Semi-supervised Clustering by Selecting Informative Constraints

Vidyadhar Rao and C.V. Jawahar

IIIT-Hyderabad, India
{vidyadhar.rao@research.,jawahar@}iiit.ac.in

Abstract. Traditional clustering algorithms use a predefined metric and no supervision in identifying the partition. Existing semi-supervised clustering approaches either learn a metric from randomly chosen constraints or actively select informative constraints using a generic distance measure like Euclidean norm. We tackle the problem of identifying constraints that are *informative* to learn appropriate metric for semi-supervised clustering. We propose an approach to simultaneously find out appropriate constraints and learn a metric to boost the clustering performance. We evaluate clustering quality of our approach using the learned metric on the MNIST handwritten digits, Caltech-256 and MSRC2 object image datasets. Our results on these datasets have significant improvements over the baseline methods like MPCK-MEANS.

Keywords: Semi-supervised Clustering, Constraint Selection, Metric Learning.

1 Introduction

Clustering aims to partition a collection of data items/objects into clusters, such that objects within a cluster are more "similar" to each other than they are to objects in other clusters. A common approach is to infer the similarity based on a distance measure on the data objects. A good distance metric between objects accurately reflects the true underlying relationships i.e., reports small distances for similar objects and large distances for unrelated objects. In literature, the notion of similarity has many variants and formulations [9]; some according to the purpose of the study [8], some making domain-specific assumptions [8] while some relying on prior knowledge of the problem [1] [3] [15].

Semi-supervised clustering algorithms offer a way to specify the prior information in the form of instance level pairwise constraints like must-link constraints and cannot-link constraints. Many works have demonstrated that the use of such constraints could help in guiding the clustering algorithm towards a more appropriate data partitioning. Some of these approaches incorporate the constraints by modifying the clustering objective function so that it includes enforcing constraints [15], or initializing and constraining clustering based on labeled examples [1]. However, it has been shown that constrained clustering is a hard problem [4] and it is not necessarily a good idea to derive the partitions strictly satisfying every constraint [16]. Instead of enforcing the constraints directly, recent techniques introduced penalties on constraint violations.

P. Maji et al. (Eds.): PReMI 2013, LNCS 8251, pp. 213–221, 2013.

Other approaches employ distance metric into the clustering algorithm; where the metric is first trained to satisfy the constraints. The unified approach, MPCK-MEANS [3] uses pairwise constraints along with unlabeled data for constraining the clustering and learning distance metrics. This technique penalizes the constraint violations on must-link and cannot link constraints. However, the constraint specification effort can become a significant burden as they do not offer any mechanism for selecting informative pairwise constraints. Moreover, the cost associated with constraint violations do not include any prior knowledge from the data.

One of the solution to this problem is to adapt an active learning strategy where the algorithm queries an oracle that can assign a must-link or cannot-link label to a given constraint pair. The goal is to query potentially interesting constraint pairs and obtain a better partition of data with minimal number of queries. However, the existing approaches assume that the target metric is available for clustering data. Active PCKMeans [2], depends on farthest-first strategy, can pose a constant number of queries to the oracle by actively selecting informative constraints to get improved clustering performance. Active query selection algorithm [11] is a special case of min-max approach, using a Gaussian kernel to measure the uncertainty in deciding the cluster memberships.

To this end, the previous semi-supervised clustering approaches either learn a metric from randomly chosen constraints or actively select informative constraints using generic distance measures like Euclidean norm or a Gaussian kernel. In this work, we focus on how to utilize the user specified constraints effectively to infer the cluster labels by learning appropriate distance metrics. Unlike MPCK-means, we impose pairwise distance constraints to simultaneously learn metric and perform clustering to guide the process of active constraint set acquisition. We empirically evaluate the effectiveness of our approach on Caltech-256, MSRC2 and MNIST image datasets. The proposed approach performs better than previously proposed methods that either use a pre-defined metric or use random set of constraints for learning a metric.

2 Semi-supervised Clustering

Our goal of learning distance metrics is to improve the clustering performance by selecting informative pairwise constraints. Metric learning often learns the appropriate distance function (metric) from a set of examples in a supervised setting. In a semi-supervised clustering setting, one could learn appropriate metric, if enough pairwise constraints (must-link and cannot-link) are provided. However, getting appropriate constraints from an oracle is often hard. One needs to actively minimize the extra supervision for solving the clustering problem.

Thus, we need to solve these two sub-problems in a coupled manner. Specifically, when the scheme has an access to a set of constraining pairs, our goal is to find potential supervision information and learn appropriate distance metric for semi-supervised clustering.

2.1 K-Means with Mahalanobis Metric

Given N points $X = \{x_i\}_{i=1}^{N}$, $x_i \in R^d$, we consider the K-means problem as a disjoint K-partition of $X = \{X_h\}_{h=1}^{K}$. We wish to find the K cluster centroids $\{\mu_h\}_{h=1}^{K}$ and $l = \{l_i\}_{i=1}^{N}$, the cluster labels , where $l_i \in \{1, \ldots, K\}$. Points in the same partition are assigned with the same cluster label. The goal is to minimize the objective function

$$\phi_A(X) = \sum_{x_i \in X} d_A(x_i, \mu_{l_i}) \tag{1}$$

where the squared Mahalanobis distance between the points x and y is defined as $d_A(x, y) = (x - y)^T A(x - y)$ and A is a $d \times d$ positive semi-definite matrix. Note that, when $A = I$, the above equation is equivalent to the K-means clustering problem with Euclidean distance metric. Our objective is to learn the distance metric, A using the constraints given by the user and later adopt it in Eqn.(1) so that $\phi_A(X) < \phi_I(X)$.

2.2 Learning with ITML

Information theoretic metric learning (ITML) method of [6] learns a Mahalanobis distance metric, A, under a given set of constraints. Given an initial $d \times d$ positive definite matrix, A_0, specifying the prior knowledge about the inter-point distances, ITML [6] minimizes the LogDet divergence between matrices A and A_0 subject to a set of constraints specified by the user. The goal is to enforce simple distance constraints for similar(S) and dissimilar(D) points and solve the following optimization problem:

$$\begin{aligned} \min \;& D_{ld}(A, A_0) \\ \text{s.t.} \;& A \succeq 0 \\ & d_A(x_i, x_j) \leq u \qquad (i, j) \in S \\ & d_A(x_i, x_j) \geq v \qquad (i, j) \in D \end{aligned} \tag{2}$$

where $D_{ld}(A, A_0) = tr(AA_0^{-1}) - \log det(AA_0^{-1}) - d$; v and u are large and small values, respectively. Solving Eq.(2) involves repeatedly projecting the current solution onto a single constraint, via an update:

$$A_{t+1} = A_t + \beta_t A_t (x_{i_t} - x_{j_t})(x_{i_t} - x_{j_t})^T A_t, \tag{3}$$

where x_{i_t} and x_{j_t} are the constrained data points for iteration t, and β_t is a projection parameter computed by the ITML algorithm. Our goal is to identify informative constraints from the data that minimizes the Eq.(2) to learn the appropriate distance metrics for semi-supervised clustering problem.

2.3 Semi Supervised Clustering by Selecting Informative Constraints

We perform K-means clustering and metric learning using ITML [6] such that both the objectives in Eq.(1) and Eq.(2) are solved simultaneously. We also

provide a quantitative way to measure the constraint-set utility for paritional clustering as adapted in [5].

Informativeness refers to the amount of information in the constraint set that the algorithm cannot determine on its own. *Coherence*, measures the amount of agreement with in the constraints themselves, with respect to a given distance metric. Constraint sets with high informativeness and coherence tend to result in increase in clustering performance [5]. In our approach, we use pairwise distance constraints which are more *informative* than simple must-link and cannot-link constraints. We perform ITML to learn the metric and subsequently use it for the K-means clustering. The main steps for selecting constraints with high *coherence* is summarized in Algorithm 1.

Algorithm 1: Semi Supervised Clustering

Data: X, K, S, D, u, v
Result: A, S_a
$A_0 = I$ *// Initial prior about the inter-point distances*;
$S_u = S \cup D$ *// User specified similar and dissimilar constraints*;
$S_a = \{\}$ *// Active(informative) constraints selected by our algorithm*;
repeat
  foreach $(i, j) \in S_u$ **do**
    //learn metric (A^{ij}) *if a new constraint* $(i, j) \in S_u$ *is added to* S_a
    $A^{ij} \leftarrow \text{ITML}(X, A_0, S_a \cup (i, j), u, v)$;
    // Obtain cluster assignments (l^{ij}) *and quality of clustering* (Q^{ij})
    $(Q^{ij}, l^{ij}) \leftarrow K\text{-means}(X, A^{ij}, K)$; *//use metric* A^{ij} *for* K*-means*
  end
  $(i^*, j^*) \leftarrow \arg\max_{ij}(Q^{ij})$;
  $A_0 \leftarrow A^{i^* j^*}$;
  $S_a \leftarrow S_a \cup (i^*, j^*)$;
  $S_u \leftarrow S_u \setminus (i^*, j^*)$;
until *convergence*;

In each iteration, a new constraint from a set of user specified constraints S_u is added into the active constraint set, S_a. ITML learns the distance metric by using constraints from the active constraint set, S_a. Thereafter, we perform the K-means clustering again using the learned distance metric, A^{ij}, to obtain the cluster labels, l^{ij} and the clustering quality, Q^{ij}. Then update the active constraint set, S_a, with a constraint which resulted in maximum performance in the clustering. We also update S_u accordingly and repeat this process until a satisfactory clustering performance is achieved.

In our experiments, we have selected 50 informative constraints to show the effectiveness of Algorithm 1. In addition to the active constraints generated from Algorithm 1, we can also infer additional constraints using the transitive closure of the set of constraints [2]. Given three data points x, y, z, if $(x, y) \in S$ and $(y, z) \in S$ then $(x, z) \in S$; if $(x, y) \in S$ and $(y, z) \in D$ then $(x, z) \in D$.

Our algorithm requires $O(mn)$ metric learning and clustering operations, where $m = |S_a|$ is the number of informative constraints selected and $n = |S_u|$ is the number of user specified constraints. The complexity of our algorithm scales linearly with the number of user specified constraints. Empirical results suggest only a few informative constraints have significant effect on clustering performance.

3 Experiments and Results

3.1 Experimental Setup

Datasets: To assess the viability of proposed approach, we have performed experiments using MNIST [10] database of handwritten digits, Caltech-256 [7] and MSRC2 [13] object datasets in our experiments. We have selected 11 objects from Caltech-256 dataset and all 20 objects from the MSRC2 dataset for object clustering. For MNIST dataset, we have used all the ten digits for clustering. The parameter K in Algorithm 1, which is the number of clusters in each dataset, is set as the real number of classes in each dataset. Fig.1 shows sample images.

For ITML algorithm, distances are constrained to be either similar or dissimilar, based on the class values, and are drawn only from training set. For MNIST dataset, we choose constraints only from 10 training examples per digit and evaluate the clustering on the 100 testing examples per digit. For the Caltech-256 dataset, we have used 5 images for training and 45 images for testing from each of the 11 classes. For MSRC2 dataset, we have used 5 images for training and 25 images for testing from each of the 20 classes.

Fig. 1. Sample object and handwritten digit images used in our experiments: Caltech-256 (left), MSRC2 (middle), and MNIST (right)

Features: Clustering is performed based on visual features extracted automatically from the images. For the Caltech-256 and MSRC2 object clustering, the image data is supplied in the form of visual words with a vocabulary size of 600 using the popular SIFT descriptors as in VLFeat library [14]. In the MNIST dataset, the digits have been normalized to fit in a 20x20 pixel box and the resulting 400 pixel values are used as feature representation for each image. In our implementation, we empirically set the slack variables v and u in Eqn.(2) to the 95^{th} and 5^{th} percentiles of the distribution of pairwise Euclidean distances within the dataset, respectively.

Evaluation Methods: A number of ways have been developed to validate unsupervised clustering algorithms [17]. In our case, the ground truths for the datasets are naturally available, that is, the digit labels and object categories. We also report the clustering performance by comparing total squared Mahalanobis distances computed from Eq.(1). We also evaluate the quality of clusters using the F_1-measure [2] and Rand Index [12] commonly used performance metric for semi-supervised clustering algorithms.

3.2 Results and Discussion

We compare our approach to the baseline methods like popular K-means which is an unsupervised clustering with Euclidean distance metric and MPCK-MEANS which is a semi-supervised clustering algorithm that simultaneously learns metric under some constraints. We quantify the comparison by using the same initial centroids for K-means in all the methods. For semi-supervised algorithms, we show the results by selecting 50 informative constraints using Algorithm 1.

Three variants of Algorithm 1 are implemented: Semi-supervised clustering with random constraints (SSC-rand), Semi-supervised clustering with online distance metric learning (SSC-OLDML) and Semi-supervised clustering with active constraint set generation (SSC-active). In SSC-rand, we perform K-means clustering with a distance metric learned directly from random constraints. In SSC-OLDML, we perform K-means clustering with a distance metric learned using the most recently obtained metric as prior (using Eq.(3) for ITML). In SSC-active, we perform K-means clustering with a distance metric learned using the active constraint set acquired in each iteration (using Eq.(2) for ITML).

We present experimental results of our approach on three image datasets and compare them with the baseline methods. Table 1 shows the K-means error for all methods. The results show that the SSC-active performs better than popular K-means (does not learn a metric), SSC-rand (learns metric from random constraints) and SSC-OLDML methods. We have not included the MPCK-means algorithm in Table 1, as it includes penalties for constraint violations in the objective function.

Table 1. Unsupervised K-means clustering error, $\phi_I(.)$, along with Semi-Supervised clustering error, $\phi_A(.)$, with 50 informative constraints

	Algorithm			
Dataset	Popular K-means	SSC-rand	SSC-OLDML	SSC-active
MNIST	37380	36562	61474	**34726**
Caltech-256	2.665	2.565	2.618	**2.020**
MSRC2	2.059	2.275	3.344	**1.991**

In Table 2, we have used rand index [12] to evaluate the semi-supervised clustering algorithms. We notice that MPCK-means does not boost the performance when compared to popular K-means on MNIST and MSRC2 datasets. This shows that incorporating random constraints might degrade the clustering

performance and it is very critical to choose constraints that are informative for semi-supervised clustering.

SSC-active adopts distance metric learning by choosing informative constraints, and always performs better than the unsupervised K-means on all three datasets. The underlying reasoning seems to be that the pairwise distance constraints are more *informative* and also that the constraints selected by our approach have high *coherence* with respect to the learned metric.

In contrast to what we observed for MNIST and MSRC2 datasets, SSC-active did not quite perform well on the Caltech-256 images in comparison to the MPCK-means. The images in Caltech dataset typically share multiple object categories and we believe that this can be surmounted by use of large vocabularies or using more features for image representation.

Table 2. Results of the semi-supervised clustering based methods, measured in Rand Index (higher is better)

	Algorithm				
Dataset	Popular K-means	SSC-rand	SSC-OLDML	MPCK-means	SSC-active
MNIST	0.875	0.881	0.861	0.862	**0.921**
Caltech-256	0.769	0.758	0.827	**0.841**	0.807
MSRC2	0.892	0.895	0.881	0.859	**0.904**

In Table 3, we compare the performance of clustering methods in terms of F_1-measure [2]. The results demonstrate that our approach performs close to the MPCK-MEANS on the object datasets and outperforms on handwritten digits images. However, we can see that SSC-active always performs better than the unsupervised clustering on all three datasets.

Table 3. Comparison of the different semi-supervised methods on three datasets, measured in F_1 score (higher is better). For each dataset, we see that SSC-active performs better than unsupervised clustering and performs close to MPCK-MEANS.

	Algorithm				
Dataset	Popular K-means	SSC-rand	SSC-OLDML	MPCK-means	SSC-active
MNIST	0.410	0.434	0.334	0.377	**0.621**
Caltech-256	0.150	0.156	0.195	**0.249**	0.215
MSRC2	0.155	0.162	0.128	**0.226**	0.203

The *SSC-OLDML* approach sometimes degrades the clustering performance (See Tables 1, 2 and 3). This happens because of the greedy fashion in which the metric has been learned; using most recently learned metric as prior can constrain later ones due to potential conflicts between the prior metric and the new constraints, and there is no mechanism for backtracking. From this we can infer that, the constraint set generated by *SSC-OLDML* has low *coherence* and therefore can lead the algorithm into unpromising areas of search space. However, the *SSC-active* continued to be consistent as it learns the metric from all the constraints using the prior as $A_0 = I$ and thereby acquiring the constraints sets

with high *coherence* i.e., amount of agreement within the constraints themselves is high with respect to the learned distance metric.

Overall, our results show that the coupled approach (SSC-active), to utilize informative constraints and learn metric, can boost the performance of semi-supervised clustering. Our approach required only a very small number of informative constraints to gain significant improvements in the clustering over the existing semi-supervised clustering approaches like MPCK-means.

4 Conclusion

Pairwise constraints would facilitate accurate metric learning and boost the quality of semi-supervised clustering algorithms. This paper has presented an approach to jointly learn metric and select informative constraints from a given set of pairwise distance constraints. We partition the entire data using K-means with the learned metric. Our semi-supervised algorithm was applied on the image datasets and its application always achieved better than the baseline methods like unsupervised K-means with Euclidean metric and semi-supervised MPCK-means clustering. Our results demonstrate that the more informative constraints are under the learned metric, the more likely they are to improve clustering.

References

1. Basu, S., Banerjee, A., Mooney, R.: Semi-supervised clustering by seeding. In: Machine Learning-International Workshop then Conference, pp. 19–26 (2002)
2. Basu, S., Banerjee, A., Mooney, R.J.: Active semi-supervision for pairwise constrained clustering. In: SDM, pp. 333–344 (2004)
3. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: ICML, p. 11. ACM (2004)
4. Davidson, I., Ravi, T.: Clustering with constraints: feasibility issues and the fk-means algorithm. In: SDM, pp. 138–149 (2005)
5. Davidson, I., Wagstaff, K.L., Basu, S.: Measuring constraint-set utility for partitional clustering algorithms. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 115–126. Springer, Heidelberg (2006)
6. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: ICML, pp. 209–216. ACM (2007)
7. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset (2007)
8. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recognition Letters 31(8), 651–666 (2010)
9. Jain, A.K., Dubes, R.C.: Algorithms for clustering data. Prentice-Hall, Inc. (1988)
10. LeCun, Y.: Mnist dataset (2000), `http://yann.lecun.com/exdb/mnist/`
11. Mallapragada, P.K., Jin, R., Jain, A.K.: Active query selection for semi-supervised clustering. In: ICPR, pp. 1–4. IEEE (2008)
12. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association 66(336), 846–850 (1971)
13. Shotton, J., Winn, J.M., Rother, C., Criminisi, A.: Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)

14. Vedaldi, A., Fulkerson, B.: Vlfeat: An open and portable library of computer vision algorithms. ACM Multimedia, 1469–1472 (2010)
15. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: Machine Learning-International Workshop then Conference, pp. 577–584 (2001)
16. Wagstaff, K.L., Basu, S., Davidson, I.: When is constrained clustering beneficial, and why? Ionosphere 58(60.1), 62–63 (2006)
17. Xu, R., Wunsch, D.: Clustering, vol. 10. Wiley-IEEE Press (2008)