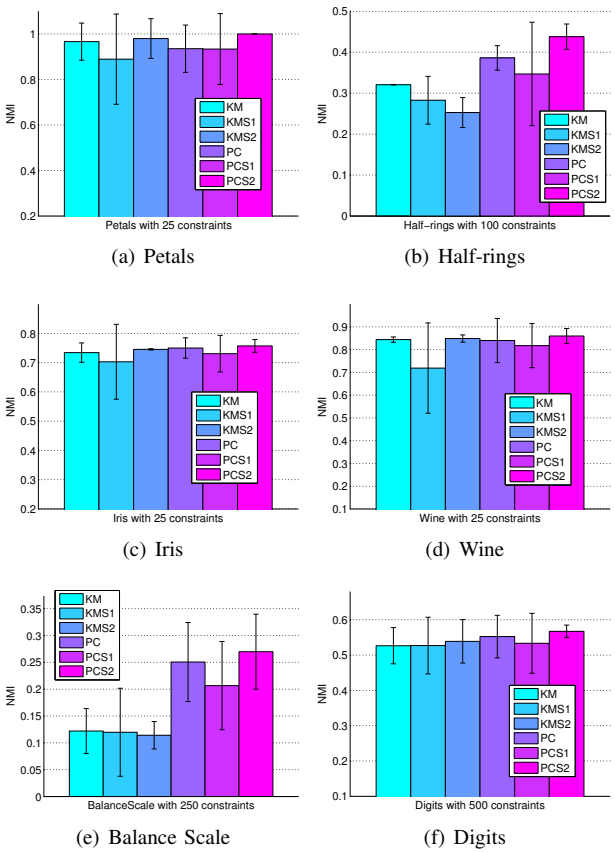
We use Normalized Mutual Information (NMI) [26] as the main measure to evaluate the performance of the proposed approach. It will also be important to evaluate how well the final clustering solution agree with both the aligned size constraints and the ground truth label. To our best knowledge, there is not yet a standard measure for this purpose. As a result, we introduce a new measure, Alignment Score (AS), in addition to the standard NMI. Specifically, for a clustering solution $X = \{X_h\}_{h=1}^{k}$ with its ground truth label Y and its aligned size constraints $\mathbf{E}' = \{E_h^{'}\}_{h=1}^{k}$, we first detect the dominant class for each cluster, say P_h (i.e., the points from the class P_h form the majority in cluster X_h). Denoting these points from each dominant class as $V_h = \{\mathbf{x} : \mathbf{x} \in P_h \text{ and } \mathbf{x} \in X_h\}$, AS is defined as follows

$$AS(X, Y, \mathbf{E}') = \frac{|Y|}{k} \sum_{h=1}^{k} \frac{|V_h|/|P_h|}{|E_h^{'} - |V_h|| + ||P_h| - |V_h|| + 1} \tag{24}$$

where $|Y|$ is the cardinality of a set Y, and $|p_h^{'} - |V_h||$ represents the absolute value of the difference between $p_h^{'}$ and $|V_h|$. Higher values of AS will result if the majority class of a cluster is more dominant . For a perfect clustering solution, AS will be equal to $|Y|$, the number of points in the data set.

C. Methodology

Note that the performance of the Kmeans-like algorithms is sensitive to their initialization conditions. Therefore, to have a fair comparison, we use the same random initialization conditions for all the algorithms, i.e., the same initialized set of centroids. Corresponding parameters for the different algorithms are set to identical values: the penalty weights for violating ML/CL pairwise constraints are set to the square root of the number of dimensions D (i.e., $\sqrt{D}$), the three parameters α, β, γ for size constraints are set to 1. The penalty threshold value t in Eq. (21) is set to 0.8. The maximum iteration number T is set to 400 for all the algorithms. The incurred penalty weights ϕ_{ij} and φ_{ij} are set to 1 according to the setting for UCI data sets in the original paper [4]. Forty independent trials for each experiment are conducted and the mean results are reported.

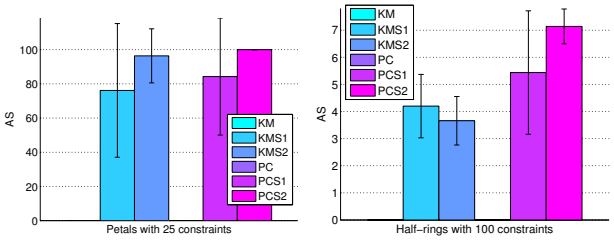


(a) Petals (b) Half-rings (c) Iris (d) Wine (e) Balance Scale (f) Digits

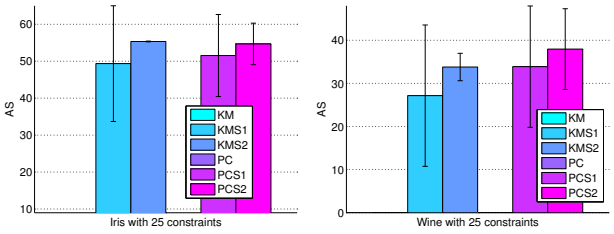
Fig. 4. Performances under moderate number of constraints: Normalized Mutual Information (NMI).

D. Results and Discussion

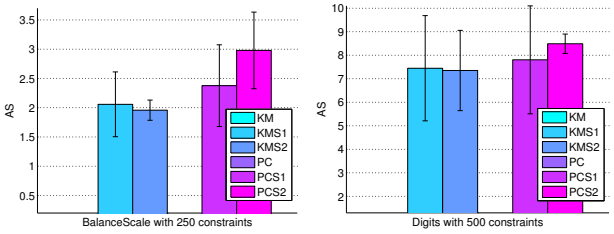
1) Performances under moderate number of constraints: First, we compare our algorithms with the standard unsupervised Kmeans algorithm, and the partial constrained PCKmeans algorithm under a moderate number of constraints. We also compare our algorithms with the alternative versions without filter-refinement. As a result, six algorithms are compared in this experiment and their performances are shown in Figure 4: Kmeans (KM), KmeansS without filter-refinement (KMS1), KmeansS with filter-refinement scheme (KMS2), PCKmeans (PC), PCS without filter-refinement scheme (PCS1), and PCS with filter-refinement scheme



(a) Petals (b) Half-rings

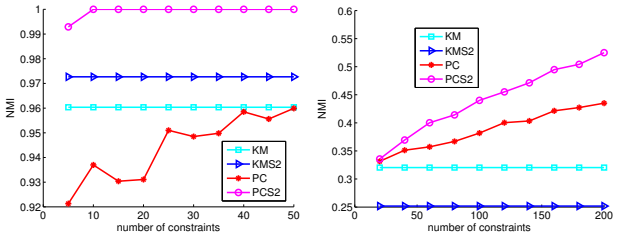


(c) Iris (d) Wine

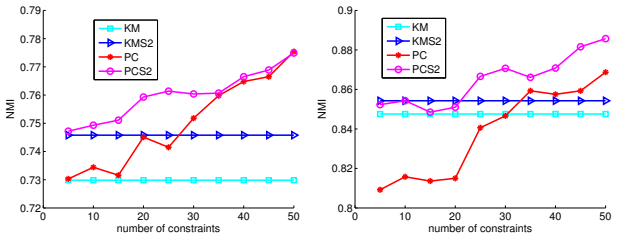


(e) Balance Scale (f) Digits

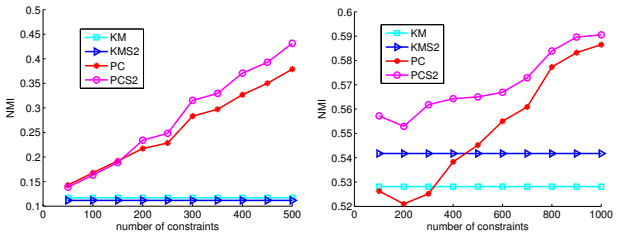
Fig. 5. Performances under moderate number of constraints: Alignment Score (AS).



(a) Petals (b) Half-rings



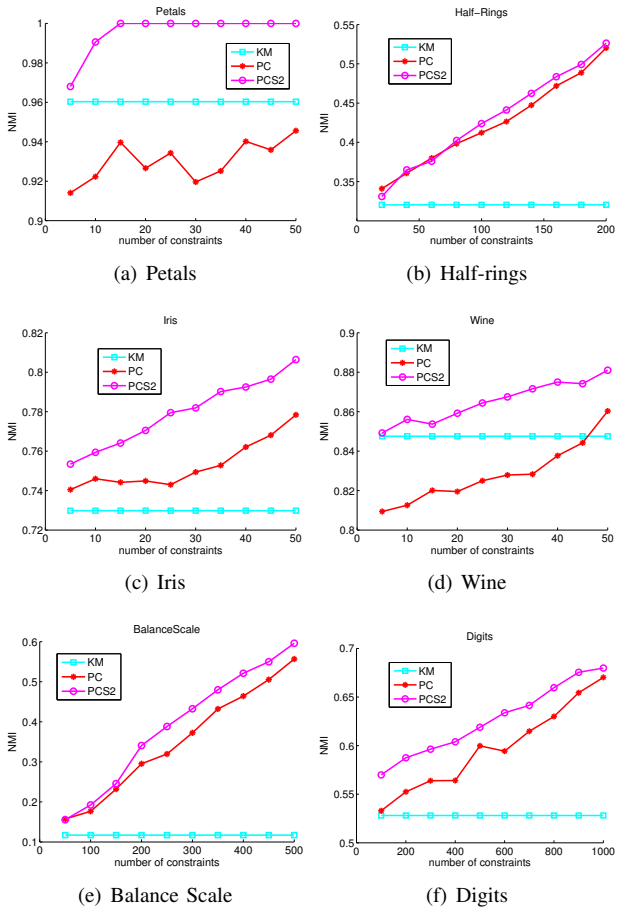
(c) Iris (d) Wine



(e) Balance Scale (f) Digits

Fig. 6. Performances under different numbers of constraints.

(PCS2). Mean and standard deviation values of Normalized Mutual Information (NMI) and Alignment Score (AS) for each algorithm are reported in the form of error bar plots in Figure 4 and Figure 5 respectively. Note that the size constraints are not used in Kmeans (KM) and PCKmeans (PC), so the corresponding bars for these two algorithms do not appear in Figure 5. From the resulting figures, we can obtain several interesting observations: (1) In terms of the mean NMI values, clustering with only the size constraints, i.e., KMS1 and KMS2 in Figure 4, shows little improvement of clustering quality over the standard unsupervised clustering algorithm Kmeans. For the imbalanced Half-rings data set, the results of KMS1 and KMS2 are even less satisfactory than KMeans. On the other hand, we can observe that a clustering algorithm which uses the pairwise constraints, e.g., PCKmeans (PC in Figure 4), performs in general better than Kmeans except for the case of the Petals data set. These observations show that the pairwise constraints are in general more useful than the size constraints, but clustering with only a single kind of constraint does not always outperform the standard unsupervised Kmeans. For all the cases, our new algorithm with the filter-refinement scheme, i.e., PCS2, which incorporates both pairwise constraints and size constraints, is consistently better than Kmeans. This suggests that it is meaningful to consider different kinds of constraints simultaneously and suitable combinations of this partial information might achieve a better solution. (2) It is interesting to observe that, for the two imbalanced data sets, Half-rings and Balance Scale, PCKmeans outperforms Kmeans significantly, while for the other four balanced data sets, the improvement of PCKmeans over Kmeans is smaller than those in the two imbalanced data sets. We believe that this is due to the tendency of Kmeans to produce balanced results, while PCKmeans could find clustering solutions which are relatively more imbalanced due to the pairwise constraints. Another interesting observation is that the size constraint in KMS1 and KMS2 do not provide significant contributions. From Figure 5, we can also observe that PCS1 and PCS2 have higher Alignment Score (AS) values than KMS1 and KMS2 respectively. We believe that the reason lies in the uncertain alignment of the current clustering solution with the size constraints, and a better alignment with these constraints may improve the result. However, in PCS2, which is initialized with PCKmeans, the pairwise constraints are made use of to provide a comparatively better clustering solution in the filtering step, such that it can benefit more from the size constraints. Therefore, we can observe that PCS2 can be applied to different kinds of data sets, either balanced or imbalanced. It is also notable to observe that the results based on PCS2 in general have smaller standard deviations than PCKmeans (PC in Figure 4), which suggests



(a) Petals (b) Half-rings

(c) Iris (d) Wine

(e) Balance Scale (f) Digits

Fig. 7. Performances under different numbers of active constraints.

that it might be more stable against different initialization conditions. (3) We also study the different performances of the algorithms with and without filter refinement, and we observe that, with the filter-refinement scheme, KMS2 is generally better than KMS1 for all the data sets except for the Half-rings data set, while PCS2 is consistently better than PCS1 for all the cases. Specifically, while PCS2 is consistently better than PC for all the cases, PCS1 is less satisfactory than PC. We believe that the main reason is due to the capability of the filter-refinement scheme to search for better initial centroids. On the other hand, if filter-refinement is not used, the search space will be limited by both the pairwise constraints and size constraints at the same time. In general, we can see that the filter-refinement scheme plays an important role in improving the performance of our algorithm. (4) Finally we study the clustering results based on the Alignment Score (AS) in Figure 5, and we obtain similar conclusions as in Figure 4. Specifically, the adoption of the filter-refinement scheme will result in better AS values with pairwise constraints, i.e., PCS2 is consistently better than PCS1. In addition, PCS2 tends to have better results than others. However, it is interesting to observe a number of different results. Although PCS1 results in better AS values than KMS2, as shown in Figure 5(f), it does not outperform KMS2 in terms of NMI, as shown in Figure 4(f). Also, KMS2 has less satisfactory AS values when compared with KMS1 in Figure 5(f), while it is better than KMS1 in terms of NMI in Figure 4(f). These observations show that the different measures serve to complement each other in characterizing the clustering results.

2) Performance under different number of constraints: After performing investigation on the performance of the observed algorithms with a moderate number of constraints, it is also important to see their performance under different number of constraints. Since the KMS2 and PCS2 algorithms with the filter-refinement scheme are in general better than their non-filter-refinement versions (i.e., KMS1 and PCS1), we do not consider KMS1 and PCS1 further in these experiments. NMI results for the four algorithms on the various data sets are shown in Figure 6: Kmeans (KM), KMS2, PCKmeans (PC) and PCS2. Note that Kmeans and KMS2 are independent of constraints, so we use their mean values in all the cases, which results in two horizontal lines in Figure 6. Several interesting observations can be obtained from Figure 6: (1) As in the previous experiments in Figure 4, we can observe that KMS2 and PCKmeans may not be better than the standard Kmeans for all the cases. Specifically, for the Half-rings data sets, the KMS2 curve is below that of Kmeans. PCKmeans and PCS2 tend to improve their performances correspondingly with increasing numbers of constraints. For the data sets of Petals, Wine and Digits, when the number of constraints is very small, the results based on PCKmeans are even less satisfactory than those of Kmeans, while PCS2 has comparable or better performances as those of Kmeans and KMS2. In addition, PCS2 improves its performance consistently with increasing number of constraints and it outperforms all the other competing algorithms. Interestingly, we can also observe that PCKmeans performs as good or even slightly better than PCS2 when the number of constraint becomes very large. This can be explained by the capability of a large number of constraints to restrict the sizes of different clusters in balanced data sets. In general, the number of available constraints will not usually be too large for most clustering problems, and within this range of constraint numbers, PCS2 performs better or sometimes at least as good as the other algorithms. (2) Another interesting observation is that PCS2 and PCKmeans tend to perform much better than Kmeans in imbalanced data sets, e.g., Half-rings and Balance Scale. In addition, with a large number of constraints, PCS2 tends to benefit from the size constraints and outperforms PCKmeans in imbalanced data sets. This suggests that incorporating different kinds of constraints might be a better alternative for imbalanced data sets rather than using a single kind of constraint.

3) Performance under different numbers of active constraints: We also conduct experiments based on different numbers of active pairwise constraints. In this experiment, we use the Explore and Consolidate approach (EC) [4] to select active constraints. For a particular data set with k classes, the EC approach is used to find a disjoint k

closure (a closure is a point set belonging to the same class) based on querying constraints between selected candidate points. The interested reader is referred to [4] for further information. Performances of PCKmeans and our proposed algorithm PCS2 are shown in Figure 7. The performance of the standard Kmeans algorithm serves as a baseline, which is represented as a horizontal line since it is independent of constraints. From Figure 7, we can observe that: (1) Different from the case when random constraints are used, this time the improvement of the performance of PCS2 over that of PC becomes significant for most of the balanced data sets. However, for the Half-rings data set, the improvement is less significant than that in the case of random constraints, as shown in Figure 6. (2) PCS2 results in the best performance among all the data sets. In particular, while PCKmeans is less satisfactory than Kmeans in the cases of the Petals and Wine data sets, PCS2 is consistently better than Kmeans. This suggests that PCS2 can perform clustering effectively under different conditions, i.e., with different constraints generated by different approaches.

IV. Conclusion and Future work

In this paper, we propose PCS, a new semi-supervised algorithm which incorporates pairwise constraints and size constraints into clustering. We have experimentally compare our new algorithm with different clustering algorithms, including (i) Kmeans which is a standard clustering algorithm; (ii) KmeansS in which the size constraints are used and (iii) PCKmeans which is a partial constrained clustering algorithm with pairwise constraints. Experimental results on several benchmark data sets demonstrate that by incorporating prior information in the form of both kinds of constraints, the new algorithm generally outperforms all the previous algorithms.

References

[1] K.Wagstaff, C. Cardie, S. Rogers, and S. Schroed, "Constrained k-means clustering with background knowledge," in *Proc. of 18th Intl Conf. on Machine Learning*, 2001.

[2] S. Zhang and H.-S. Wong, "Partial closure-based constrained clustering with order ranking," in *Proceedings of the 2008 IEEE International Conference on Pattern Recognition*, Tampa, Florida, USA, 2008.

[3] A. Banerjee and J. Ghosh, "Scalable clustering algorithms with balancing constraints." *Data Mining and Knowledge Discovery*, vol. 13, no. 3, pp. 365–395, 2006.

[4] S. Basu, M. Bilenko, and R. J. Mooney, "Active semi-supervision for pairwise constrained clustering," in *Conf of 4th SIAM Data Mining*, 2004.

[5] K. Krishna and M. Narasimha Murty, "Genetic k-means algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 29, no. 3, pp. 433–439, Jun 1999.

[6] G. Babu, N. Murty, and S. Keerthi, "A stochastic connectionist approach for global optimization with application to pattern clustering," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 30, no. 1, pp. 10–24, Feb 2000.

[7] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[8] A. Strehl and J. Ghosh, "Relationship-based clustering and visualization for high-dimensional data mining." *INFORMS Journal on Computing*, vol. 15, no. 2, pp. 208–230, 2003.

[9] S. Zhong and J. Ghosh, "Scalable, balanced model-based clustering." in *Proceedings of the 3rd SIAM Conference on Data Mining*, D. Barbar and C. Kamath, Eds. SIAM, April 2003, pp. 71–82.

[10] A. Banerjee and J. Ghosh, "Frequency-sensitive competitive learning for scalable balanced clustering on high-dimensional hyperspheres," *IEEE Transactions on Neural Networks*, vol. 15, no. 3, pp. 702–719, May 2004.

[11] H. Xiong, J. Wu, and J. Chen, "K-means clustering versus validation measures: A data-distribution perspective," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 2, pp. 318–331, April 2009.

[12] F. Höppner and F. Klawonn, "Clustering with size constraints." in *Computational Intelligence Paradigms*, ser. Studies in Computational Intelligence, L. C. Jain, M. Sato-Ilic, M. Virvou, G. A. Tsihrintzis, V. E. Balas, and C. Abeynayake, Eds. Springer, 2008, vol. 137, pp. 167–180.

[13] K. Wagstaff, S. Basu, and I. Davidson, "When is constrained clustering beneficial, and why?" in *Twenty-First National Conference on Artificial Intelligence*, 2006.

[14] I. Davidson, K. Wagstaff, and S. Basu, "Measuring constraint-set utility for partitional clustering algorithms," in *The 17th European Conference on Machine Learning*, 2006.

[15] I. Davidson and S. S. Ravi, "Identifying and generating easy sets of constraints for clustering," in *Twenty-First National Conference on Artificial Intelligence*, 2006.

[16] Y. Hong and S. Kwong, "Learning assignment order of instances for the constrained k-means clustering algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 2, pp. 568–574, April 2009.

[17] N. Grira., M. Crucianu, and N. Boujemaa, "Active semi-supervised fuzzy clustering," *Pattern Recognition*, vol. 5, no. 41, pp. 1834–1844, 2008.

[18] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 1967, pp. 281–297.

[19] T. Cover and J. Thomas, *Elements of Information Theory*. Wiley, New York, NY, 1991.

[20] S. Kullback and R. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 1, no. 22, pp. 79–86, 1951.

[21] M.Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Proc. of 21st ICML*, 2004.

[22] A. L. N. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 835–850, 2005.

[23] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information." in *Advances in Neural Information Processing Systems 15*, 2003, pp. 505–512.

[24] Y. Liu, R. Jin, and A. K. Jain, "Boostcluster: boosting clustering by pairwise constraints." in *Proc. ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*. ACM, 2007, pp. 450–459.

[25] L. I. Kuncheva and D. Vetrov, "Evaluation of stability of k-means cluster ensembles with respect to random initialization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1798–1808, 2006.

[26] A. Strehl and J. Ghosh, "Cluster ensembles — a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2002.