

Partial Closure-based Constrained Clustering with Order Ranking

Shaohong Zhang, Hau-San Wong

Department of Computer Science, City University of Hong Kong
shazhang@student.cityu.edu.hk, cshswong@cityu.edu.hk

Abstract

In this paper we propose a new partial closure-based constrained clustering algorithm. We introduce closures into the partial constrained clustering and we propose a new measurement to order the importance of the constrained closures. Experiments on public datasets demonstrate the advantages of our algorithm over the standard Kmeans and two state-of-the-art constrained clustering algorithms.

1. Introduction

Recently, semi-supervised clustering methods [1, 5, 6] have become very popular because they can also take advantage of additional supervisory information in the form of pairwise constraints, i.e., pairs of instances labeled as belonging to same or different clusters. These two kinds of constraints are usually termed as : (i) Must-link (ML) and (ii) Cannot-link (CL) constraints. Constrained-based methods rely on these constraints to guide the clustering algorithms towards a more appropriate partitioning. These methods can be categorized to two classes listed below. (i) Algorithms that try to satisfy each constraints beyond considering distances to cluster centroids, such as COPKmeans [5];(ii) algorithms that optimize some cost functions which take into consideration not only distances but also constraints, such as Pairwise Constrained K-means(PCKmeans) [1].

Recent research works believe that ML constraints are more useful and easier to satisfy for a feasible solution than CL constraints[4]. One popular approach to take advantage of ML constraints is first to merge the points within ML constraints to a set of closures such that points in each particular closure belong to the same cluster. Moreover, most of the constrained clustering algorithms are sensitive to the ordering of the points to be clustered[1][3]. Davison et al. investigate relative problems and propose to identify and generate easy

constraints sets for COPKmeans [3]. In this paper, we extend the above ideas to the partial constrained clustering algorithm category(i.e., PCKmeans) to develop a new algorithm, PCKmeans(Partial Closure-based Constrained Kmeans).

2. Constrained clustering

Given a dataset $\{x_i\}_{i=1}^N$, the traditional unsupervised partitioning clustering algorithms (e.g. Kmeans[2]) aim to find a disjoint k partition $\{X_h\}_{h=1}^k$ (each with a centroid μ_h respectively) of X such that the total distance between points and their centroids is minimized. For constrained clustering algorithms(e.g., COPKmeans and PCKmeans), they take into consideration not only distances but also constraints in different fashions. PCKmeans is more general and scalable than COPKmeans since it is to optimize a weighed cost function while COPKmeans is a hard-constrained solution[1]. Therefore, we would like to investigate PCKmeans rather than COPKmeans in this paper.

PCKmeans initializes the cluster centroids with the well separated closures at first and then alternates between the cluster assignment and centroid estimation steps until converged. Each point is assigned to the cluster such that it minimizes a cost function. Formally, given a dataset X , a set of ML constraints M , a set of CL constraints C , the corresponding incurred penalty weights w_{ij} (\vec{w}_{ij}) for violating ML(CL) constraints and the number k of clusters, PCKmeans aims to find a disjoint k partitioning $\{X_h\}_{h=1}^k$ (each with its centroid μ_h) so as to minimize the following cost function [1]

$$J_{pckm} = \frac{1}{2} \sum_{h=1}^k \sum_{x_i \in X_h} \|x_i - \mu_h\|^2 + \sum_{(x_i, x_j) \in M} w_{ij} \delta(l_i \neq l_j) + \sum_{(x_i, x_j) \in C} \vec{w}_{ij} \delta(l_i = l_j) \quad (1)$$

with the indicator function $\delta(true) = 1, \delta(false) = 0$.

In general PCKmeans can perform well given enough constraints and appropriate weights. However these

weight are not easy to set properly. The author set w_{ij} and w_{ij}^* to be similar values for different datasets respectively (e. g. 0.001 for text-datasets and 1 for Iris datasets in [1]) but these weights might not always suitable for other datasets. In fact different constraints should have different contributions and the difference is not easy to be identified. Moreover, the cluster assignment step of PCKmeans is order-dependent and the authors consider a random ordering of the points in the assignment step. Also this random ordering might be improved by some motivation-driven solutions. In view of these motivations, we extend PCKmeans to a closure-based algorithm, PCKmeans, which also aims to attach different importance to different kinds of constraints and to remove the difficulty to set the appropriate weights.

3. PCKmeans

There are mainly two stages in the PCKmeans algorithm. In the first stage, PCKmeans generate closures based on ML constraints and initialize the cluster centroids; In the clustering stage, we adopt the classical EM schemes, including the assign-cluster step and the estimate-means step, which is repeated until converged:(i) assign-cluster: assign closures to clusters according to some cost function. (ii) estimate-means: calculate the centroids for each cluster with closures assigned to them.

3.1 Closure

In the closure stage, points which are directly or transitively in ML constraints are merged together to form closures. We also assign the separated points, which can not be merged, to odd closures. Therefore, the clustering task is changed to cluster these closures $\{C_i\}_{i=1}^{N_c}$. Note that there are no known ML constraints between any two closures otherwise those two closures will further be merged to larger ones. The original CL constraint set C can be updated to form a new CL constraint set C_C for the closure set as follows:

$$(x, y) \in C \& x \in c_i \& y \in c_j \Rightarrow (c_i, c_j) \in C_C \quad (2)$$

3.2 Assign-cluster

In this step, PCKmeans first construct a ranked order for all the closures and then assign closures to clusters in this order to minimize our new cost function.

(i)Order-Ranking. Most of the constrained clustering algorithms are order-dependent, more specifically, on the order of the constrained points in the assign-cluster

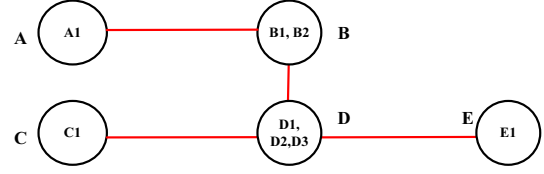


Figure 1. A CL graph example.

Table 1. Removal List for the Example

| Step | A | B | C | D | E | Removal |
|------|-------|-------|-------|-------|-------|---------|
| 0 | 1.406 | 3.750 | 1.406 | 7.219 | 1.406 | A |
| 1 | 0 | 2.750 | 1.406 | 7.219 | 1.406 | C |
| 2 | 0 | 2.750 | 0 | 6.219 | 1.406 | E |
| 3 | 0 | 2.750 | 0 | 4.219 | 0 | B |
| 4 | 0 | 0 | 0 | 3.219 | 0 | D |

step. A common method to reduce this effect is to generate a random ordering of the points for each iteration. Davison et al. investigate relative problems and propose to identify and generate easy constraints sets for COPKmeans [3]. In their approach, an undirected graph $G(V, E)$ for CL constraints is constructed where V is the node set for points(closures) and each edge in E is a CL constraint linking two nodes in V . They recursively remove the nodes in ascending order of the degrees of these nodes to insert, in reverse order, to build a q-inductive order list Q . When a node is removed, the degrees of the nodes linked to the removed one will decrease by 1. For example, one of the q-inductive order lists for the CL graph in Fig. 1 is $\{E, D, C, B, A\}$. The later a node is inserted to Q , the top of its order is. The constraints in this kind of q-inductive order are reported to have more chances to be a easy set than the same constraint set in random order[3].

However, this approach can not directly be used to identify the importance of the nodes. We find the graph degree measurement cannot always work well. Nodes with similar degrees could be in different graph structures. For example, in Fig. 1, after the removal of the node A , node B and C have a similar degree(i.e., $d_B = d_C = 1$), while they are quite different in the original graph. Moreover, the approach does not take the size of the closures into consideration. Usually a larger closure should be more important than an odd closure and larger closures have fewer chances to be outliers. In view of these cases, we extend the approach to rank the constrained closures with an extended degree measurement, rather than the original graph degree measurement for nodes. Formally, we add two weighed components:

$$ed(c_i) = (1 + \frac{w_1 * d_i}{D}) * (1 + \frac{w_2 * N_i}{N}) * d_i \quad (3)$$

here d_i is the degree for node c_i , D is the edge number for the whole CL graph ($D = 0.5 * \sum_{i=1}^{N_c} d_i$), N_i is the size of the closure c_i and N is the point number for the whole dataset ($N = \sum_{i=1}^{N_c} N_i$). For c_i , two weights w_1 and w_2 are used to measure the impact of the closure size and the initial degree respectively and both are set to 1 in this paper. Nodes with the minimum extended degree will be removed to a list in reverse order, and the remained nodes linked to the removed node decrease their extended degree by 1. This removal repeats until the CL graph is empty. The (reverse) removal list of the CL graph in Fig. 1 is $\{D, B, E, C, A\}$, shown in Table 1: in step 0, extended degrees for each node are calculated with (3) and in the other steps the removed nodes will have zero extended degrees while the remained nodes will update their extended degrees corresponding to the removed nodes, i.e., decrease by 1 if one nodes linked to it is removed. It is obvious that our (reverse) removal list $\{D, B, E, C, A\}$ is better than the former one $\{E, D, C, B, A\}$, since D and B are more constrained than A, C, E in the CL graph.

A threshold is adopted to divide the closures to two parts. Closures will be selected into a top-priority set to assigned labels in high priority if their initial extended degrees are beyond the threshold. The other constrained closures in the low-priority set are ordered randomly behind the top-priority set.

(ii) Assign-Cluster. It is not reasonable that different constraints have the same importance. However, to identify all the importance weigh for each constraint is unpractical and intractable. In general, to consider the nodes (closures) in the CL graph is more effective than the edges (CL constraints). It is easier to tell different constrained conditions for different nodes in the first case. Another interesting motivation is to consider the cluster number for the violated closures. For example, if c_i is assigned to X_j and this assignment violates some CL constraints between c_i with c_1, c_2, c_3 , then it is meaningful to take in consideration how many clusters totally that c_1, c_2, c_3 belong to. The more the violated cluster number is, the worse the assignment would be. With the above motivations, PCKmeans adopt a new cost function. Formally, for a closure set $\{c_i\}_{i=1}^{N_c}$ with the closure size $\{N_i\}_{i=1}^{N_c}$ and a CL constraint set C_c , PCKmeans aims to find a k disjoint partition $\{X_i\}_{i=1}^k$ (with their centroids $\{\mu_i\}_{i=1}^k$) so as to minimize the following cost function .

$$J_{pckm} = \frac{1}{2} \sum_{h=1}^k \sum_{c_i \in X_h} (N_i * \|m_{c_i} - \mu_h\|^2) + \frac{1}{2} \sum_{c_i \in X_h} \sum_{(c_i, c_j) \in C_c} (v_{ih} * N_j * \delta(l_i = l_j)) \quad (4)$$

where v_{ih} is the total number of the violated clusters that c_i is assigned to X_h , m_{c_i} is the mean(centroid) of closure c_i and l_i is current cluster label for c_i . Comparing to the cost function 1 for PCKmeans, this cost function replaces the uncertain weights $w_{ij}(w_{ij}^*)$ with the closure sizes (e.g. N_j) and the violated cluster numbers(e.g. v_{ih}), which can be automatically calculated during loop iterations.

3.3 Estimate-Means

After the closures $\{c_i\}_{i=1}^{N_c}$ are assigned to the clusters $\{X_h\}_{h=1}^k$, they are used to calculate the centroids $\{\mu_h\}_{h=1}^k$ with their sizes $\{N_i\}_{i=1}^{N_c}$ as weights:

$$\mu_h = \frac{\sum_{c_i \in X_h} (N_i * c_i)}{\sum_{c_i \in X_h} N_i} \quad (5)$$

3.4 Summary

The whole algorithm is summarized in Fig. 2. In the final step, the cluster label L_c for each closure is assigned to the points in it such that the cluster label L for all the original points is acquired.

Algorithm 1: PCKmeans

INPUT: dataset X , cluster number k , ML/CL constraint set M/C , maximum iteration m ;

OUTPUT: clustered result L of X

METHOD:

1. generate a closure set G using ML constraints M ;
 2. generate CL constraint set C_c for G from C ;
 3. initialize clusters centroids $\{\mu_i\}_{i=1}^k$;
 4. cluster G with the EM scheme;
 5. **Repeat**
 6. reorder closures in G according to (3);
 7. assign cluster label L_c to the closure set G according to (4);
 8. update clusters $\{\mu_i\}_{i=1}^k$ according to (5);
 9. **Until** L_c converges or maximum iteration m reaches
 10. assign cluster label L to X according to L_c .
 11. return L .
-

Figure 2. The PCKmeans algorithm

4 Experiments

We implement experiments on six public UCI¹ datasets (IRIS, Wine, Glass, Protein, Balance-Scale and pima) with the min-max normalization in each dimension. For example, denote by min_A/max_A the minimum/maximum value for dimension A respectively, the normalized value for v is

$$v' = \frac{v - min_A}{max_A - min_A} \quad (6)$$

¹<http://archive.ics.uci.edu/ml/>

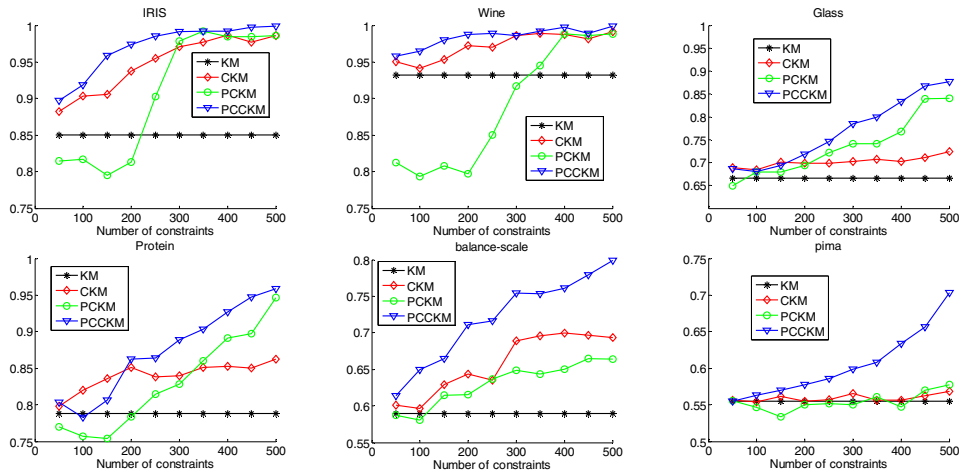


Figure 3. Clustering results.

For each dataset, Point pairs are selected randomly to form the constraint set. Rand index is adopted to evaluate agreements between clustering results and the correct labels. For an dataset X with N points, let l_i be the clustered label assigned to point x_i , t_i be the true label of x_i and $\delta(\cdot)$ be the indicator function. Rand index is defined as [5]

$$R = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \frac{\delta(\delta(l_i = l_j) = \delta(t_i = t_j))}{0.5N(N-1)} \quad (7)$$

We compare PCKmeans (PCKM) with the classical unconstrained Kmeans (KM) and two constrained clustering method, COPKmeans (CKM) and PCKmeans (PKM). The maximum iteration number for all the algorithms is set to 800. For PCKmeans (PKM), the violated weights w_{ij} and w_{ij}^* are set to 1 as [1]. For fair comparisons, all the algorithms are initialized with the same cluster centroids. Rand Index results averaged over 20 independent runs are shown in Fig. 3. We can observe that PCKmeans outperforms the other algorithms significantly. With the number of constraints increasing, Rand Index results for all the three algorithms increase correspondingly and PCKmeans is the best of all. Another interesting observation is that PCKmeans does not always benefit from constraints. For example, for datasets Iris, Wine and Protein, PCKmeans with small numbers of constraints(e.g, 150 constraints) even has a worse Rand result than Kmeans. However, PCKmeans and COPKmeans are always better than Kmeans.

5 Conclusions

In this paper we propose a new partial closure-based constrained clustering algorithm PCKmeans, based on

PCKmeans. We introduce closures into the partial constrained clustering and we propose a new measurement to order the importance of the constrained closures. We advance a novel cost function to remove the uncertain weigh dependence in PCKmeans. Experiments on several public datasets demonstrate the advantages of our algorithm over the standard Kmeans and two state-of-the-art constrained clustering algorithms.

6 Acknowledgments

The work described in this paper was partially supported by a grant from the Research Grants Council of Hong Kong Special Administrative Region, China [Project No. CityU 121005], and a grant from the City University of Hong Kong [Project No. 7001965 and 7002141].

References

- [1] S. Basu, M. Bilenko, and R. J. Mooney. Active semi-supervision for pairwise constrained clustering. In *Conf of 4th SIAM Data Mining*, 2004.
- [2] P. S. Bradley and U. M. Fayyad. Refining initial points for k-means clustering. In *Proc. of 15th ICML*, 1998.
- [3] I. Davidson and S. S. Ravi. Identifying and generating easy sets of constraints for clustering. In *Twenty-First National Conference on Artificial Intelligence*, 2006.
- [4] I. Davidson and S. S. Ravi. Intractability and clustering with constraints. In *Proc. of 24th ICML*, 2007.
- [5] 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.
- [6] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proc. of 21st ICML*, 2004.