Sensors & Transducers, Vol. 177, Issue 8, August 2014, pp. 238-245

Sensors & Transducers

Based on Similarity Metric Learning for Semi-Supervised Clustering

Wei QIU
School of Computer Science, Jiaying University, Meizhou, Guangdong, 514015, China
Tel.: 13421003080, fax: 2186960
E-mail: qiuwei@jyu.edu.cn

Received: 21 May 2014 /Accepted: 31 July 2014 /Published: 31 August 2014

Abstract: Semi-supervised clustering employs a small amount of labeled data to aid unsupervised learning. The focus of this paper is on Metric Learning, with particular interest in incorporating side information to make it semi-supervised. This study is primarily motivated by an application: face-image clustering. In the paper introduces metric learning and semi-supervised clustering, Similarity metric learning method that adapt the underlying similarity metric used by the clustering algorithm. This paper provides new methods for the two approaches as well as presents a new semi-supervised clustering algorithm that integrates both of these techniques in a uniform, principled framework. Experimental results demonstrate that the unified approach produces better clusters than both individual approaches as well as previously proposed semi-supervised clustering algorithms. This paper followed by the discussion of experiments on face-image clustering, as well as future work. *Copyright © 2014 IFSA Publishing, S. L.*

Keywords: Semi-supervised clustering, Clustering algorithm, Similarity metric learning.

1. Introduction

Clustering is a task whose goal is to determine a finite set of categories (clusters) to describe a dataset according to similarities among its objects [1]. The applicability of clustering is manifold, ranging from market segmentation and image processing [2] through document categorization and Web mining. An application field that has shown to be particularly promising for clustering techniques is bioinformatics. Indeed, the importance of clustering gene expression data measured with the aid of microarray and other related technologies has grown fast and persistently over the past recent years [3]. Data analysis techniques can be broadly classified into two major types (Tukey, 1977): (i) exploratory or descriptive, meaning that the investigator does not have pre-specified models or hypotheses but wants to understand the general characteristics or structure of the high-dimensional data, and (ii) confirmatory or inferential, meaning that the investigator wants to confirm the validity of a hypothesis/model or a set of assumptions given the available data.

There is a growing interest in a hybrid setting, called semi-supervised learning (Chapelle et al., 2006); in semi-supervised classification, the labels of only a small portion of the training data set are available. In semi-supervised clustering, instead of specifying the class labels, pair-wise constraints are specified, which is a weaker way of encoding the prior knowledge. In spite of the type of algorithm (partitional, hierarchical, or overlapping), the main goal of clustering is maximizing both the homogeneity within each cluster and the heterogeneity among different clusters [2]. In other words, objects that belong to the same cluster should

be more similar to each other than objects that belong to different clusters. The problem of measuring similarity is usually tackled indirectly, i.e., distance measures are used for quantifying the degree of dissimilarity among objects, in such a way that more similar objects have lower dissimilarity values [4]. Several dissimilarity measures can be employed for clustering tasks. Each measure has its bias and comes with its own advantages and drawbacks. Therefore, each one may be more or less suitable to a given analysis or application scenario. Indeed, it is well known that some measures are more suitable for gene clustering in bioinformatics [2], whereas other measures are more appropriate for text clustering and document categorization, for instance.

2. Semi-supervised Clustering

Clustering is deemed one of the most difficult and challenging problems in machine learning, particularly due to its unsupervised nature. The unsupervised nature of the problem implies that its structural characteristics are not known, except if there is some sort of domain knowledge available in advance. Specifically, the spatial distribution of the data in terms of the number, volumes, densities, shapes, and orientations of clusters (if any) are unknown. These adversities may be potentialized even further by an eventual need for dealing with data objects described by attributes of distinct natures (binary, discrete, continuous, and categorical), conditions (complete and partially missing), and scales (ordinal and nominal) [2, 3]. From an optimization perspective, clustering can be formally considered as a particular kind of NP-hard grouping problem. This has stimulated the search for efficient approximation algorithms, including not only the use of ad hoc heuristics for particular classes or instances of problems, but also the use of general-purpose metaheuristics.

Clustering has traditionally been a tool of unsupervised learning. Despite widespread usage across several fields there is not yet a well-established theory to describe clustering [15]. Traditional approaches to clustering optimize some objective function, like the k-means or the kmedian, over the given set of points [16]. These approaches work under the implicit assumption that by minimizing a certain objective function one can reach close to the underlying ground truth clustering. Alternatively, another line of work makes strong assumptions on the nature of the data. One popular in literature is the assumption that data is coming from a mixture of Gaussians [17]. Information explosion is not only creating large amounts of data but also a diverse set of data, both structured and unstructured. Unstructured data is a collection of objects that do not follow a specific format. For example, images, text, audio, video, etc. On the other hand, in structured data, there are semantic relationships within each object that are important. Most clustering approaches ignore the structure in the objects to be clustered and use a feature vector based representation for both structured and unstructured data. A brief summary of some of the recent trends in data clustering is presented below.

Clustering is inherently an ill-posed problem where the goal is to partition the data into some unknown number of clusters based on intrinsic information alone. The data-driven nature of clustering makes it very difficult to design clustering algorithms that will correctly find clusters in the given data. Any external or side information available along with the pattern matrix or the similarity matrix can be extremely useful in finding a good partition of data. Clustering algorithms that utilize such side information are said to be operating in a semi-supervised mode [15]. There are two open questions: (i) how should the side information be specified; and (ii) how is it obtained in practice? One of the most common methods of specifying the side information is in the form of pair-wise constraints.

3. Related Work

Existing methods for semi-supervised clustering fall into two general approaches we call constraint-based and metric-based. In constraint-based approaches, the clustering algorithm itself is modified so that user-provided labels or pairwise constraints are used to guide the algorithm towards a more appropriate data partitioning. This is done by modifying the clustering objective function so that it includes satisfaction of constraints [5], enforcing constraints during the clustering process [6], or initializing and constraining clustering based on labeled examples [7]. In metric-based approaches, an existing clustering algorithm that uses a distance metric is employed; however, the metric is first trained to satisfy the labels or constraints in the supervised data. Several distance measures have been used for metric-based semi-supervised clustering including Euclidean distance trained by a shortest-path algorithm [7], string-edit distance learned using Expectation Maximization (EM) [8], KL divergence adapted using gradient descent [9], and Mahalanobis distances trained using convex optimization [10]. Previous metric-based semi-supervised clustering algorithms exclude unlabeled data from the metric training step, as well as separate metric learning from the clustering process. Also, existing metric-based methods use a single distance metric for all clusters, forcing them to have similar shapes. We propose a new semi-supervised clustering algorithm derived from K-Means, MPCK-MEANS that incorporates both metric learning and the use of pairwise constraints in a principled manner. MPCK-MEANS performs distance-metric training with each clustering iteration, utilizing both unlabeled data and pairwise constraints. The algorithm is able to learn individual metrics for each cluster, which permits clusters of different shapes. MPCKMEANS also

allows violation of constraints if it leads to a more cohesive clustering, whereas earlier constraint-based methods forced satisfaction of all constraints, leaving them vulnerable to noisy supervision. By ablating the metric-based and constraint-based components of our unified method, we present experimental results comparing and combining the two approaches on multiple datasets. The two methods for semi-supervision individually improve clustering accuracy, and our unified approach integrates their strengths. Finally, we demonstrate that the semi-supervised metric learning in our approach outperforms previously proposed methods that learn metrics prior to clustering, and that learning multiple clusterspecific metrics can lead to better results.

4. Semi-supervised Clustering with Metric Learning

While pair wise constraints can guide a clustering algorithm towards a better grouping, they can also be used to adapt the underlying distance metric. Pair wise constraints effectively represent the user's view of similarity in the domain. Since the original data representation may not specify a space where clusters are sufficiently separated, modifying the distance metric warps the space to minimize distances between same-cluster objects, while maximizing distances between different-cluster objects. As a result, clusters discovered using learned metrics adhere more closely to the notion of similarity embodied in the supervision. We parameterize Euclidean distance using a symmetric positive-definite matrix A as following:

$$\left\|x_i - x_j\right\|A = \sqrt{(x_i - \mu l_i)^T A(x_i - \mu l_i)}\,, \quad (1)$$

The same parameterization was previously used by [10] and Bar-Hillel et al. (2003). If A is restricted to a diagonal matrix, it scales each dimension by a different weight and corresponds to feature weighting; otherwise new features are created that are linear combinations of the original ones. In previous work on adaptive metrics for clustering [7-9], metric weights are trained to simultaneously minimize the distance between must-linked instances and maximize the distance between cannot-linked instances. A fundamental limitation of these approaches is that they assume a single metric for all clusters, preventing them from having different shapes. We allow a separate weight matrix for each cluster, denoted Ah for cluster h. This is equivalent to a generalized version of the K-Means model, where cluster h is generated by a Gaussian with covariance matrix A_h^{-1} [4]. It can be shown that maximizing the complete data log-likelihood under this generalized K-Means model is equivalent to minimizing the objective function:

$$J_{mkmeans} = \sum_{x_i \in x} (\left\| x_i - \mu_{l_i} \right\|^2_{A_{l_i}} - \log(\det(A_{l_i}))), \quad (2)$$

where the second term arises due to the normalizing constant of l_i^{th} Gaussian with covariance matrix $A_{l_i}^{-1}$.

4.1. Similarity Metric Learning Method

Now consider a different scenario: we are given points in their original space, being asked to compute a good similarity measure for them with partial supervision. Then we have the freedom to parameterize a global similarity measure for all points, e.g. let $S_D(x_i, x_j) = x_i^T D x_j$ where $x_i, x_j \in R^P$ are unit norm vectors and D is a diagonal positive semi-definite matrix. Note that, with D equal to the identity matrix, $S_D(x_i, x_j)$ measures the correlation between datapoints x_i and x_i. Optimizing over parameter D and labels c_{ij} for some loss function may simultaneously yield a good similarity measure and cluster assignments. This approach is motivated by [5]. We propose maximizing the following objective function:

$$\max_{c_{ij}} \sum S_D(i,j) - \lambda\Big(\sum_{(i,j)\in M} s_{\max} - s_{ij}\Big) I\{\vec{c}_i \neq \vec{c}_j\} - \sum_{(i,j)\in C} (s_{ij}) I\{\vec{c}_i = \vec{c}_j\}\Big)' \quad (3)$$

such that $\left\|diag(D)\right\|_2 = 1$ and $s_{\max}$ is set to be a large positive constant. Problem optimizes for a weighted version of clustering performance and constraint compliance. The first term encourages a good clustering assignment for all the data points, and the second term penalizes the violation of constraints in M and C. Also, constraints on distant must-link pairs and nearby cannot-link pairs are weighed more heavily.

While affinity propagation works toward a better clustering performance (the first term), the following method can be used to update the similarity measure for better constraint compliance:

$$\max I(D) = \sum_{(i,j)\in M} s_D(i,j) I\{\vec{c}_i \neq \vec{c}_j\} - \sum_{(i,j)\in C} s_D(i,j) I\{\vec{c}_i = \vec{c}_j\})' \quad (4)$$

such that $\left\|diag(D)\right\|_2 = 1$.

Let $D = diag(d_1, d_2, ..., d_n)$ where $d_k > 0, \forall k$ and $\sum_k d_k^2 = 1$. The derivative of (4) w.r.t d_k can be computed by

$$\frac{\partial I}{\partial d_k} = \sum_{(i,j)\in M \vec{c}_i \neq \vec{c}_j} (x_i)_k (x_j)_k - \sum_{(i,j)\in M \vec{c}_i = \vec{c}_j} (x_i)_k (x_j)_k\,,$$

where $()_k$ denotes the k^{th} element of a vector.

To increase *I*, we can update *d* (diagonal vector of *D*) in the gradient direction $\frac{\partial L}{\partial d}$ while keeping all d_k positive, then re-normalize *d* back to the unit l_2 norm.

Notice that this method only takes into consideration the label information that the clustering has gotten wrong. We discuss another variant that utilizes all of the label information at each iteration in the Future Work section.

4.2. Boost the Similarity Metric Method

The previous section uses the gradient ascent of the objective function to update the similarity measure. In this section, we use boosting as a supervised learning tool to simultaneously enhance the similarity metric and clustering. As an augment to the previous approach, it is able to learn a positive semi-definite metric instead of a diagonal one. Also, it offers a new perspective on formulating the clustering problem.

Clustering is the assignment of a set of *n* points into *K* subsets (i.e. clusters) so that points in the same cluster are similar in some sense. There are two types of clustering: hard clustering and fuzzy clustering. In hard clustering, each point is assigned to a single cluster. Fuzzy clustering, on the other hand, assigns a probability distribution of different clusters to each point. For both cases, we can use a group cluster matrix $X \in R^{n\times K} s.t. \forall i \sum_j X_{ij} = 1$ to fully describe cluster membership, with an extra condition $X_{ij} \in \{0,1\}$ for hard clustering. Note that fuzzy clustering can be easily transformed to a hard clustering by taking maximum over rows, but not vice versa.

An important observation is that matrix *X* always has rank *K* for a *K* cluster problem. Therefore $G = XX^T$ also has rank *K*, which is usually a lot fewer than the size of matrix $G.G(i,j) = \sum_{k=1}^{K} X_{ik} X_{jk}$ is then the probability of point *i* and *j* being in the same cluster, given the independence among clusters. In semi-supervised clustering, we are given two sets of constraints: must links *M* and cannot links *C*. Therefore $\{(i,j) \in M \bigcup C\}$ serve as labeled points: $G(i,j) = 1$ if $(i,j) \in M$ or 0 if $(i,j) \in C$.

To formulate a boosting problem, the input of the classifier is a pair of points (x_i, x_j) and the output label $Y(i, j) = 1$ if x_i and x_j are classified as in the same cluster and $Y(i; j) = -1$, otherwise. We are given a set of labels in advance: $y_{ij} = 1$ for $(i,j) \in M$ and $y_{ij} = -1$ for $(i,j) \in C$. We define a set of weights on these labeled points: $\{w_{ij} : (i,j) \in M \bigcup C\}$. At the i^{th} iteration, a new distance metric is learned w.r.t these weights. Affinity propagation takes in the new pair wise similarity to produce a basic clustering, which can be easily transformed to binary square matrix Y_t. Then we update weights based on Y_t, i.e. increase weight of points that are classified wrongly, and decrease the weight of points that are correct. The final aggregated classification matrix $Y = \sum_{t=1}^{T} \alpha_t Y_t$, where larger α_t indicates better classification accuracy of Y_t. Finally we transform *Y* to a clustering probability matrix G then decompose $G \approx XX^T$ to obtain group cluster matrix *X*.

Now we iterate each step in greater detail. At the first iteration, the weights are set to be uniform: $w_{ij} = \frac{1}{|M| + |C|}$. Let the similarity metric be $s_D(x_i, x_j) = x_i^T P x_j$ where $x_i, x_j \in R^p$ are unit norm vectors and *P* is a positive semi-definite matrix. The initial *P* is set to be the identity matrix. At each boosting iteration, we solve the following problem:

$$\max_{\Delta} \sum_{(i,j)\in M} w_{ij} x_i^T \Delta x_j - \sum_{(i,j)\in C} w_{ij} x_i^T \Delta x_j = \Delta\Theta\left(\sum_{(i,j)\in M} w_{ij} x_i x_j^T - \sum_{(i,j)\in C} w_{ij} x_i x_j^T\right) = trace(USU^T\Delta), \quad (1)$$

where USU^T is the SVD of symmetric matrix $\left(\sum_{(i,j)\in M} w_{ij} x_i x_j^T - \sum_{(i,j)\in C} w_{ij} x_i x_j^T\right)$.

Therefore $\Delta = U\tilde{S}U^T$ where $\tilde{S}(i,i) = 1$ if $\tilde{S}(i,i) > 0$ and 0 otherwise. The new positive semi-definite matrix is $P = P + \in \Delta$, which is used by affinity propagation in the next step. Affinity propagation takes in the adjusted similarity, and returns clusters of points with an exemplar for each cluster, which uniquely determines Y_t. That is, if x_i and x_j are clustered together, $Y_t(i, j) = 1$. Otherwise $Y_t(i, j) = -1$. If the classification error *et* of labeled points is above 50 %, discard Y_t, further adjust the similarity by $P = P + \in \Delta$ then run affinity propagation again. Otherwise, the coefficient for the t^{th} clustering matrix is $\alpha_t = \frac{1}{2} \ln \frac{1 - e_t}{e_t}$.

The final step of each iteration is updating the weight on each label, $w_{ij} = e^{-\alpha_t y_{ij} Y_t(i,j)}$.

The aggregated clustering matrix $Y = \sum_{t=1}^{T} \alpha_t Y_t$, where Y_t is the final classification for each pair of points to be in the same cluster or

not. The sign of $Y(i, j)$ shows such classification, while the magnitude of it means the confidence of such classification. Boosting theory says that the classification error of labeled points decreases exponentially if *et* is strictly less than half at each iteration. Empirically the test error will decrease along with the training error even after the training error reaches 0.

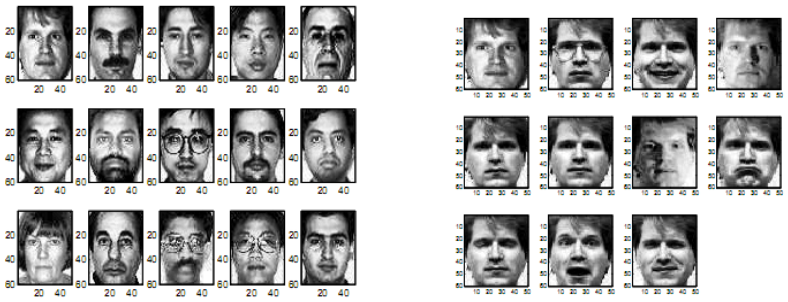
To obtain G then decompose it to obtain group cluster matrix X such that $G \approx XX^T$, two approaches can be used. The first one set G to be the thresholded Y at 0 then computes the SVD of G. Since $Y_{ij} < 0$ means that the majority vote for i and j to be in the same cluster is negative, we threshold them to 0. G is now a non-negative matrix. Computing the SVD of G leads to orthogonal X, yet X is not necessarily non-negative. Finally we take the absolute value of X and assign the index of the maximum of the i^{th} row X_i to the label of the i^{th} point.

The second approach first maps Y into G such that $0 \le G_{ij} \le 1$, then conduct the nonnegative matrix factorization: $G \approx XX^T$ s.t. $X >= 0$ element-wise. An advantage of this approach is the preservation of the confidence of classification. In addition, every entry of X will be non-negative. However, to use this approach, one has to know or estimate the number of clusters. Furthermore, it might suffer from local minima due to the non-convexity of the optimization problem.

5. Experiment

We applied our two semi-supervised algorithms outlined in §3 to a face-image clustering problem – given a database of face images, cluster the images based on identity. This problem fits well into the semi-supervised clustering problem, since some of the faces may be tagged by identity. These tags imply a set of link constraints: a must-link constraint for each pair of data points that are tagged the same, and a cannot-link constraint for each pair of data points that are tagged as different identities.

We used two freely-available face databases: the ORL face database 1, and AT&T's ORL face database 2. The ORL face database contains 11 example images for each of 15 individuals, while the ORL database is larger, with 40 individuals and 10 examples per individual. In both databases, the images were acquired under different lighting configurations, with and without glasses, and different facial gestures. The example images for 15 individuals from the ORL face database are shown in Fig. 1(a), and 11 images for one individual are shown in Fig. 1(b).



(a) (b)

Fig. 1. ORL database illustration.

As a preprocessing step, Principal Component Analysis (PCA) was used to reduce the size of the data and hopefully remove irrelevant dimensions. Given a matrix $X \in R^{d \times N}$ whose columns correspond to face images, PCA computes an m ≤ d dimensional approximate representation $X \approx U_m \Lambda_m V_m^T$, where $U \in R^{d \times m}$ and $V \in R^{d \times m}$ are matrices with orthogonal columns, and $\Lambda_m \in R^{m \times m}$ is a diagonal matrix of singular values. This approximation implies an m-dimensional coordinate representation for the set of face images: $X_{PCA} = U_m^T X = \Lambda_m V_m^T$. The matrix X_{PCA} was input as the data matrix to our algorithms.

We had a parameter – the percentage of constrained data, which is the percentage of data labeled a-priori. To test each value of percent labeled data, we ran 25 uniformly random initializations to collect labels. Then we averaged the results over those 25 runs.

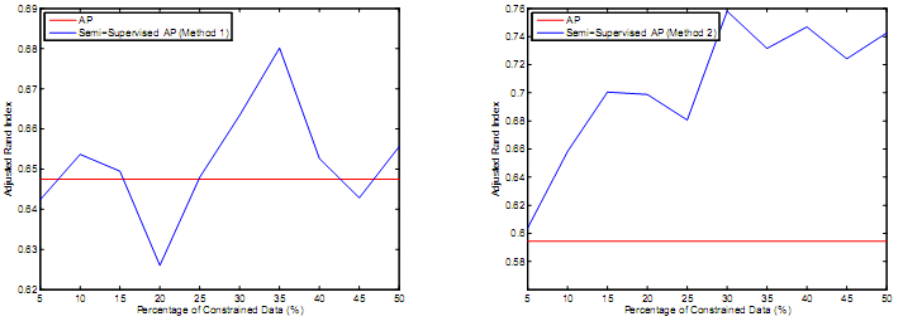
Selecting an objective method to evaluate the clustering performance is important. When the ground truth label information is not available, reconstruction error is often used as a measure of exemplar-based clustering accuracy, which is defined as the sum of errors incurred by approximating each data point by its cluster exemplar. This does not give much insight into how close the clustering is to the true data partition, since the clustering with minimal

reconstruction error is the trivial one – assign each data point to its own cluster. Since we had the true face-image cluster labels, we instead chose to use the adjusted Rand index [7] to evaluate the clustering produced by affinity propagation and semi-supervised affinity propagation.

The adjusted Rand index R is equal to the corrected-for-chance probability that the two clusterings agree on whether or not two randomly selected elements from X belong in the same cluster. Thus, $R \in [0, 1]$, with $R = 1$ implying that the two clusterings partition the data the same.

Since the semi-supervised algorithm takes advantage of side information that affinity propagation does not have access to, we made the comparison fair in the following way: when computing the adjusted Rand index on the semi-supervised clustering result, we excluded data points that were labeled a-priori.

Fig. 2 plots the adjusted Rand index for both affinity propagation and semi-supervised affinity propagation using the two methods. Although the first method based on altering the similarities using a Gaussian kernel does not work well for face clustering, as shown in Fig. 2(a), the second method based on metric learning effectively utilizes the label information to improve the clustering performance. When applying the MTP algorithm proposed in [14], we also noticed idiosyncratic results similar to the first method. Why is the first approach (and the MTP algorithm) unsuccessful in face clustering? The mechanism of these approaches is to draw closer the unsatisfied must-link pairs, and expel the cannot-link pairs. This is accomplished by encouraging the neighbors (in the original feature space) of these points to either share cluster exemplars or differ in cluster exemplars. If we are given good discriminative features in the first place, i.e. data points that already appear to be well separated, then the first approach as well as Givoni's method [6] should work well. However, for high dimensional datasets such as face images, neither the original pixel space nor the reduced PCA space partitions the faces well. As a result, propagating label information from a constrained datapoint to its neighbors may actually pass information across different clusters.

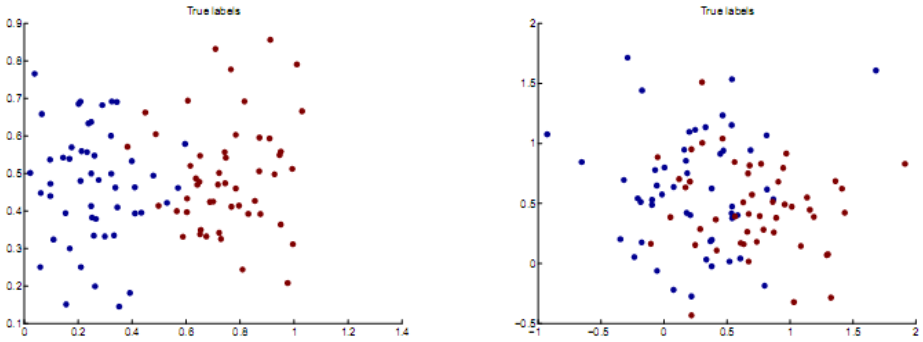


(a) Rand Index for the Gaussian method (b) Rand Index for the Metric Learning method

Fig. 2. Clustering performances of semi-supervised.

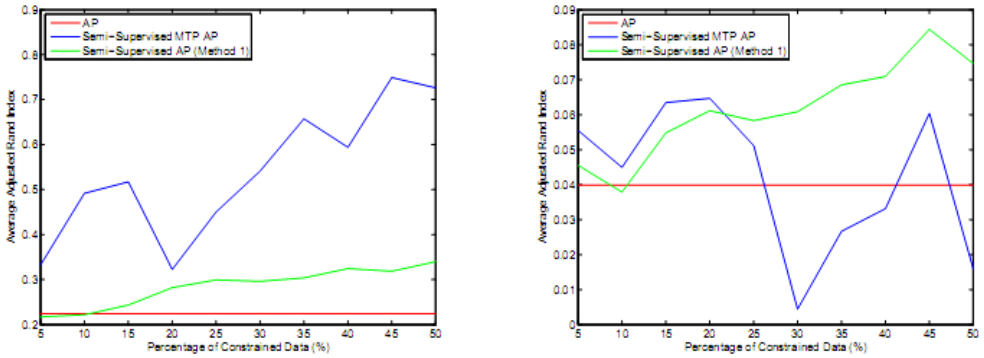
To validate this claim, we tested both the Gaussian similarity method and the MTP algorithm on a simple synthetic data example. We generated data from a mixture of two Gaussians, and varied the separation between the two classes. While both algorithms effectively utilize the partial label information when the clusters are separated, in this example, the side information adversely influences the MTP algorithm when there is overlap between the clusters. In other simulations, we see similar effects on the Gaussian similarity method.

In contrast, our proposed metric learning method performs well on face clustering because a new distance metric is learned for the original data from the partial labels. In this sense, it performs feature selection and clustering simultaneously. The converged diagonal values of the matrix D inform the relative importance of the different features of the data. For example, Fig. 3(a) records 50 converged diagonal entries of D (averaged from 25 random runs), corresponding to the first 50 principal components of the face dataset. Fig. 3(b) shows the components with 5 largest weights, and Fig. 3(c) shows the least 5. An important remark about Fig. 3 is that the largest principal components do not necessarily achieve the highest weights. This is because leading principal components seek to maximize the variance of the data, so they may not necessarily serve as good discriminative features between clusters. Actually, as shown in Fig. 5, the second and third principal components both received some of the lowest weights. The second principal component (left, first row) seems to capture left/right illumination variation and the third principal component (middle, second row) appears to capture the large hair variance. The most weighted components in Fig. 4 are instead the 7, 12, 13, 9, and 6th components, which appear to be more discriminative. The algorithm therefore allows us to identify the set of informative features and has the potential of effective dimensionality reduction.



(a) 2-class clustering with low overlap between classes

(b) 2-class clustering with large overlap between classes



(c) AP and Semi-Supervised AP results on data in Figure 4(a)

(d) AP and Semi-Supervised AP results on data in Figure 4(b)

Fig. 3. The Semi-Supervised Synthetic Clustering Example.

Fig. 4. Analysis of the metric learning approach.

Fig. 5. Analysis for ORL face database.

6. Conclusions

Semi-supervised clustering employs a small amount of labeled data to aid unsupervised learning. Previous work in the area has utilized supervised data in one of two approaches: 1) constraint-based methods that guide the clustering algorithm towards a better grouping of the data, and 2) distance-function learning methods that adapt the underlying similarity metric used by the clustering algorithm. This paper provides new methods for the two approaches as well as presents a new semi-supervised clustering algorithm that integrates both of these techniques in a uniform, principled framework. Experimental results demonstrate that the unified approach produces better clusters than both individual approaches as well as previously proposed semi-supervised clustering algorithms.

By ablating the individual components of our integrated approach, we have experimentally compared metric learning and constraints in isolation with the combined algorithm. Our results have shown that by unifying the advantages of both techniques, the integrated approach outperforms the two techniques individually. We have shown that using individual metrics for different clusters, as well as performing feature generation via a full weight matrix in contrast to feature weighting with a diagonal weight matrix, can lead to improvements over our basic algorithm. Extending our approach to high-dimensional datasets, where Euclidean distance performs poorly, is the primary avenue for future research. Other interesting topics for future work include selection of most informative pairwise constraints that would facilitate accurate metric learning and obtaining good initial centroids, as well as methodology for handling noisy constraints and cluster initialization sensitive to constraint costs.

Acknowledgements

This work is supported by the science foundation of Guangdong province under grant (No. S2013010013307).

References

[1]. L. Kaufman and P. J. Rousseeuw, Finding groups in data an introduction to cluster analysis. Series in probability and mathematical statistics, *Wiley*, New York, 1990.

[2]. A. K. Jain and R. C. Dubes, Algorithms for clustering data, *Prentice-Hall*, Englewood Cliffs, NJ, 1988.

[3]. D. Jiang, C. Tang, and A. Zhang, Cluster analysis for gene expression data: A survey, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, Issue 11, 2004, pp. 1370–1386.

[4]. A. K. Jain, M. N. Murty, and P. J. Flynn, Data clustering: A review, *ACM Computer Survey*, Vol. 31, Issue 3, 1999, pp. 264-323.

[5]. C. Ruiz, M. Spiliopoulou, E. Menasalvas, Density-based semi-supervised clustering, *Data Mining and Knowledge Discovery*, Vol. 3, Issue 21, 2010, pp. 345-370.

[6]. Ioannis A Maraziotis, A semi-supervised fuzzy clustering algorithm applied to gene expression data, *Pattern Recognition*, Vol. 45, Issue 1, 2012, pp. 637-648.

[7]. B. J. Jain, K. Obermayer, Elkan'S k-means algorithm for graphs, in *Proceedings of the 9th Mexican International Conference on Artificial Intelligence Conference on Advances in Soft Computing (MICAI'10)*, Berlin Springer-Verlag, 2010, pp. 22-32.

[8]. M. E. Ares, Javier Parapar, Barreiro Alvaro, An experimental study of constrained clustering effectiveness in presence of erroneous constraints, *Information Processing and Management*, Vol. 48, Issue 3, 2012, pp. 537-551.

[9]. S. Faüer, F. Schwenker, Semi-supervised kernel clustering with sample to cluster weights, in *Proceedings of the 1st IAPR TC3 Conference on Partially Supervised Learning (PSL'11)*, Springer-Verlag, Berlin, 2011, pp. 72-81.

[10]. Sugato Basu, Mikhail Bilenko, and Raymond J. Mooney, A probabilistic framework for semi-supervised clustering, in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, New York, NY, USA, 2004, pp. 59-68.

[11]. Brendan J. Frey and Delbert Dueck, Clustering by passing messages between data points, *Science*, Vol. 315, No. 5814, 2007, pp. 972–976.

[12]. I. Givoni and B. Frey, Semi-supervised affinity propagation with instance-level constraints, in *Proceedings of the 12th International Conference on Articial Intelligence and Statistics (AISTATS)*, Clearwater Beach, Florida, USA, 2009, pp. 1-8.

[13]. Nizar Grira, Michel Crucianu, and Nozha Boujemaa, Unsupervised and semi-supervised clustering: a brief survey, A review of machine learning techniques for processing multimedia content, *Report of the MUSCLE European Network of Excellence (FP6)*, 2004, (http://www-rocq.inria.fr/%7Ecrucianu/src/BriefSurveyClustering.pdf).

[14]. A. K. Jain, M. N. Murty, and P. J. Flynn, Data clustering: a review, *ACM Computing Surveys*, Vol. 31, Issue 3, 1999, pp. 264-323.

[15]. Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger, Factor graphs and the sum-product algorithm, *IEEE Transactions on Information Theory*, Vol. 47, Issue 2, 2001, pp. 498-519.

[16]. Michele Leone and Martin Weigt, Unsupervised and semi-supervised clustering by message passing: Soft-constraint affinity propagation, *The European Physical Journal B*, Vol. 66, Issue 1, November 2008, pp. 125-135.

[17]. Lihi Zelnik-Manor and Pietro Perona, Self-tuning spectral clustering, Advances in Neural Information Processing Systems 17, *MIT Press*, 2004, pp. 1601-1608.