Computing and Informatics, Vol. 31, 2012, 1557–1574

MST-BASED SEMI-SUPERVISED CLUSTERING USING M-LABELED OBJECTS

Xiaoyun Chen, Mengmeng Huo,Yangyang Liu

School of Information Science and Engineering
Lanzhou University, China
e-mail: hemm09@lzu.edu.cn

Communicated by Ján Paralič

Abstract. Most of the existing semi-supervised clustering algorithms depend on pairwise constraints, and they usually use lots of priori knowledge to improve their accuracies. In this paper, we use another semi-supervised method called label propagation to help detect clusters. We propose two new semi-supervised algorithms named K-SSMST and M-SSMST. Both of them aim to discover clusters of diverse density and arbitrary shape. Based on Minimum Spanning Tree's algorithm variant, K-SSMST can automatically find natural clusters in a dataset by using K labeled data objects where K is the number of clusters. M-SSMST can detect new clusters with insufficient semi-supervised information. Our algorithms have been tested on various artificial and UCI datasets. The results demonstrate that the algorithm's accuracy is better than other supervised and semi-supervised approaches.

Keywords: Data mining, semi-supervised learning, clustering, label propagation, MST

Mathematics Subject Classification 2010: 62H30, 91C20

1 INTRODUCTION

Semi-supervised learning has been becoming an important topic in the field of pattern recognition and machine learning recently. Semi-supervised clustering can significantly improve the quality of unsupervised clustering by using some weak form of supervision [1], such as partially labeled and pairwise constraints. The general idea is to cluster the unlabeled samples with the useful information given by labeled

objects. Clustering, also called unsupervised learning, is a process that groups a set of physical or abstract objects into classes of similar objects. Objects in the same cluster are similar to one another while in different clusters they are dissimilar. Labeled objects are used in clustering algorithms to help determine which group each object should belongs to. Such a clustering process based on user feedback or guidance constraints is called semi-supervised clustering.

Clustering has a long and rich history in a variety of scientific fields [2]: Taxonomists, social scientists, psychologists, biologists, statisticians, mathematicians, engineers, computer scientists, medical researchers, and others who collect and process real data all have contributed to clustering methodology. In general, the major clustering algorithms include partitioning methods, hierarchical methods, density-based methods, grid-based methods, model-based methods, clustering high-dimensional data, constraint-based clustering, etc., and their various combinations and improvements, etc. Well-known clustering algorithms include K-MEANS (partitioning) [3], K-MEDOIDS (partitioning) [4], BIRCH (hierarchical) [5], ROCK (hierarchical) [6], CHAMELEON (hierarchical) [7], DBSCAN (density-based) [8], DENCLUE (density-based) [9], STING (grid-based) [10], EM (model-based) [11], SOM (model-based) [12], CLIQUE (clustering high-dimensional data) [13]. They provide good results in detecting certain cluster structures.

Most of the clustering methods need setting more than one parameter. By adjusting the parameters, the clustering results will change accordingly. The quality of clustering depends on the parameters setting. Some clustering algorithms can only detect certain cluster structures. When it comes to arbitrary shape clusters, bad clustering results are probably generated. They will have trouble in some of the challenging datasets shown in Figures 1 a) and 1 b). Another difficult task is clustering non-uniform density clusters. In Figures 1 c) and 1 d), two examples of multi-density clusters are presented. Traditional methods cannot provide good quality clustering in this case.

Compared with supervised learning methods, semi-supervised clustering algorithms still require too much training data to keep the accuracy of the results, though they need less labeled data. Furthermore, as it is difficult to identify the boundary of the cluster, detecting new clusters is also a challenge for semi-supervised clustering algorithms.

In this paper, we propose two new semi-supervised clustering algorithms, MST-based Semi-Supervised clustering using K-labeled objects (K-SSMST) and MST-based Semi-Supervised clustering using M-labeled objects (M-SSMST). K-SSMST algorithm allows to expand clusters only using K labeled objects as the labeled dataset, where K is the number of the clusters; that means there should be one labeled object from each cluster. Contrary to most of the existing clustering algorithms, K-SSMST does not need any parameter when expanding clusters. M-SSMST algorithm is able to detect new clusters when the number of the labeled objects is M, where M is less than K. It only needs one parameter to detect boundary of clusters which can help find new clusters and noise. Both of our two algorithms can automatically discover clusters of different densities and arbitrary shapes. And experiments

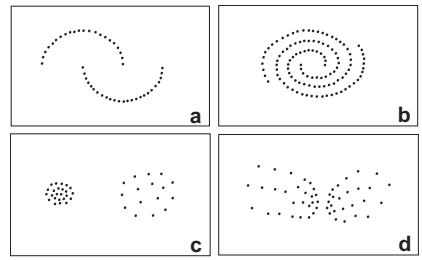


Fig. 1. a) 2-Half ring dataset, b) 2-Spiral dataset, c)–d) Multi-density dataset

in Section 4 use several datasets including UCI and manual datasets to verify the accuracy of algorithms.

The rest of this paper is organized as follows: Section 2 provides a background discussion of semi-supervised clustering algorithms. In Section 3 and 4, our main algorithms are presented. Section 5 then shows and discusses the experimental results. In the final section, we present our conclusions and future work.

2 RELATED WORK

Semi-supervised learning, combining advantages of supervised and unsupervised learning, aims to increase the ability of mining the structure of unlabeled data. Since it generally achieves high accuracy with less number of labeled objects, it has received significant amount of attention in recent studies on pattern recognition, machine learning and data mining. Generally, the most common method for semi-supervised clustering can be classified into two categories. One is pairwise constraints which use the concepts must-link constraints and not-link constraints (Figure 2 b)). The other is label propagation [18]. This method assigns labels to parts of the objects, and expands unlabeled dataset using labeled objects (Figure 2 c)).

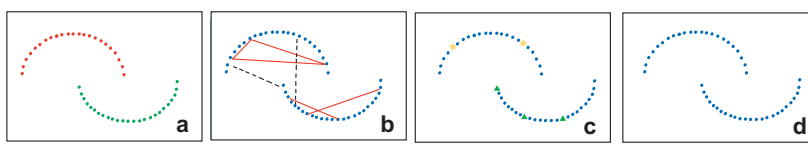


Fig. 2. a) Supervised, b) Partially constrained: the must-link and cannot-link constrained is denoted by red solid and gray dashed lines, c) Partially labeled, d) Unsupervised

One of the most common semi-supervised methods is pairwise constraints. Wagstaff et al. [14] defined the concepts of the two basic kinds of pairwise constraints that made the insertion of domain knowledge into the clustering process possible: the must-link and cannot-link constraints [15]. On one hand, must-link constraints specify that two points connected by the constraint must belong to the same cluster.

On the other hand, cannot-link constraints specify that two instances cannot belong to the same cluster. Objects are assigned to clusters following these constrains. The constraints are generally given by the domain expert. More methods [20, 21, 22, 23] to semi-supervised clustering alter the objective function of existing clustering algorithms to incorporate the pairwise constrains [2]. The most common methods change and improve on the usage of constraints for the K-MEANS algorithm, e.g. MPC-K-MEANS [19].

Label propagation algorithms choose little data as labeled dataset, and then the knowledge supplied by those labeled data will affect the clustering process. A small number of algorithms are proposed. Böhm and Plant presented HISSCLU in [16], a hierarchical density-based clustering algorithm based on OPTICS. Lelis and Sander presented SSDBSCAN in [15]. The algorithm can automatically find the natural cluster structure when the densities among cluster vary widely in a dataset.

For the existing semi-supervised methods, although compared with supervised learning methods the size of labeled dataset becomes much smaller, it still requires a lot of labeled objects to keep accuracy of the final result. Reducing the number of labeled objects is one of the main focuses of our methods.

3 SEMI-SUPERVISED CLUSTERING BASED ON MST USING K-LABELED OBJECTS

In this section we describe K-SSMST, which is based on Minimum Spanning Tree's algorithm variant.

3.1 K-MST

Minimum Spanning Tree, also abbreviated as "MST", is an important concept in data structure. MST is used frequently to solve the problems in practical applications, e.g. building a communication network between n cities. There are many methods to structure MST. Two of the commonest algorithms for solving the MST problem are Kruskal's algorithm and Prim's algorithm. In view of our algorithm, we mirror Prim's algorithm to expand clusters. Description of Prim's algorithm is as follows.

Suppose $G = (V, E)$, where V is the set of all points in dataset D, E is the set of potential interconnections between pairs of points, and for each edge $(p, q) \in E$, the weight $w(p, q)$ specifying the distance between p and q. In Prim's Algorithm, the edge added to MST is always a least-weight edge connecting the tree to a vertex not in the tree. The algorithm randomly chooses a point from V, and adds it into an empty set U; then repeats the following step: finding a least-weight edge (u, v) where u belongs to U and v belongs to $V - U$. Adding the point v to U, meanwhile, adding the edge (u, v) to E, until $U = V$.

K-MST is a new concept which is proposed to base on Prim's algorithm. Prim's algorithm constructs MST starting with one point, and the result MST is a single

tree. In our paper we use K points to expand cluster, which means the number of U is K. During each expand step, we firstly consider the minimum edge (u_i, v_i) from U_i $(i = 1, 2, \ldots, K)$ where $u_i \in U_i$ and $v_i \in V - \cup_{i=1}^{K} U_i$, then choose the shortest edge (u_p, v_p) of K edges $(1 \leq p \leq K)$, add it to E, and add v_p to U_p. Repeat the expand step until $\cup_{i=1}^{K} U_i = V$. The expand result is K trees. The right part of Figure 3 shows the different results structured by Prim's algorithm and K-MST algorithm.

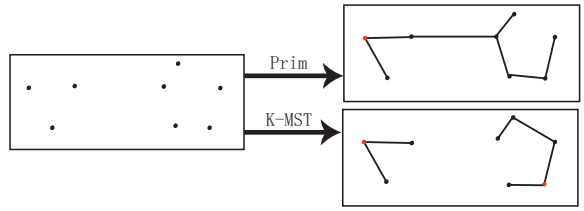


Fig. 3. Prim's algorithm and K-MST

3.2 K-SSMST Algorithm

Dataset D in K-SSMST will be separated into two subsets: supervised subset D_L and unsupervised subset D_U. There is only one labeled object from each cluster in D_L.

$$D_L = \{p | p \in C_i (i = 1, 2, \ldots, K)\} \tag{1}$$

where p represents the labeled objects chosen from cluster C_i and K is the number of clusters in D. In order to better measure the accuracy of our algorithm in different situations, we choose labeled objects from D randomly. D_U includes the rest of objects in D.

The next step assigns the objects in D_U using the information given by D_L. The K point in D_L is deemed as the initial expanding point of each subtree. During execution of the assigning process, algorithm will build a set of disjoint subtrees $S_T = \{T_1, T_2, \ldots, T_K\}$. Each object is treated as a vertex in the subtrees, and each of the subtree T_i is treated as a cluster. During iteration process, we use K-MST to choose the shortest edge as follows:

(i) Choose the shortest edge SC_i of any edge connecting to a vertex in the subtree T_i.

$$SC_i(p, q) = \min\{Dist(p, q) | p \in C_i, q \in D_U (i = 1, 2, \ldots, K)\} \tag{2}$$

(ii) Choose the shortest edge SE of all classes.

$$SE(p, q) = \min\{SC_i(p, q) | i = 1, 2, \ldots, K\} \tag{3}$$

Then we set the label of point q as label of point p, and add $SE(p,q)$ to U_i, add the q to the cluster C_i. Repeat steps (i) and (ii) until all objects in D_U are labeled. The clustering process is considered complete, and the algorithm has produced the final K clusters. The K-SSMST algorithm is explained in detail in Algorithm 1.

Algorithm 1 K-SSMST(D_U,D_L)

```
1. while D_U ≠ ∅
2.     for all p ∈ C_i, q ∈ D_U
3.         compute SC_i(p,q)
4.     end for
5.     compute SE(p,q)
6.     q.label = p.label
7.     D_L = D_L ∪ {q}
8.     D_U = D_U − {q}
9. end while
```

4 SEMI-SUPERVISED CLUSTERING BASED ON MST USING M-LABELED OBJECTS

In the prior section, we introduced the K-SSMST algorithm. To implement the algorithm, we need K labeled objects to supervise the clustering process. As a matter of fact, it is not easy to get one object from each cluster of the dataset. If one or more clusters are not given any labeled object, all points in these clusters cannot assign to the right clusters. To address this issue, we propose an algorithm which can find these clusters using M ($M < K$) labeled objects. We use four concepts to help detect new clusters.

Definition 1 (Average Distance of Intra-Cluster, avgC_i)**.** Average Distance of Intra-Cluster is the average distance of all edges of subtree T_i. The initial value of avgC_i is 0. Average distances of clusters differ from each other. The distances are constantly changed. The current average distance of each cluster can be computed as follows:

$$\text{avg}C_i = \frac{\text{avg}_{old} * (LN - 1) + SE(p,q)}{LN}, \tag{4}$$

where avgC_i denotes the average distance after adding a new edge, avg$_{old}$ denotes the average distance before added edge, LN denotes the number of labeled objects, and $SE(p,q)$ is the length of current shortest edge.

Definition 2 (Extension Threshold ϵ)**.** Extension Threshold ϵ is used to decide whether a cluster should be extended. If the current shortest edge $SE(p,q) > \epsilon$, cluster C_i will stop expanding, and this algorithm will find new clusters.

$$\epsilon = \delta * \text{avg}C_i \tag{5}$$

The parameters δ of the dataset differ from one another.

If no cluster needs to expand and there still remain unlabeled objects, the algorithm will start to find new clusters. In order to avoid choosing outliers, we use influence function to find a relative density object.

Definition 3 (Influence Function)**.** The influence function can be an arbitrary function that can be determined by the distance between two objects in a neighborhood [1]. In order to create fast algorithm, we choose the square wave influence function which is the simplest influence function.

Then we will describe another new concept named extension threshold δ. This concept is used to decide whether it is time to stop expanding, and the algorithm should find new clusters.

$$f(x,y) = \begin{cases} 0, & \text{if Dist}(x,y) > \sigma \\ 1, & \text{otherwise.} \end{cases} \tag{6}$$

We employ maximum average distance of existing clusters to be the threshold σ:

$$\sigma = \max\{\text{avg}C_i | i = 1, 2, \ldots, M\}. \tag{7}$$

Definition 4 (Field Function)**.** Based on the influence function, we define the field function if an object o_i in a dataset as

$$F(o_i) = \sum_{j=1, j\neq i}^{|D_U|} f(o_i, o_j) \quad (o_i, o_j \in D_U, i = 1, 2, \ldots, |D_U|). \tag{8}$$

If the field function value of an object is zero and all clusters have stopped to expand, this object is considered to be an outlier.

Figure 4 shows a simple sample of field function. We choose three typical objects. Set the value of σ. It is easy to compute the field function value of three objects: $F(a) = 6$, $F(b) = 2$, $andF(c) = 0$. From the data of the result, we can infer that object a is located at a density field, and c is far away from the remaining set of data. This is obvious from the figure that our inference is right.

According to the previous notions, our second algorithm M-SSMST is proposed. The algorithm is designed to discover new clusters when there are not enough labeled objects.

In M-SSMST algorithm dataset D will also be divided into D_L and D_U. The size of D_L is M $(M < K)$. This algorithm is described as follows:

1. Use K-MST to find the shortest edge $SE(p, q)$ of each cluster.
2. If $SE(p, q)$ is larger than extension threshold ϵ, cluster C_i will end its expand process. Otherwise, object q will be added in D_L and cluster C_i. Meanwhile, q will be deleted from D_U. Then update the average distance of cluster C_i.
3. Repeat step 1. and 2. until no cluster can expand.

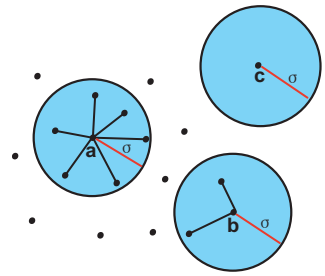


Fig. 4. Field function

4. If D_U is null, end the clustering process. Otherwise choose one object which has the max field function value. If the max value is zero, the object will be treated as an outlier; then end the clustering process, otherwise add it into D_L.
5. Repeat steps 1. and 2. until the new cluster ends its expand process.
6. Repeat steps 4. and 5.

MST-based Semi-supervised clustering using M-labeled objects (M-SSMST) is introduced in Algorithm 2.

Algorithm 2 M-SSMST(D_U, D_L)

```
1. while D_U ≠ ∅
2.     while cluster can expand
3.         for all p ∈ C_i, q ∈ D_U
4.             compute SC_i(p, q)
5.         end for
6.         compute SE(p, q)
7.         if SE(p, q) > ε
8.             q.label = p.label
9.             D_L = D_L ∪ {q}
10.            D_U = D_U − {q}
11.            compute avgC_i
12.        else
13.            canExpandC_i = false
14.        end if
13.    end while
14.    expandNewCluster(D_U, D_L)
15.end while
```

A sample is given for the reader to understand our algorithm more easly (cf. Figure 5). The sample dataset has 13 objects in total. Extension threshold parameter δ is set to 2. There are two labeled objects $\{V_2, V_5\}$ (Figure 5 a)). The two objects

Algorithm 3 expandNewCluster(D_U, D_L)

1. $p = \max F(o_i)$
2. $D_L = D_L \cup \{p\}$
3. **for** all $p \in C_i$, $q \in D_U$
4. **compute** $SC_i(p, q)$
5. **end for**
6. **compute** $SE(p, q)$
7. **if** $SE(p, q) > \epsilon$
8. q.label = p.label
9. $D_L = D_L \cup \{q\}$
10. $D_U = D_U - \{q\}$
11. **compute** avgC_i
12. **else**
13. canExpandC_i = false
14. **end if**

belong to cluster C_1 and C_2, and their labels are 1 and 2, respectively. We use these two objects to supervise the following clustering process.

Firstly, we find the minimum edges connected with objects of each existing cluster; they are $E(V_2, V_3) = 4$ and $E(V_5, V_4) = 1$ (Figure 5 b)). The shortest edge is $E(V_5, V_4)$ clearly, so we add object V_4 to the cluster C_2 (Figure 5 c)). avgC_2 is initialized as 1, $\delta * \text{avg}C_2 = 2$. Repeat the finding step; the shortest edge connected with objects of cluster C_1 and C_2 is $E(V_5, V_6) = 1$ (Figure 5 d)). Because $E(V_5, V_6) < \delta * \text{avg}C_2, V_6$ is added to C_2. For the same reason V_7 is also assigned to the second cluster (Figure 5 e)). Now the avgC_2 is changed to 4/3. In the next edge we have found is $E(V_4, V_3) = 3$, but V_3 cannot be added to C_2, because $E(V_4, V_3) > \delta * \text{avg}C_2 = 8/3$. The cluster C_2 stops expanding.

Cluster C_1 iterates the finding and the assigning work until the distance of new finding edge is larger than $\delta * \text{avg}C_1$ (Figure 5 g)), then cluster C_1 also stops the expand process. Now the existing clusters cannot expand anymore. Since there are still some objects that are not assigned to any cluster, the algorithm prepares to find new clusters.

M-SSMST starts a new cluster with an object which has the highest field function value, then extends the cluster as in the prior steps. In this example, the average distance edges of existing clusters avgC_1 and avgC_2 are 5 and 4/3, respectively. So we pick the larger one (5) as the field function threshold σ. Values of all unlabeled objects are listed in Table 1. Obviously V_{10} has the highest value, so we choose it as the first member of Cluster C_3.

After the cluster C_3 stops to expand (Figure 5 j)), there is only one unlabeled object left, namely V_{13}. The value of σ is still 5, so field function value of V_{13} remains 0. According to our rules, it is an outlier.

When all the objects are labeled, M-SSMST halts. Final clustering result which has three clusters and one outlier is shown in Figure 5 k).

Object	Field Function Value
V_8	3
V_9	2
V_{10}	4
V_{11}	3
V_{12}	1
V_{13}	0

Table 1. Field function value of unlabeled objects

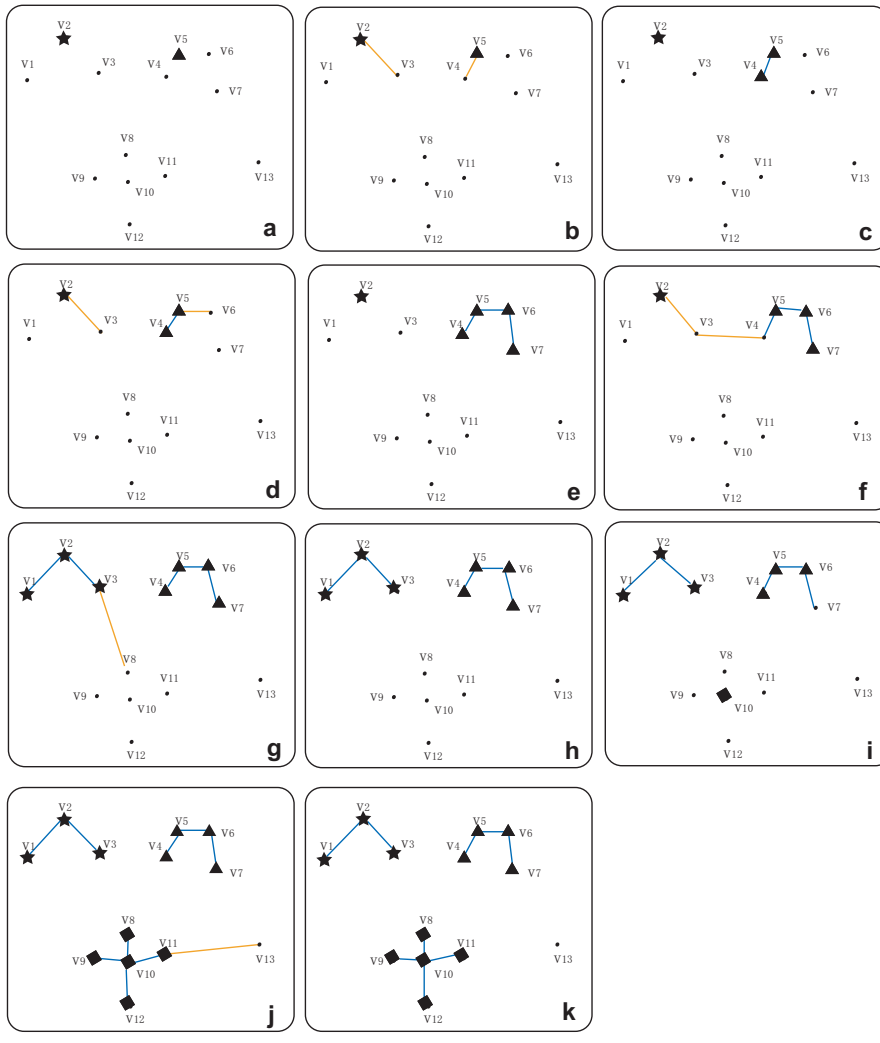


Fig. 5. M-SSMST sample

5 EXPERIMENTAL VALIDATION

5.1 Datasets

To test our algorithms, we use several datasets for experiments. The test datasets include UCI datasets and several manual datasets.

Experiments with benchmark datasets are the best way to verify the effectiveness of the algorithms. We can compute the accuracy of the algorithms easily by comparing the generated clusters with the real class labels. Therefore we conducted experiments on the following datasets: Iris, Wine, Glass, and Ecoli. They are obtained from University of California Irvine Machine Learning Repository [24]. Iris, which is one of the best known datasets in the data mining literature, is a real dataset having 4 attributes, 150 objects and 3 classes. Wine dataset is a multivariate, real dataset, with 13 attributes, 178 objects and 3 classes. Glass identification dataset is a multivariate, real dataset with 9 attributes, 214 objects, 6 classes. Ecoli is also a multivariate dataset, with 7 real type attributes, 336 objects and 8 classes.

The 2-half rings, 2-spiral, MD1 and MD2 datasets [25] are shown in Figure 1 a)–d). These datasets are synthetically generated; they contain 49, 113, 40 and 92 objects, respectively. Although they only have 2 dimensions, finding the natural clusters of these datasets is a challenge for many clustering algorithms.

	K	N	Attribute
Iris	3	150	4
Wine	3	178	13
Glass	6	214	9
Ecoli	8	336	7
2-HalfRing	2	49	2
2-Spiral	2	113	2
MD1	2	40	2
MD2	2	92	2

Table 2. Test datasets

5.2 Experimental Evaluation and Analysis

In this section, we present several experiments to verify our algorithms. We use accuracy to be the evaluation method. Comparing the resulting and original labels, the correct object sharing the proportion of the whole dataset is called accuracy. Obviously, the higher the accuracy, the better the clustering result is.

Sometimes in a dataset, data of certain rows are too much larger than the other data in other rows. If these data are not very important, it is clear that they have negative influence on the clustering. In order to avoid this, we use min-max normalization to convert the data of each row to range $(0, 1)$.

5.2.1 Evaluation of K-SSMST

In the first group of experiments, K-SSMST is compared with KNN and SSDBSCAN. KNN is a well known supervised classification algorithm which decides the label of the object depending on its K neighbors.

We ran K-SSMST on all test datasets 10 times, chose the worst and the best and computed average results. The labeled objects of K-SSMST, KNN and SSDBSCAN were randomly chosen. Two groups of labeled objects were used to be as labeled datasets: K objects and 20 per cent of whole datasets.

Input	K-SSMST			KNN		SSDBSCAN	
	MIN	MAX	AVG	LN = K	LN = 0.2N	LN = K	LN = 0.2N
Iris	100	100	100	33.3	90.8	74.1	78.1
Wine	100	100	100	39.4	94.4	57.1	83.9
Glass	93.9	100	98.3	35.8	56.4	36.5	56.4
Ecoli	98.8	100	99.1	43.3	82.5	54.6	70.6
2-HalfRing	100	100	100	51.1	97.4	46.8	89.7
2-Spiral	100	100	100	50.5	100	50.4	52.3
MD1	100	100	100	60.5	87.5	60.5	84.4
MD2	100	100	100	51.1	97.3	54.4	90.0

Table 3. Cluster result on datasets

Table 3 shows the result of K-SSMST, KNN and SSDBSCAN for the datasets listed in Table 2. It can be observed that K-SSMST performs better than KNN and SSDBSCAN. K-SSMST algorithm showed great advantages when three algorithms chose K labeled objects at the same time. Even when KNN and SSDBSCAN use 20 percent objects of datasets, the accuracies were not higher than K-SSMST.

5.2.2 Evaluation of M-SSMST

Algorithm M-SSMST is proposed to find new clusters. The following experiments were conducted to test whether M-SSMST had this ability.

	2.6	2.7	2.8	2.9	3.0	3.1	3.2	3.3	3.4	3.5
MIN	31.8	35.1	35.1	95.3	97.2	98.0	99.3	99.3	98.6	66.2
MAX	99.3	99.3	99.3	99.3	99.3	100	100	100	100	100
AVG	90.3	91.9	91.8	98.0	98.8	99.2	99.4	99.6	99.8	89.6

Table 4. M-SSMST running on Iris

The Iris dataset has 3 classes as we know. We randomly chose 2 classes. Then we randomly chose one object in each of the chosen classes. That means we had 2 objects in labeled dataset. The remaining objects were unlabeled. M-SSMST ran 10 times on each extension threshold parameter δ. Table 4 shows the minimum, maximum and average results. When the value of δ was between 2.9 and 3.4, the

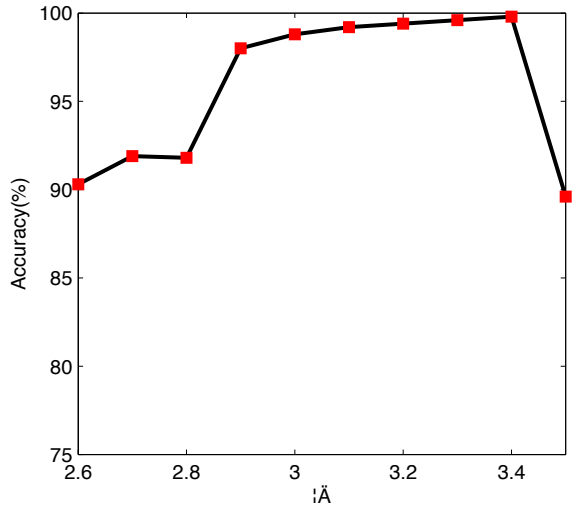


Fig. 6. Result of M-SSMST on Iris

	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2	2.3	2.4
MIN	48.9	79.5	82.4	69.9	70.5	66.5	59.1	59.7	59.7	59.7
MAX	76.7	88.6	95.5	97.2	99.4	100	99.4	99.4	70.5	70.5
AVG	68.6	82.7	92.1	92.0	89.1	78.4	79.7	74.7	65.2	65.3

Table 5. M-SSMST running on Wine

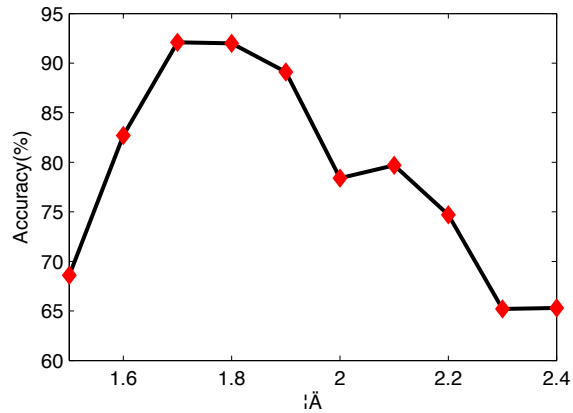


Fig. 7. Result of M-SSMST on Wine

	2.5	2.6	2.7	2.8	2.9	3.0	3.1	3.2	3.3	3.4
MIN	38.7	56.5	62.1	52.8	53.3	45.8	44.4	44.4	14.6	9.4
MAX	91.0	92.5	91.6	94.4	84.1	93.9	88.8	94.4	96.7	96.2
AVG	78.4	79.6	81.4	74.2	72.1	74.5	71.6	69.5	73.5	64.2

Table 6. M-SSMST running on Glass

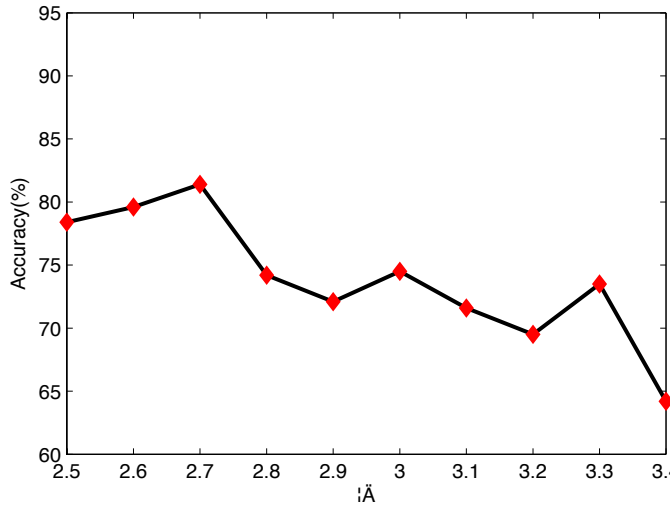


Fig. 8. Result of M-SSMST on Glass

accuracy was higher than 98 percent. Figure 6 shows the average results on Iris visually.

Wine also has 3 classes. We randomly chose 2 classes. Also randomly chose one object in each chosen classes. So we had 2 objects in labeled dataset. The rest objects added to unlabeled dataset. M-SSMST ran 10 times on each extension threshold parameter δ. Table 5 shows the minimum, maximum and average results. When the value of δ was between 1.6 and 2.1, the accuracy was higher than 75 percent. The average results on Wine are shown in Figure 7.

Experiments on benchmark datasets Glass also ran 10 times. We randomly chose 2 of 6 classes, and randomly chose one object in each of the chosen classes. Then the

	1.8	1.9	2.0	2.1	2.2	2.3	2.4	2.5	2.6	2.7
MIN	56.3	70.4	77.2	82.0	89.2	88.6	52.4	52.4	42.8	19.5
MAX	77.2	83.5	88.3	90.1	93.7	93.7	94.6	94.6	94.3	95.2
AVG	72.3	79.4	83.6	86.5	86.8	86.8	89.5	87.8	83.7	69.9

Table 7. M-SSMST running on Ecoli

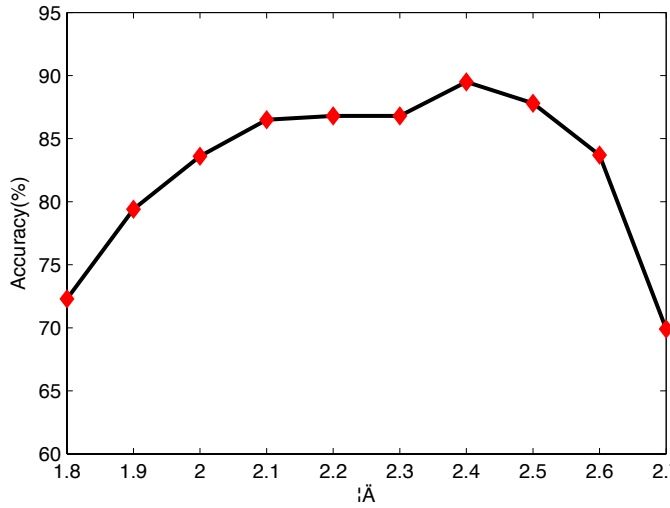


Fig. 9. Result of M-SSMST on Ecoli

2 objects were added into labeled datasets. Table 6 shows the minimum, maximum and average results. The accuracy depended on the chosen labeled objects. When picking the right objects, M-SSMST could work well; but once the labeled dataset was chosen inappropriately, the result was unsatisfactory. Figure 8 shows the average results on Glass.

The Ecoli dataset has 8 classes. We randomly chose 2 classes, and randomly chose one object in each of the chosen classes. In other words, we had 2 objects as the training set, the rest was testing sets. M-SSMST ran 10 times on each extension threshold δ. Table 7 shows the clustering results. When the value of δ was between 2.0 and 2.6, the accuracy was higher than 80 percent. Figure 9 shows the average results on Ecoli.

6 CONCLUSION

In this paper, we presented two algorithms. The first algorithm is a new semi-supervised clustering algorithm named K-SSMST. By using the minimum spanning tree, the algorithm uses K labeled objects to expand the clusters. When label information is not sufficient, we propose another method, namely M-SSMST, which could automatically discover new clusters.

As future work, we plan to solve the problem when expanding from sparse region to the density region and when finding new clusters. Besides, we will improve the time complexity of the algorithms.

REFERENCES

[1] Han, J.—Kamber, M.: Data Mining Concepts and Techniques. 2nd ed. China Machine Press.

[2] Jain, A. K.: Data Clustering: 50 Years Beyond K-Means. ECML/PKDD (1), Lecture Notes in Computer Science, Springer2008, Vol. 5211, pp. 3–4.

[3] MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability 1967, pp. 281–297.

[4] Kaufman, L.—Rousseeuw, P. J.: Finding Groups in Data: An Introduction to Cluster Analysis. 2nd ed. Wiley-Interscience, New York 2005.

[5] Zhang, T.—Ramakrishnan, R.—Livny, M.: BIRCH: An Efficient Data Clustering Method for Very Large Databases. Proceedings of the 1996 ACMSIGMOD International Conference on Management of Data 1996, pp. 103–114.

[6] Guha, S.—Rastogi, R.—Shim, K.: Rock: A Robust Clustering Algorithm for Categorical Attributes. IEEE Comput. Vol. 32, 1999, No. 8.

[7] Karypis, G.—Han, E.-H.—Kumar, V.: Chameleon: Hierarchical Clustering Using Dynamic Modeling. Addison-Wesley, Harlow, England 1999.

[8] Ester, M.—Kriegel, H. P.—Sander, J.—Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining 1996, pp. 226–231.

[9] Hinneburg, A.—Keim, D. A.: An Efficient Approach to Clustering in Large Multimedia Databases with Noise. Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining 1998, pp. 58–65.

[10] Wang, W.—Yang, J.—Muntz, R.: STING: A Statistical Information Grid Approach to Spatial Data Mining. In Proceedings of the 23rd VLDB Conference, Athens, Greece 1997, pp. 186–195.

[11] Dempster, A. P.—Laird, N. M.—Rubin, D. B.: Maximum Likelihood from Incomplete Data via the EMAlgorithm. Journal of the Royal Statistical Society, Series B (Methdological), Vol. 39, 1977, pp. 1–38.

[12] Kohonen T.—Somervno, P.: How to Make Large Self-Organizing Maps for Non-Vectorial Data. Neural Networks 2002, pp. 945–952.

[13] Agrawal, R.—Gehrke, J.—Gunopulos, D.—Raghavan, P.: Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, New York 1998, ACM Press 1998, pp. 94–105.

[14] Wagstaff, K.—Cardie, C.—Rogers, S.—Schroedl, S.: Constrained K-Means Clustering with Background Knowledge. In Proc. 18th International Conf. on Machine Learning 2001, pp. 577–584, 2001.

[15] Lelis, L.—Sander, J.—Menasalvas, E.: Semi-Supervised Density-Based Clustering. 3rd ed. ICDM 2009.

[16] Böhm, C.—Plant, C.: HISSCLU: A Hierarchical Density Based Method for Semi-Supervised Clustering. In Proc. 11th Int. Conf. on Extending Database Technology, Nantes, France 2008.

[17] Ankerst, A.—Breunig, M.—Kriegel, H.-P.—Sander, J.: OPTICS: Ordering Points to Identify the Clustering Structure. In ACM SIGMOD International Conference on the Management of Data, Philadelphia, PA, USA 1999.

[18] Zhu, X.—Ghahramani, Z.: Learning from Labeled and Unlabeled Data with Label Propagation. Technical report 2002.

[19] Mimarouglu, S.—Erdil, E.: Combining Multiple Clusterings Using Similarity Graph. Pattern Recognition 2011, pp. 694–703.

[20] Bar-Hillel, A.—Hertz, T.—Shental, N.—Weinshall, D.: Learning Distance Functions Using Equivalence Relations. In: Proc. 20th Internat. Conf. on Machine Learning 2003, pp. 11–18.

[21] Basu, S.—Bilenko, M.—Mooney, R. J.: A Probabilistic Framework for Semisupervised Clustering. Proc. 10th KDD 2004, pp. 59–68.

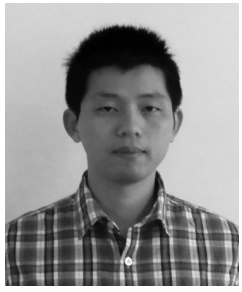
[22] Chapelle, O.—Schölkopf, B.—Zien, A.: Semi-Supervosed lLearning. MIT Press, Cambridge, MA. 2006.

[23] Lu, Z.—Leen, K. T.: Penalized Probabilistic Clustering. Neural Comput., Vol. 19, 2007, No. 6, pp. 1528–1567.

[24] Asuncion, A.—Newman, D.: UCI Machine Learning Repository. `www.ics.uci.edu/mlearn/epository.html`, 2007.

Xiaoyun Chen is Professor and Doctoral Supervisor at School of Information Science and Engineering, Lanzhou University. She is currently the Head of Computer Software and Theory Lab and a member of China Computer Federation Technical Committee on Databases. Her research interests can be summarized as developing effective data analysis techniques for novel data intensive applications. She is currently interested in various techniques of data warehousing, data mining, database systems and social networks.

Mengmeng Huo graduated from School of Information Science and Engineering, Lanzhou University. In 2012, she received her Master's degree in computer software and theory. Her research interests include data mining, cluster analysis and semi-supervised learning.



Yangyang Liu graduated from School of Information Science and Engineering, Lanzhou University. In 2012, he received his Master's degree in computer software and theory. His research focuses on data mining, cluster analysis and semi-supervised learning.