PAPER

Semi-Supervised Clustering Based on Exemplars Constraints

Sailan WANG[†a)], ***Member***, **Zhenzhi YANG**[††], **Jin YANG**[†††], ***and*** **Hongjun WANG**[††††], ***Nonmembers***

SUMMARY In general, semi-supervised clustering can outperform unsupervised clustering. Since 2001, pairwise constraints for semi-supervised clustering have been an important paradigm in this field. In this paper, we show that pairwise constraints (ECs) can affect the performance of clustering in certain situations and analyze the reasons for this in detail. To overcome these disadvantages, we first outline some exemplars constraints. Based on these constraints, we then describe a semi-supervised clustering framework, and design an exemplars constraints expectation–maximization algorithm. Finally, standard datasets are selected for experiments, and experimental results are presented, which show that the exemplars constraints outperform the corresponding unsupervised clustering and semi-supervised algorithms based on pairwise constraints.

key words: *semi-supervised clustering, mixture model, pairwise constraints, exemplars constraints*

1. Introduction

Clustering is an important aspect of unsupervised learning and can be regarded as an independent tool or preprocessing step for other learning models. Clustering is a totally unsupervised learning method, but in real situations, some degree of prior knowledge about the data can often be obtained. Naturally, if this prior knowledge can be integrated into the algorithm, the performance will improve. In the past decade, semi-supervised clustering has become an important variant of the traditional clustering paradigm [1], [2]. Of the existing semi-supervised clustering methods, the instance levels of must-link and cannot-link constraints are very popular because they are simple, effective, and interpretable. Wagstaff [3] defined a must-link (ML) constraint as one in which two data points must be in the same cluster, whereas a cannot-link (CL) constraint states that two data points must not be placed in the same cluster. Moreover, she proposed a typical constrained k-means algorithm based on these constraints. These advances led to many semi-supervised clustering methods based on pairwise constraints, such as constrained complete-link [4], constrained expectation–maximization (EM) [5], HMRFK-means [6], MPCKmeans [7], kernel methods [8]–[13], matrix factorization-based methods [14], and constraint projection [15]–[18]. Kulis et al. [8] reported a kernel algorithm to minimize a semi-supervised clustering objective function. Their method can cluster both vector-based and graph-based data. Shental et al. [5] developed a framework to incorporate MLs and CLs into the mixture model estimation procedure. Pairwise constraints on the original data points can be projected to a low-dimensional space for ensemble learning [16]. The new data points in the low-dimensional space contain additional information about the original data that can be used for semi-supervised learning. Pairwise constraints can also be transformed into a sequence of convex quadratic programming problems under a constrained concave–convex procedure [18]. A sub-gradient projection optimization algorithm can then be used to solve the problems. Constraint clustering can also be employed to find a set of features in scene images [19], and this set can then be integrated using pairwise constraints to enable scene classification. Elite pairwise constraints are suitable in each optimal partition, and do not present any conflicts [20]. Pairwise constraints can be constructed according to labels, but they can also be generated independently. Based on labels and constraints, the reproduction of constraints can be inferred and propagated [21]. Online constraint clustering algorithms [22] are designed to handle large datasets while preserving their simplicity and effectiveness. For sparse data, a genetic algorithm can be applied to preserve the pairwise constraints while simultaneously completing the dimensionality reduction procedure.

Furthermore, the neighbors of constraint data points can be considered as potential constraints, which is effectively propagating additional information [23], [24]. Motivated by active learning, both Xu et al. [25] and Basu et al. [26] used the active selection of pairwise constraints to achieve improved clustering performance, and Basu et al. [26] extended the pairwise constraints for active semi-supervised clustering. An alternative approach is to view clustering as a form of generative modeling, and learn a semi-supervised generative model [27] using approximate Bayesian posterior inference in the paper, where a function is learned from class labels and latent variables associated with the data.

Most methods consider the positive side of pairwise constraints. One exception is a paper by Davidson [28], who

Manuscript received May 13, 2016.
Manuscript revised December 15, 2016.
Manuscript publicized March 21, 2017.

[†]The author is with the School of Tourism, Sichuan University, China.
[††]The author is with Department of Computer Science and Software Engineering, Jincheng Institute of SiChuan University, China.
[†††]The author is with the Department of Computer Science, Leshan Normal University, China.
[††††]The author is with the School of Information Science and Technology, Southwest Jiaotong University, China.
a) E-mail: wangsailan@stu.scu.edu.cn

DOI: 10.1587/transinf.2016EDP7201

considered their negative influence. Motivated by this, we consider the following:

1. Pairwise constraints have a negative influence, decreasing the accuracy of clustering and, in some cases, increasing the runtime. To some extent, the model proposed in this paper solves these problems.
2. What are the optimal pairwise constraints? and how can we find them and improve the clustering results?

In this paper, we address these two motivations and illustrate ECs in detail. There are two main contributions of this paper.

1. We discover that not all constraints improve the clustering accuracy, and in some cases, they decrease the accuracy and increase the computational load. We study the problem of selecting good data points to form constraints. This is the first attempt to state ECs for the improvement of clustering performance. We also explain why ECs improve clustering performance. The reason is that ECs can reduce the *ambiguousness* and increase the *coherence* [28] of the constraints.
2. A semi-supervised clustering framework based on ECs is designed, and an ECs mixture model (ECMM) is proposed for semi-supervised clustering. Furthermore, the difference between ECMM and constrained EM is illustrated, and the reason why ECMM improves the clustering results is discussed.

The remainder of this paper is organized as follows. ECs are illustrated in detail in Sect. 2. In Sect. 3, a semi-supervised clustering framework based on ECs is proposed, and a mixture model is formulated using these ECs. Experimental results are presented in Sect. 4. The paper ends with our conclusions in Sect. 5.

2. Exemplars Constraints

In this work, we propose a new constrained clustering method as an extension of pairwise constraint clustering. The disadvantages are illustrated in detail using examples. We will show that, compared with pairwise constraints clustering, ECs clustering improves performance, reduces computational complexity, and requires far fewer labeled data points. The notations used in the paper are summarized in Table 1.

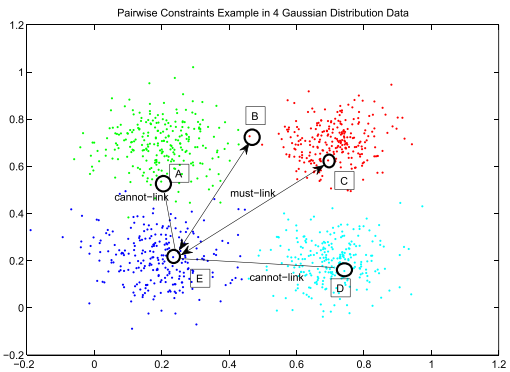
Table 1 Notations

Symbol	Explanation
K	the number of clusters
N	the number of data points
X	the set of data points
M	the set of must-links
C	the set of cannot-links
L(ϵ)	lower bound function of ϵ
i,j,k,e	the variables for count
π	the mixing coefficients
Θ	the set of parameters of a mixture model
Σ, μ	variance and mean of a Gaussian distribution

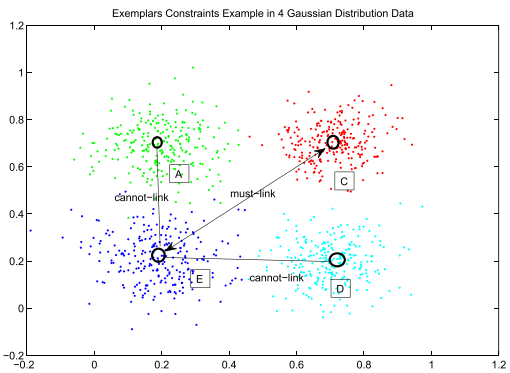
2.1 Pairwise Constraints

Pairwise constraints [3] consist of ML and CL constraints. An ML constraint is one in which two data points must be in the same cluster, and a CL constraint states that two data points must be in different clusters. These are simple but effective statements, and have therefore received considerable attention. However, there are four disadvantages of pairwise constraints.

1. Semi-supervised clustering based on pairwise constraints cannot guarantee improved clustering accuracy, and may even decrease the accuracy of clustering compared with the corresponding unsupervised clustering algorithm.
2. The runtime is long, because the algorithms consider many pairs in dealing with pairwise constraints. Furthermore, the algorithms may not converge if there are conflicts between constraints.
3. Generally, many constraints are needed to achieve good performance, but labels are often unavailable and computationally expensive. Even when labeled data are available, they may conflict with one another and affect the results of clustering.
4. Some data points with pairwise constraints are ambiguous within a cluster, such as data point B in Fig. 1 (a),



(a) Examples of pairwise constraints



(b) Examples of exemplars constraints

Fig. 1 Pairwise constraints vs. exemplars constraints

Table 2 The average micro-precision (MP) of different algorithms on each dataset. Semi-supervised clustering algorithms of COP-kmeans and constrained EM cannot always outperform the corresponding unsupervised clustering algorithms, such as kmeans and EM. *Num* denotes the number of pairwise constraints for semi-supervised clustering.

Dataset	Num	Kmeans	COP-Kmeans	EM	Constrained EM
haberman	30	0.5121±0.0254	**0.5852**±0.0216	0.6667±0.0234	**0.6729**±0.0421
iris	15	0.8933±0.0015	**0.9067**±0.0012	**0.9667**±0.0000	0.9660±0.0009
wdbc	50	**0.8541**±0.0002	0.8489±0.0023	**0.9554**±0.0008	0.9513±0.0021
wine	17	0.6632±0.0122	**0.7130**±0.0042	0.7528±0.0024	**0.7752**±0.0087
ionosphere	35	**0.7123**±0.0006	0.7068±0.0026	0.8168±0.0034	**0.8324**±0.0031
bupa	35	0.4840±0.0012	**0.5569**±0.0122	**0.5072**±0.0055	0.4991±0.0080
balance	60	0.5158±0.0048	**0.5506**±0.0016	0.5186±0.0042	**0.5280**±0.0016
heart	28	0.5926±0.0221	0.5926±0.0042	0.7148±0.0025	**0.7259**±0.0034

Table 3 Computation time (s) of different algorithms on each dataset. It is clear that the semi-supervised clustering algorithms of COP-kmeans and constrained EM take longer than unsupervised clustering algorithms.

Dataset	Num	kmeans	COP-kmeans	EM	Constrained EM
Haberman	30	0.0031	**0.0375**	0.0500	**0.2781**
iris	15	0.0047	**0.0187**	0.0437	**0.7688**
wdbc	50	0.0047	**0.0938**	0.1250	**0.3312**
wine	17	0.0016	**0.0297**	0.0938	**0.7641**
ionosphere	35	0.0652	**0.2316**	0.0984	**0.2568**
bupa	35	0.0078	**0.0359**	0.1047	**0.4625**
balance	60	0.0078	**0.1672**	0.1094	**0.7344**
heart	18	0.1652	**0.3473**	0.2023	**0.3021**

which may cause a constraint conflict.

For an experimental test, COP-kmeans [3], kmeans, constrained EM [5], and EM algorithms were applied to several UCI datasets. The accuracies and runtime are listed in Tables 2 and 3, respectively. There are positive and negative results; the negatives are more interesting. The algorithms based on pairwise constraints can have lower clustering accuracy (Table 2). Additionally, the time complexities of Cop-kmeans and kmeans are $O(nkct)$ and $O(nkt)$, respectively. Thus, more constraints will lead to increased computation time. From Table 3, it is clear that semi-supervised clustering algorithms require more computation time than the corresponding unsupervised clustering algorithms.

2.2 Exemplars Constraints

Exemplars are the center objects that represent a group in a dataset [29]. For example, consider two clusters of 1000 data points generated by four Gaussian distributions, as shown in Fig. 1. In Fig. 1 (b), A, C, D, and E are exemplars of data points. In this example, there are a total of $\sum_{i=1}^{1000-1} i = 499500$ pairwise constraints. If exemplars were used to represent all the other data points in the same Gaussian distribution, there would be a total of only $\sum_{i=1}^{4-1} i = 6$ pairwise constraints. These are known as ECs. From this point of view, ECs are special cases of pairwise constraints.

We consider ECs for two reasons: the disadvantages of pairwise constraints, and the advantages of ECs. There are generally too many pairwise constraints within a dataset, and using them is therefore time-consuming, as can be inferred from Table 3. Moreover, semi-supervised learning based on pairwise constraints cannot always improve the clustering performance, and is sometimes worse than unsupervised learning (see Table 2). Finally, ECs are less likely to cause conflicts. ECs can be split into two categories: exemplar-must-link (EML) denotes that the two exemplars must be in the same cluster, and exemplar-cannot-link (ECL) implies that the two exemplars must be in different clusters.

In practice, clustering has no ground-truth label information with which to judge the correctness of prior knowledge. Indeed, the prior knowledge could be uncertain and affect the clustering accuracy. If prior knowledge is used to generate the ECs, then these constraints may conflict. In this paper, we use *coherence* [28] to avoid conflicting constraints. Coherence measures the degree of agreement within the constraints themselves, with respect to a given distance metric [28].

Another measure, *ambiguousness*, can be defined to evaluate the quality of the ECs and pairwise constraints. Ambiguousness refers to the normalized distance to the cluster center. If the distance of a point to its two nearest cluster centers is the same, the ambiguousness of the point is 1. The *ambiguousness* can be measured by

$$\Lambda(x_i) = \begin{cases} \frac{|x_i-\mu_k|}{|x_i-\mu_j|} & \text{if} \quad |x_i-\mu_k| \le |x_i-\mu_j| \\ \frac{|x_i-\mu_j|}{|x_i-\mu_k|} & \text{if} \quad |x_i-\mu_k| > |x_i-\mu_j| \end{cases} \tag{1}$$

where μ_k and μ_j are the two nearest cluster centers to x_i; $\Lambda(x_i)$ takes values from 0 to 1. More *ambiguousness* leads to less *coherence*, and less *coherence* is an important factor in decreasing the clustering performance. In contrast, less *ambiguousness* is likely to lead to better performance. Thus, ECs are better than pairwise constraints in improving the clustering performance. For example, in Fig. 1 (a), $\Lambda(A) = 0.68$, $\Lambda(B) = 0.97$, and $\Lambda(C) = 0.06$. Point B has the most *ambiguousness* of these three data points, and will thus degrade the clustering performance. In Fig. 1 (b), $\Lambda(A) = 0.05$, $\Lambda(C) = 0.03$, and $\Lambda(D) = 0.02$; all *ambiguousness* values are close to zero, which suggests high *coherence* in the constraint set. High *coherence* in the constraint set will improve the clustering performance, as shown in [28].

2.3 Finding Exemplars

In this subsection, we address the problem of finding a good exemplar that represents the other data points in a group. Formally, an objective function is defined as follows:

$$F(x_e) = \frac{1}{N}\sum_{e=1}^{K}\sum_{\{x_i \to x_e, i=1\}}^{N} |x_i - x_e|^2, \{i_1^N, e_1^K\}, \tag{2}$$

where x_e is an exemplar and $x_i \to x_e$ means that x_e can represent the data point x_i. There are a number of methods for deriving objective functions to find the exemplars [30]. Exemplar-based clustering is the task of not only partitioning each group, but also of identifying its most representative member [30]. Affinity propagation (AP) [31], density

peak (DP) [32], and k-centroid algorithms have been developed to find exemplars. In this paper, we do not place any requirement on the number of exemplars to be found, although this is commonly greater than the number of classes (the ground truth).

2.4 Exemplars Constraints Propagation

EML constraints are those that require two exemplars to be in the same cluster. EML constraints are positively transitive as pairwise constraints [33]. Let x_i, x_j, and x_k be exemplars such that

$$(x_i, x_j) \in M, (x_i, x_k) \in M \Longrightarrow (x_j, x_k) \in M, \tag{3}$$

where M is the set of EMLs. ECL constraints specify that two exemplars must be placed in different clusters. Let x_i, x_j, and x_k be exemplars such that

$$(x_i, x_j) \in C, (x_i, x_k) \in M \Longrightarrow (x_j, x_k) \in C, \tag{4}$$

where C is the set of ECLs. Let x_i, x_j be the exemplars, and let x_k be a neighbor of x_i. Then,

$$(x_i, x_j) \in C \Longrightarrow (x_j, x_k) \in C, \tag{5}$$
$$(x_i, x_j) \in M \Longrightarrow (x_j, x_k) \in M. \tag{6}$$

For different types of dataset, neighbors are selected in different ways. Given the data $X = \{x_1, x_2, \ldots, x_N\}$ with m dimensions, let k_i and k_j be two connected components, $M = \{(x_i, x_j)|x_i \in k_i; x_j \in k_i\}$ be the EML constraint set, $C = \{(x_i, x_j)|x_i \in k_i; x_j \in k_j; k_i \neq k_j\}$ be the ECL constraint set, and $\mu = \{x_l|\rho \geq d(x_e); x_l \in X; l = (1, \ldots, L)\}$ be the set of neighbors of x_i. The simplest approach is to use the Euclidean distance to choose the neighbors, so $d(x_e) = \sqrt{||x_i - x_e||^2}$. The geodesic distance can also be used to select neighbors. We use a Gaussian function centered at the given constraint x_A, x_B to determine the weight of x_i, x_j, because a Gaussian function can propagate constraints that are closest to the source ECs and will fall off smoothly. If the dataset is in discrete space, we can use the normalized mutual information (NMI) $NMI(x_i, x_j) = \frac{I(x_i, x_j)}{\sqrt{H(x_i)H(x_j)}}$ [34] for constraint propagation to select neighbors.

3. A Semi-Supervised Clustering Framework Based on Exemplars Constraints

3.1 Framework Description

In this subsection, we describe a framework based on ECs for semi-supervised clustering. The framework proceeds as follows.

Framework 1: (A semi-supervised learning framework based on ECs [SSL-EC])
Input: Dataset $\{x_i\}_{i=1}^N$, where x_i denote data points.
Output: Cluster membership of every point.

1. Select one algorithm to find exemplars in a dataset:
   a. k-centroid is used to find exemplars in the dataset;
   b. Or AP is used to find exemplars in the dataset;
   c. Or DP is used to find exemplars in the dataset.
2. The set of EMLs and ECLs is generated according to the labels of the exemplars.
3. Avoid EC conflicts: the coherences of M and C are calculated. $coh(M) = \sum_{i=1}^{N_M} \frac{N_M - con(m_i)}{N_M}$ where $con(m_i)$ is a function that returns 1 if m_i is not a conflict pair, according to the pair's true labels; otherwise, the function returns 0. $coh(C)$ is calculated in the same way.
4. If $coh(M) < \omega$ or $coh(C) < \omega$, then the conflict constraints are deleted one by one until $coh(M) \geq \omega$ and $coh(C) \geq \omega$ (generally, if we want to delete all conflicting constraints, $\omega = 1$).
5. Select one constraint-based algorithm (clustering algorithms based on ML and CL) for clustering:
   a. COP-kmeans is used for clustering;
   b. Or constrained EM is used for clustering;
   c. Or constrained FCM is used for clustering.

In this framework, the EC conflicts are detected and avoided.

3.2 An Exemplars Constraints Mixture Model

In statistical machine learning, mixture models are very popular for unsupervised learning problems, as they can sample data from a weighted sum of several distributions. Gaussian mixture models (GMMs) are usually applied to process continuous data that is assumed to follow a Gaussian distribution. Let $X = \{x_i\}, i \in (1, \ldots, N)\}$ be the set of all data points, $C = \{(x_a, x_b), (a, b) \in (1, \ldots, S), S \leq N\}$ be the set of ECs, and $Y = \{y_i, \ldots, y_N\}$ be the assignment of the original data points. Finally, let E denote the event. More formally, a GMM is given by:

$$P(x|\Theta) = \sum_{i=1}^{N} \pi_i p(x|\theta_i), \tag{7}$$

where π_i is the weight of each Gaussian distribution and θ_i is its corresponding parameter. The expectation of the log-likelihood is the following:

$$E[\log(p(X, Y|\Theta_{(t+1)}, E))|X\Theta_t, E] = \sum_{Y} \log(p(X, Y|\Theta_{(t+1)}, E))P(Y|X, \Theta_t, E). \tag{8}$$

Using Bayesian rules,

$$P(Y|X, \Theta, E) = \frac{\prod_{j=1}^{S} \delta y_j p(y_i|x_j, \Theta)}{\sum_Y \prod_{j=1}^{S} \delta y_j p(y_i|x_j, \Theta)}. \tag{9}$$

The points in the set of ECs depend on one another. Thus,

$$P(X, Y|\Theta^{(t+1)}, E) = \prod_{j=1}^{S} \pi_{y_j} \prod_{i=1}^{N} p(x_i, y_i|\Theta^{(t+1)}). \tag{10}$$

Hence, the log-likelihood is:

$$\log P(Y|X,\Theta,E) = \sum_{j=1}^{S}\sum_{x_i\in M}\log P(x_i|y_i,\Theta^{(t+1)}) + \sum_{j=1}^{S}\log(\pi_{y_j}), \quad (11)$$

and hence the posterior probability is:

$$P(y_j = S\,|x_j,\Theta) = \frac{\pi_k\prod_{x_i\in M}p(x_i|y_i^j = k,\Theta)}{\sum_{k=1}^{K}\pi_k\prod_{x_i\in M}P(x_i|y_i,\Theta)}. \quad (12)$$

Commonly, EM is used to obtain the probability by estimating the parameters as follows.

$$\pi^{(t+1)} = \frac{1}{N}\sum_{i=1}^{N}P(y_i = k|x_i,\Theta^t), \quad (13)$$

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^{N}\overline{X}_i P(y_i = k|x_i,\Theta^t)|x_i|}{\sum_{i=1}^{K}P(y_i = k|x_i,\Theta^t)|x_i|}, \quad (14)$$

$$\Sigma_k^{(t+1)} = \frac{\sum_{i=1}^{N}\Sigma_{ik}^t P(y_i = k|x_i,\Theta^t)|x_i|}{\sum_{i=1}^{K}P(y_i = k|x_i,\Theta^t)|x_i|}. \quad (15)$$

In the original EM algorithm, the E-step sums all of the probabilities of different assignments. However, in this modified EM, the E-step only sums the probabilities that comply with the constraints.

3.3 EM Algorithm Procedure

In this subsection, the EM algorithm for ECs is described in detail according to the above inference. The proposed algorithm requires the initial parameters (π_0,μ_0,Σ_0). It can be started from a random guess, with each data point in each constituent distribution computed by calculating the expectation values for the membership variables. The algorithm alternates between the E-step and M-step until convergence, and the optimal parameters (π^*,μ^*,Σ^*) can be found for the proposed model.

Exemplars Constraints EM Algorithm:
Input: Dataset and random parameters: $\{x_i\}_{i=1}^N$ and (π_0,μ_0,Σ_0)
Output: (π^*,μ^*,Σ^*), where π^* is the probability of cluster membership of every point.

1. AP is used to find exemplars according to the parameter (π_t). The sets M and C can be changed in each iteration according to (π_t).
2. Avoid EC conflicts according to the third and fourth steps of Framework 1.
3. E-Step: Calculate the expectation of the log-likelihood over all possible assignments of data points that comply with the given constraints.

$$\pi^{(t+1)} = \frac{1}{N}\sum_{i=1}^{N}P(y_i = k|x_i,\Theta^t).$$

4. M-Step: Maximize the expectation by differentiating with respect to the current parameters.

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^{N}\overline{X}_i P(y_i = k|x_i,\Theta^t)|x_i|}{\sum_{i=1}^{K}P(y_i = k|x_i,\Theta^t)|x_i|},$$

$$\Sigma_k^{(t+1)} = \frac{\sum_{i=1}^{N}\Sigma_{ik}^t P(y_i = k|x_i,\Theta^t)|x_i|}{\sum_{i=1}^{K}P(y_i = k|x_i,\Theta^t)|x_i|}.$$

After t iterations, the expectation value is π^t. In the $(t+1)^{th}$ iteration,

$$\sum_{i=1}^{N}L(\pi^t,\mu^t,\Sigma^t) \le \sum_{i=1}^{N}L(\pi^t,\mu^{(t+1)},\Sigma^{(t+1)}) \quad (16)$$

$$\le \sum_{i=1}^{N}L(\pi^{(t+1)},\mu^{(t+1)},\Sigma^{(t+1)}). \quad (17)$$

The first inequality holds because, in the E-step, (16) is the maximum of $L(\pi^t,\mu^{(t+1)},\Sigma^{(t+1)})$. The second inequality holds because, in the M-step, (17) is the maximum of lower bound function $L(\pi^{(t+1)},\mu^{(t+1)},\Sigma^{(t+1)})$. Therefore, the objective function is non-decreasing until convergence [35], which means that the proposed algorithm can be convergence step by step on the condition without any constraint conflicts. We discovered that the algorithm in the paper [5] oscillated in the E- and M-steps because of constraints conflict. So in this paper, the avoiding EC conflicts steps are added before E-step, which can solve the vibrating problem among E- and M-steps. Bruneau et al. [36] proved that the EM algorithm framework can be convergence if the data points follow exponential families distribution. ECEM is designed according to the EM algorithm framework, moreover, the data points follow a Gaussian distribution which belongs to exponential families. More recently, Wu et al. [37] use oracle convergence theorem, empirical convergence theory, optimal empirical convergence theorem and optimal rate convergence theorem to formulate a theoretical framework to prove the convergence of EM algorithm. More detail information can be found in the paper [37]. Regarding the computational complexity, before the algorithm goes into the E-step and M-step iterations, avoiding EC conflicts requires $o(N\varsigma)$ operations, where ς is the percentage of data points involved. In general, ς ranges from 1–10% and $\varsigma < 1$. Calculating all possible assignments of data points to sources has a complexity in each E-step of $o(NK)$. In the M-step, all μ and Σ are updated on each iteration, which has a complexity of $o(NK)$. Overall, the complexity of the proposed algorithm is $o(2NKt + N\varsigma)$, where N is the number of data points, K is the number of clusters, and t is the number of iterations. The complexity of constrained EM is $o(2NK(t+s))$, where s is the number of iterations of constraint conflict processing and $s => 1$. Constrained EM has a higher computational load than the proposed algorithm, because $o(2NK(t+s))$ is greater than $o(2NKt + N\varsigma)$.

In [5], the constraints modify the E-step (expectation

computation) such that the sum is taken only over the assignments that comply with the given constraints. The most important difference between the proposed ECs EM and constrained EM [5] is that ECs EM can avoid constraint conflicts, whereas constrained EM cannot. These constraint conflicts pose two disadvantages to constrained EM. First, constrained EM requires many more iterations (E-step and M-step) than ECs EM, which increases the time to convergence. Second, if all pairwise constraints conflict with one another at extremely case such as $(C \ni (x_i, x_j) \in M)$, constrained EM will oscillate in the E- and M-steps. The preprocessing step can be used to delete the extremely case to avoid the algorithm vibration.

4. Empirical Study

4.1 Experimental Setup

Eighteen real-world datasets from the UCI machine learning repository were used to conduct experiments. The number of objects, features, and classes of each dataset are listed in Table 4.

To evaluate clustering performance, the micro-precision [38] was used to measure the accuracy of the cluster with respect to the true labels. The micro-precision is defined as

$$MP = \sum_{h=1}^{K} a_h / N \tag{18}$$

where a_h denotes the number of objects in cluster h that are correctly assigned to the corresponding class. We identify the *corresponding class* for a cluster h as the true class having the largest overlap with the cluster, and assign all objects in cluster h to that class. Note that $0 \leq MP \leq 1$, with 1 indicating the best possible consensus clustering (i.e., in full agreement with the class labels). We considered pairwise constraints and ECs. Pairwise constraints were constructed as follows: 5% of the data points were randomly selected and the constraints were produced according to the labels of the selected data points. To construct the ECs, the k-centroid was first used to find the exemplars among the datasets. The ECs were produced according to the labels of these exemplars.

Table 4 Number of instances, features, and classes in each dataset.

Dataset	Characteristic	Instances	Features	Categories
iris	real	150	4	3
wdbc	real	569	30	2
wine	real	178	13	3
ionosphere	real	351	34	2
bupa	discrete	345	6	2
balance	discrete	625	4	3
hear	real	270	13	2
haberman	discrete	306	3	2
wave	real	5000	40	3
labor	real	57	16	2
user	real	258	5	4
climate	real	540	18	2
seeds	real	210	7	3
plrx	real	182	12	2
vertebral	real	310	6	2
magic	real	19020	10	2
bank	real	45211	16	2
diabete	real	4839	5	2

4.2 Performance Comparison

To evaluate the proposed algorithm, we compared its performance against five conventional algorithms.

1. Unsupervised clustering:
   a. K-means: perform K-means on the original datasets;
   b. EM: perform EM on the original datasets;
2. Semi-supervised clustering:
   a. COP-Kmeans [3]: perform COP-Kmeans on the original datasets with 5% random constraints;
   b. Constrained EM [5]: perform constrained EM on the original datasets with 5% random constraints;
   c. ECs EM: perform the proposed ECs EM on the original datasets with 5% ECs.

The experiments were repeated 10 times. The averaged results are presented in Table 5. The proposed ECs EM algorithm achieved the best results among the algorithms on 15 datasets. On the balance and bupa datasets, ECs EM scored second best. Among all the results, we focus on those given by the constrained EM algorithm and ECs EM. We can see that ECs EM outperformed constrained EM on each dataset. Table 5 also shows that ECs EM achieved the best average micro-precision (MP), with a value of 0.7702.

Some of the results in Table 5 are very close to one another. To give a more detailed comparison, a $1 \times n$ comparison of Table 6 was performed by means of the Friedman Aligned Rank test [39]. ECs EM was selected as the control method. For the five comparative algorithms and 18 datasets, the aligned values and corresponding ranks are recorded in Table 6.

On average, ECs EM ranked first, with a rank score of 23.5833; pairwise constrained EM ranked second with 29.3889, followed by EM in third with a rank of 33.9167, COP-kmeans in fourth with a rank of 67, and kmeans in fifth with a rank of 73.6111. Under the null hypothesis, the role of the Friedman Aligned Rank test is to check whether there is a significant difference between the measured sum of aligned ranks and the total aligned ranks $\widehat{R}_j = 819$.

$$\sum_{j=1}^{k} \widehat{R}_{\cdot,j}^2 = 1325^2 + 1206^2 + 610.5^2 + 529^2 + 424.5^2 = 4042812.5, \tag{19}$$

$$\sum_{j=1}^{k} \widehat{R}_{i,\cdot}^2 = 221^2 + 242^2 + 230^2 + \ldots + 185^2 = 940397, \tag{20}$$

Table 5 Average accuracies achieved in the experiments. ECs EM is the proposed method.

Dataset	Kmeans	COP-Kmeans	EM	Constrained EM	ECs EM
haberman	0.5121±0.0254	0.5852±0.0216	0.6667±0.0234	0.6729±0.0421	**0.6850**±0.1890
iris	0.8933±0.0015	0.9067±0.0012	0.9667±0.0000	0.9660±0.0009	**0.9767**±0.0016
wdbc	0.8541±0.0002	0.8489±0.0023	0.9554±0.0008	0.9513±0.0021	**0.9554**±0.0001
wine	0.6632±0.0122	0.7130±0.0042	0.7528±0.0024	0.7752±0.0087	**0.8039**±0.0026
ionosphere	0.7123±0.0006	0.7068±0.0026	0.8168±0.0034	0.8324±0.0031	**0.8535**±0.0042
bupa	0.4840±0.0012	**0.5569**±0.0122	0.5072±0.0055	0.4991±0.0080	0.5154±0.0026
balance	0.5158±0.0048	**0.5506**±0.0016	0.5186±0.0042	0.5280±0.0016	0.5376±0.0017
heart	0.5926±0.0221	0.5926±0.0042	0.7148±0.0025	0.7259±0.0034	**0.7333**±0.0026
wave	0.3824±0.0002	0.4723±0.0026	0.8224±0.0008	0.8198±0.0014	**0.8242**±0.0006
labor	0.5789±0.0000	0.5789±0.0000	0.7795±0.0062	0.8024±0.0034	**0.8135**±0.0042
user	0.4845±0.0024	0.5246±0.0037	0.6171 ±0.0044	0.6321±0.0092	**0.6988**±0.0084
climate	0.5093±0.0017	0.5439±0.0025	0.6065 ±0.0164	0.6766±0.0522	**0.6909**±0.0046
seeds	0.8952±0.0021	0.8874±0.0415	0.9219±0.0075	0.9228±0.0042	**0.9248**±0.0094
plrx	0.5138±0.0013	0.5224±0.0562	0.7066±0.0093	0.6973±0.0259	**0.7225**±0.0125
vertebral	0.6710±0.0029	0.6824±0.0032	0.7871 ±0.0000	0.7903±0.0073	**0.7911**±0.0044
magic	0.6491±0.0000	**0.6541**±0.0021	0.6289 ±0.0014	0.6320±0.0036	0.6336±0.0028
bank	0.8536±0.0011	0.8448±0.0023	0.8605 ±0.0008	0.8598±0.0018	**0.8648**±0.0021
diabete	0.5121±0.0038	0.6642±0.0052	0.8380 ±0.0046	0.8324±0.0047	**0.8390**±0.0038
average	0.6265±0.0046	0.6570±0.0094	0.7482 ±0.0052	0.7565±0.0102	**0.7702**±0.0143

Table 6 Aligned observations of five algorithms examined in the experimental study. The ranks in parentheses are used in the computation of the Friedman Aligned Ranks test. The smallest rank is the best.

Dataset	Kmeans	COP-Kmeans	EM	Constrained EM	ECs EM	Total
haberman	-0.1123(84)	-0.0392(67)	0.0423(30)	0.0485(21)	0.0606(19)	221
iris	-0.0486(68)	-0.0352(66)	0.0248(37)	0.0241(38)	0.0348(33)	242
wdbc	-0.0589(69)	-0.0641(72)	0.0424(28.5)	0.0383(32)	0.0424(28.5)	230
wine	-0.0784(78)	-0.0286(65)	0.0112(44)	0.0336(34)	0.0623(17)	238
ionosphere	-0.0721(74)	-0.0776(77)	0.0324(35)	0.0480(22)	0.0691(14)	222
bupa	-0.0285(64)	0.0444(25)	-0.0053(55)	-0.0134(60)	0.0029(50)	254
balance	-0.0143(61)	0.0205(39)	-0.0115(58)	-0.0021(52)	0.0075(46)	256
heart	-0.0792(79.5)	-0.0792(79.5)	0.0430(26)	0.0541 (20)	0.0615(18)	223
wave	-0.2818(90)	-0.1919(88)	0.1582(2)	0.1556(3)	0.1600(1)	184
labor	-0.1317(86.5)	-0.1317(86.5)	0.0689(15)	0.0918(9)	0.1029(5)	202
user	-0.1069(82)	-0.0668(73)	0.0257(36)	0.0407(31)	0.1074(4)	226
climate	-0.0961(81)	-0.0615(70)	0.0011(51)	0.07129(13)	0.0855(11)	226
seeds	-0.0152(62)	-0.0230(63)	0.0115(42)	0.0124(41)	0.01449(40)	248
plrx	-0.1187(85)	-0.1101(83)	0.0741(12)	0.0648(16)	0.0900(10)	206
vertebral	-0.0734(76)	-0.0620(71)	0.0427(27)	0.0459(24)	0.0467(23)	221
magic	0.0114(43)	0.0074(47)	-0.0088(57)	-0.0057(56)	-0.0041(54)	257
bank	-0.0031(53)	-0.0119(59)	0.0038(48)	0.0031(49)	0.0081(45)	254
diabete	-0.2250(89)	-0.0729(75)	0.1009(7)	0.0953(8)	0.1019(6)	185
Total rank	1325	1206	610.5.5	529	424.5	
Average rank	73.6111	67.0000	33.9167	29.3889	**23.5833**	

$$\begin{aligned} T &= \{(5-1)(4042812.5-(5\times 18^2/4)(5\times 18+1)^2)\} \\ &\quad \div \{(5\times 18(5\times 18+1)(2\times 5\times 18+1))/5 \\ &\quad -(1/5)\times 940397\} \\ &= 25.4250. \end{aligned} \tag{21}$$

Given five algorithms and 18 datasets, T is distributed according to the chi-square distribution with $5-1=4$ degrees of freedom. The p-values for a $\chi^2(4)$ distribution are 0.00000711 (one-tailed) and 0.00001421 (two-tailed). These are far less than 0.05, so the null hypothesis is comfortably rejected. As a result, we can conclude that there is a significant difference among the algorithm results.

4.3 Parameter Tuning

In semi-supervised clustering, the number of constraints has a very important influence on the accuracy of the results. Thus, we conducted a parameter tuning experiment in which different numbers of constraints were used to verify the proposed algorithm. The corresponding accuracies are shown in Fig. 2. The x-coordinate in the figure denotes the number of pairwise constraints and ECs, and the y-coordinate measures the average accuracy with the corresponding constraints. We can see that the accuracies gradually increase with the number of constraints. Although the results intersect at some points, the proposed ECs EM algorithm generally obtains better results than the constrained EM algo-

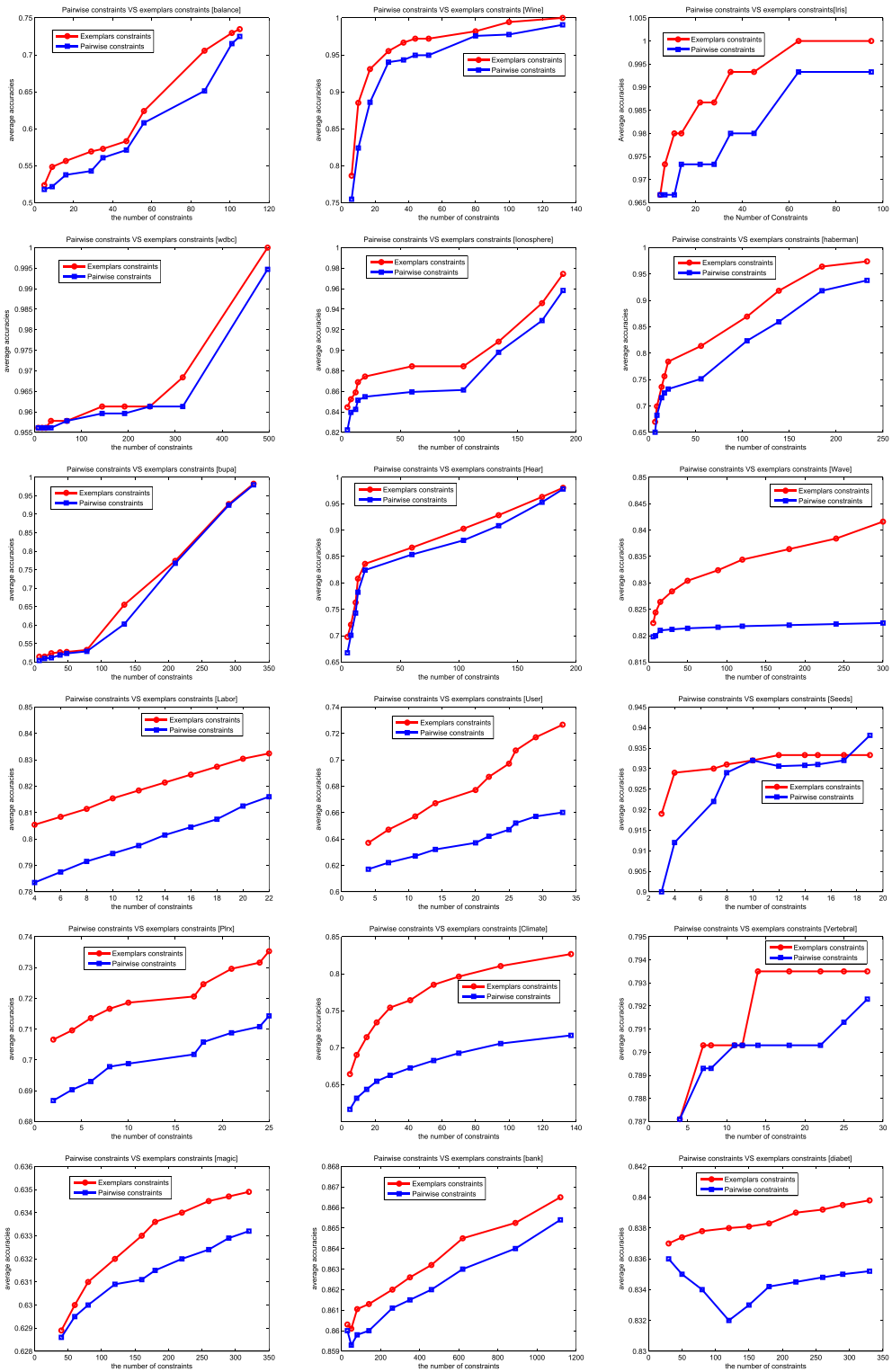


Fig. 2 Experimental results with respect to the number of constraints. This experiment considered constrained EM (blue line) and ECs EM (red line). On most of the datasets, ECs EM outperformed pairwise constraints EM.

rithm.

4.4 Exemplars Constraints Propagation Results

ECs can be propagated by the neighborhood of the exemplars, and this propagation can influence the results. Hence, we designed an experiment to examine how the radius of the ECs neighborhood influences the clustering accuracy. In this experiment, the same constraints were applied and the radius of the neighborhoods was gradually increased. In

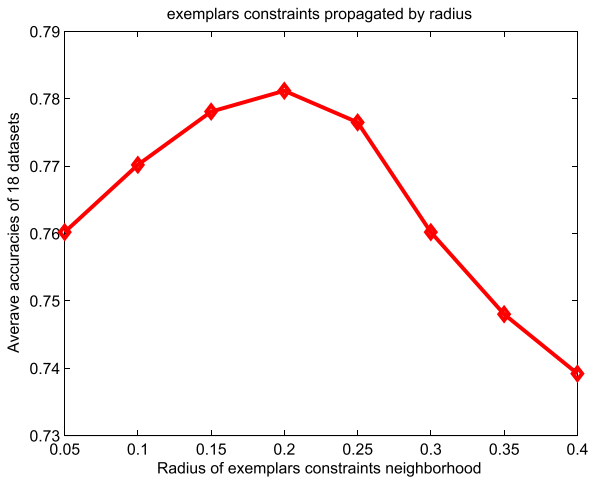


Fig. 3 Experimental results with respect to the radius of the ECs neighborhood.

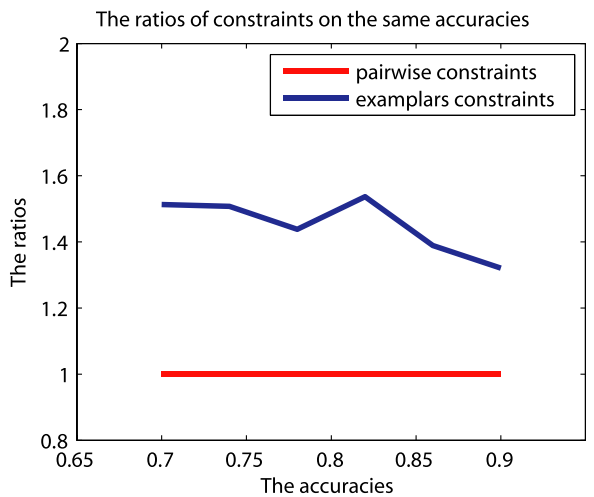


Fig. 4 Ratios of pairwise constraints to ECs with the same accuracies. In this experiment, the different numbers of constraints needed to obtain the same accuracy were recorded. We can see that, if we use pairwise constraints and ECs to obtain the same accuracy, the ratio of pairwise constraints to ECs is 1.4509:1.

Fig. 3, the x-coordinate denotes the radius of the ECs, which is defined as $radius = \frac{r}{MAX(R)}$, where R is the distance matrix of data points. The results using all datasets have been averaged to give the trend, and it is clear that the average clustering results initially improve as the radius increases. Moreover, the accuracy reaches a maximum at a radius of 0.2. As the radius continues to increase, the accuracy declines dramatically. This is because the different neighborhoods may intersect when the radius is large, and this will produce some conflicting constraints, which affect the clustering performance.

4.5 Exemplars Constraints vs. Pairwise Constraints Results

In this experiment, we examined how many pairwise constraints and ECs were required to obtain the same degree of accuracy on each dataset. For comparison, the number of constraints from each algorithm on different datasets was summed as num_e (ECs) and num_p (pairwise constraints). We found that $\frac{num_p}{num_e} = 1.4509$, as shown in Fig. 4, which indicates that the proposed method required fewer constraints to obtain the same accuracy.

5. Conclusions

This paper has made two main contributions. First, we identified the interesting phenomenon that semi-supervised clustering based on pairwise constraints can obtain worse results than the corresponding unsupervised algorithms. Although it is difficult to comprehend this phenomenon, it truly exists in the field of semi-supervised learning. Furthermore, we defined the concepts of *ambiguousness* and *coherence* to illustrate why semi-supervised clustering based on pairwise constraints can achieve worse results. The reason we identified gave us a hint as to which kinds of pairwise constraints can improve the performance of semi-supervised learning. Second, following on from the above, ECs were proposed to address the phenomenon. Specifically, a semi-supervised clustering framework based on ECs was proposed, and an ECs mixture model based on this framework was designed. Expectation-maximization was used to infer the proposed model, and the corresponding algorithm was described. Finally, experimental results on several UCI datasets demonstrate the effectiveness of our proposed method, which outperforms the corresponding semi-supervised algorithms.

The most challenging aspect of our work was to identify under what conditions pairwise constraints can guarantee improved performance. This has been somewhat, but not totally, solved in this paper. Our method represents an active way to ensure that semi-supervised learning gives improved performance. In future work we will gain additional insights. Moreover, if the data points do not belong to Gaussian distribution, but they follow an exponential family of distributions, the exponential mixture model based on exemplars can be proposed to solve the problem in the future.

Compliance with Ethical Standards

Funding: This study was funded by the National Science Foundation of China (grant numbers 61262058, 61175047, 61170111).

Conflict of Interest: All authors (Hongjun Wang, Yinghui Zhang, Tianrui Li, and Yan Yang) declare that they have no conflicts of interest.

Ethical approval: This article does not use any studies with human participants or animals.

References

[1] S. Basu, I. Davidson, and K.L. Wagstaff, "Constrained clustering," CRC Press, 2008.
[2] O. Chapelle, A. Zien, and B. Scholkopf, Semi-supervised learning, MIT Press, 2006.

[3] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, "Constrained k-means clustering with background knowledge," ICML, pp.577–584, 2001.
[4] D. Klein, S.D. Kamvar, and C.D. Manning, "From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering," ICML, pp.307–314, 2002.
[5] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall, "Computing gaussian mixture models with em using equivalence constraints," NIPS, pp.1–8, 2003.
[6] S. Basu, M. Bilenko, and R.J. Mooney, "A probabilistic framework for semi-supervised clustering," KDD, pp.59–68, 2004.
[7] M. Bilenko, S. Basu, and R.J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," ICML, pp.81–88, 2004.
[8] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, "Semi-supervised graph clustering: a kernel approach," ICML, pp.457–464, 2005.
[9] B. Yan and C. Domeniconi, "An adaptive kernel method for semi-supervised clustering," in Machine Learning: ECML 2006, vol.4212, pp.521–532, Springer, 2006.
[10] D.-Y. Yeung and H. Chang, "A kernel approach for semi-supervised metric learning," IEEE Trans. Neural Netw., vol.18, no.1, pp.141–149, 2007.
[11] S.C.H. Hoi, R. Jin, and M.R. Lyu, "Learning nonparametric kernel matrices from pairwise constraints," ICML, pp.361–368, 2007.
[12] M. Okabe and S. Yamada, "Learning similarity matrix from constraints of relational neighbors," Journal of Advanced Computational Intelligence and Intelligent Informatics, vol.14, no.4, pp.402–407, 2010.
[13] X. Yin, S. Chen, E. Hu, and D. Zhang, "Semi-supervised clustering with metric learning: An adaptive kernel method," Pattern Recognition, vol.43, no.4, pp.1320–1333, 2010.
[14] T. Li, C. Ding, and M.I. Jordan, "Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization," ICDM, pp.577–582, 2007.
[15] W. Tang, H. Xiong, S. Zhong, and J. Wu, "Enhancing semi-supervised clustering: A feature projection perspective," KDD, pp.707–716, 2007.
[16] D. Zhang, S. Chen, Z.H. Zhou, and Q. Yang, "Constraint projections for ensemble learning," AAAI, pp.758–763, 2008.
[17] J.-H. Sublemontier, L. Martin, G. Cleuziou, and M. Exbrayat, "Integrating pairwise constraints into clustering algorithms: optimization-based approaches," Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on, pp.272–279, IEEE, 2011.
[18] H. Zeng and Y.-M. Cheung, "Semi-supervised maximum margin clustering with pairwise constraints," IEEE Trans. Knowl. Data Eng., vol.24, no.5, pp.926–939, 2012.
[19] J. Yu, D. Tao, Y. Rui, and J. Cheng, "Pairwise constraints based multiview features fusion for scene classification," Pattern Recognition, vol.46, no.2, pp.483–496, 2013.
[20] H. Jiang, Z. Ren, J. Xuan, and X. Wu, "Extracting elite pairwise constraints for clustering," Neurocomputing, vol.99, pp.124–133, 2013.
[21] X. Wang, B. Qian, and I. Davidson, "Labels vs pairwise constraints: A unified view of label propagation and constrained spectral clustering," ICDM, pp.1146–1151, 2012.
[22] T.F. Covoes, E.R. Hruschka, and J. Ghosh, "Competitive learning with pairwise constraints," IEEE Trans. Neural Netw. Learning Syst, vol.24, no.1, pp.164–169, 2013.
[23] E.R. Eaton, Clustering with propagated constraints, Thesis of the University of Maryland, 2005.
[24] J. Huang and H. Sun, "Lightly-supervised clustering using pairwise constraint propagation," 2008 3rd International Conference on Intelligent System and Knowledge Engineering, pp.765–770, 2008.
[25] Q. Xu, M. desJardins, and K.L. Wagstaff, "Active constrained clustering by examining spectral eigenvectors," Discovery Science, vol.3735, pp.294–307, Springer, 2005.
[26] S. Basu, A. Banerjee, and R.J. Mooney, "Active semi-supervision for pairwise constrained clustering," SDM, pp.333–344, SIAM, 2004.
[27] D.P. Kingma, S. Mohamed, D.J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," Advances in Neural Information Processing Systems, pp.3581–3589, 2014.
[28] I. Davidson, K.L. Wagstaff, and S. Basu, "Measuring constraint-set utility for partitional clustering algorithms," Knowledge Discovery in Databases: PKDD 2006, vol.4213, pp.115–126, 2006.
[29] M. Mezard, "Where are the exemplars?," Science, vol.315, no.5814, pp.949–951, 2007.
[30] M. Khosla, K. Melhorn, and K. Panagiotou, Message Passing Algorithms, Ph.D. Thesis, Citeseer, 2009.
[31] B.J. Frey and D. Dueck, "Clustering by passing messages between data points," science, vol.315, no.5814, pp.972–976, 2007.
[32] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," Science, vol.344, no.6191, pp.1492–1496, 2014.
[33] Z. Li, J. Liu, and X. Tang, "Pairwise constraint propagation by semidefinite programming for semi-supervised classification," ICML, pp.576–583, 2008.
[34] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," J. Machine Learning Research, vol.3, pp.583–617, 2003.
[35] R.M. Neal and G.E. Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants," in Learning in graphical models, pp.355–368, Springer, 1998.
[36] M. Bruneau and E. Magnin, "Convergence of a stochastic approximation version of the em algorithm," J. American Oil Chemists Society, vol.55, no.4, pp.375–380, 1999.
[37] C. Wu, C. Yang, H. Zhao, and J. Zhu, "On the convergence of the em algorithm: From the statistical perspective," arXiv preprint arXiv:1611.00519, 2016.
[38] Z.H. Zhou and W. Tang, "Knowledge-based systems," ICML, pp.77–83, 2006.
[39] S. Garcla, A. Fernndez, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," Information Sciences, vol.180, no.10, pp.2044–2064, 2010.

Sailan Wang received her master's degree from department of design Wuhan University 2006. From 2012 until today she is studying for a doctorate at Sichuan University. Her field of expertise is the study of tourism data. Now she is a associate professor of Department of Computer Science, JinCheng Institute of SiChuan University, China since 2008.

Zhenzhi Yang received his Ph.D. in the history and culture of Chinese minority nationality and regional economic development from Sichuan Normal University. Now, he is a professor of tourism school of Sichuan University, also one of academic committee of the most authoritative academic journals 'tourism tribune'.

Jin Yang received his M.S. degree and the Ph.D. degree in computer science from Sichuan University, Sichuan, China. He is an Professor in Department of Computer Science at LeShan normal university. His main research interests include network security, artificial immune, knowledge discovery and expert systems.

Hongjun Wang graduated from Sichuan University in June 2009, majoring in Computer Science and Technology. He studied one year in University of Minnesota,USA. Now he is working in Information Science and Technology School of Southwest Jiaotong University, Sichuan Province. He is interested in machine learning, semi-supervised learning, ensemble learning and semi-supervised ensemble learning.