HSC: A SPECTRAL CLUSTERING ALGORITHM COMBINED WITH HIERARCHICAL METHOD

Li Liu[*], Xiwei Chen[*], Dashi Luo[*], Yonggang Lu[*], Guandong Xu[†], Ming Liu[‡]

Abstract: Most of the traditional clustering algorithms are poor for clustering more complex structures other than the convex spherical sample space. In the past few years, several spectral clustering algorithms were proposed to cluster arbitrarily shaped data in various real applications. However, spectral clustering relies on the dataset where each cluster is approximately well separated to a certain extent. In the case that the cluster has an obvious inflection point within a non-convex space, the spectral clustering algorithm would mistakenly recognize one cluster to be different clusters. In this paper, we propose a novel spectral clustering algorithm called HSC combined with hierarchical method, which obviates the disadvantage of the spectral clustering by not using the misleading information of the noisy neighboring data points. The simple clustering procedure is applied to eliminate the misleading information, and thus the HSC algorithm could cluster both convex shaped data and arbitrarily shaped data more efficiently and accurately. The experiments on both synthetic data sets and real data sets show that HSC outperforms other popular clustering algorithms. Furthermore, we observed that HSC can also be used for the estimation of the number of clusters.

Key words: *Data mining, clustering, spectral clustering, hierarchical clustering*

Received: May 9, 2013
Revised and accepted: October 24, 2013

1. Introduction

Clustering is a powerful tool to analysis data by assigning a set of observations into clusters so that the points in a cluster have high similarity and points in

[*]Li Liu, Xiwei Chen, Dashi Luo, Yonggang Lu
School of Information Science and Engineering, Lanzhou University, Gansu 730000, P.R.China
{liliu,chenxw2011,luodsh12,ylu}@lzu.edu.cn
[†]Guandong Xu
Advanced Analytics Institute, University of Technology Sydney, NSW 2008, Australia
Guandong.Xu@uts.edu.au
[‡]Ming Liu
School of Electrical and Information Engineering, The University of Sydney, NSW 2006, Australia
ming.liu@sydney.edu.au

different clusters have low similarity. As a typical unsupervised learning method, since there is no prior knowledge about the data set, it also acts as an important data processing and analysis tool. Many clustering applications can be found in these fields, such as web mining, biological data analysis, social network analysis [1], etc. However, clustering is still an attractive and challenging problem. It is hard for any clustering method to give a reasonable performance for every scenario without restriction on the distribution of the dataset.

Traditional clustering algorithms, such as k-means [2], GM_EM [3], etc, while simple, most of them are based on convex spherical sample space, and their ability for dealing with complex cluster structure is poor. When the sample space is not convex, these algorithms may be trapped in a local optimum [4]. The spectral clustering algorithm has been proposed to solve this issue [5]. Spectral clustering algorithm is based on spectra graph theory that partition data using eigenvectors of an affinity matrix derived from the data. It can cluster arbitrarily shaped data [6]. In recent years, spectral clustering has been successfully applied to a large number of challenging clustering applications. It is simple to implement, can be solved efficiently by standard linear algebra software, and often outperforms traditional clustering algorithms such as the k-means algorithm [7].

Due to many advantages of the spectral clustering, it has extensive applications in many fields. It has been successfully applied to image retrieval [8] and mining social networks [9]. Bach and Jordan [10] incorporate the prior knowledge of speech to produce parameterized similarity matrix that can improve the efficiency of clustering. Zhang presents a margin-based perspective on multiway spectral clustering [11]. Jiang [12] has proposed a core-tag oriented spectral clustering method to find out semantic correlation of tags on web 2.0 application. Carlos and Suykens [13] have used a weighted kernel principal component analysis (KPCA) approach based on least-square support vector machine (LS-SVM) to form new formulation for multiway spectral clustering, and the experimental results show that this formulation has improved performance in difficult toy problem and image segmentation. Li [14] pointed out that when calculating the similarity matrix, considering the weights of different attributes could improve the spectral clustering algorithms. D. Correa [15] has introduced a new method for estimating the local neighborhood and scale of data points to improve the robustness of spectral clustering algorithms. Ekin et al. [16] has proposed an initialization-independent spectral clustering that uses K-Harmonic Means (KHM) instead of k-means, and has applied the method to facial image recognition.

Although spectral clustering algorithms have shown good results in various applications, it relies on the dataset where each cluster is approximately well separated to a certain extent. The spectral clustering algorithm will fail to recognize one cluster as different clusters when the cluster has an obvious inflection point within a non-convex space. The reason is that the constructed affinity matrix, which the spectral clustering heavily relies on, will be corrupted with poor pairwise affinity values from the area of inflection points. Especially for most of the recent spectral clustering algorithms that use the traditional central grouping techniques to cluster the affinity matrix, e.g., k-means, it will amplify the misguidance of clustering because these centralized algorithms that are based on a radius distance between two data points cannot separate clusters that are very long or nonlinearly

separable. It will make the algorithm easy to fall into local optimal solutions [17]. For instance, in Fig. 1, traditional spectral clustering will separate the datasets into three clusters by the misleading information induced from the ringlike cluster. Spectral methods cannot guarantee reasonable performances while such misleading information diffuses across different clusters.

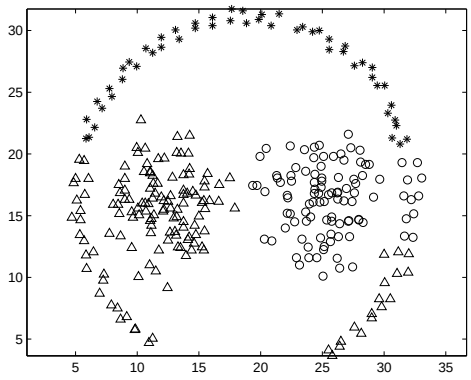


Fig. 1 *The traditional spectral clustering separates the ringlike shape into three different clusters and mistakenly recognizes other data points belong to one of them.*

In this paper, we present a clustering algorithm called HSC (Hierarchical based Spectral Clustering) that combines the spectral clustering with hierarchical method to cluster dataset in convex space, non-convex space or mixture of both while avoiding the local optimum trap. The hierarchical clustering algorithm constructs a tree by scanning sorted points in an incremental order through the whole dataset instead of the neighboring points that can avoid the connections of two different clusters near the same data points. Since the spectral clustering method easily fails in the situation that the datasets are generated from not well-separated clusters, the hierarchical method avoids the disadvantage of the spectral clustering by not considering the relation of neighborhood and handles the noisy neighboring data points effectively. In this study, the hierarchical clustering is applied to cluster the normalized affinity matrix processed by spectral methods. We use a number of lower-dimensional synthetic datasets to show that the simple hierarchical clustering procedure can eliminate the misleading information from different kinds of datasets so the obtained spectral method could cluster the dataset more accurately. Two famous UCI's [18] real datasets both of which are in higher-dimension are also used to evaluate the proposed algorithms. All of these datasets cover extensive clusters of different shapes, densities and sizes with noise and artifacts. Furthermore, since the number of clusters are unknown in most of practical clustering applications, our empirical study found that HSC is the only method, compared with other six well known clustering algorithms, that can find the optimal number of clusters by iteratively searching the integer space with the best measurement of the Adjusted Rand Index.

The rest of this paper is organized as follows. Section 2 provides the background knowledge of the spectral method and the hierarchical clustering algorithm. Section 3 presents the HSC clustering algorithm in detail. Section 4 shows the experimental results that HSC outperforms others six popular algorithms on both five artificial datasets and two real datasets overall. Section 5 discuss the estimation of the number of clusters. Finally, section 6 concludes this paper.

2. Background

2.1 Basic Concept of Spectral Clustering

Given a set of n data points $x_1, x_2, \ldots, x_n$ with each $x_i \in R^d$, we define an affinity graph $G = (V, E)$ as an undirected graph in which the i^{th} vertex corresponds to the data point x_i. For each edge $(i, j) \in E$, we associate a weight a_{ij} that encodes the similarity of the data points x_i and x_j. We refer to the matrix $A = (a_{ij})_{i,j=1}^{n}$ of affinities as the similarity matrix.

Symbol	Meaning
n	the number of data points in the dataset
x_i	the i^{th} data points of the dataset
R^d	an d-dimensional vector space over the field of the real numbers
G	an undirected graph
V	the nodes of the graph
E	the edges of the graph
a_{ij}	the weight of the data points x_i and x_j
A	the similarity matrix
$\|\|x_i - x_j\|\|$ or $dist(x_i, x_j)$	the distance between point x_i and point x_j
D	the degree matrix
L	the Laplacian matrix
α	the number of eigenvectors
t_i	the i^{th} largest eigenvectors of the L
$t_{i,j}$	the j^{th} element of the i^{th} largest eigenvectors
T	the feature vector space
k	the number of clusters
C_i	the set of data points belonging to the i^{th} cluster

Tab. I *Notations.*

In the similarity matrix $A = (a_{ij}) \in R^{n \times n}$, the weight of each pair of vertices x_i and x_j is measured by a_{ij},

$$a_{ij} = \begin{cases} \exp(\frac{-||x_i - x_j||^2}{2\sigma^2}), i \neq j \\ 0, i{=}j \end{cases}, \; i, j = 1, 2, \ldots, n,$$

and it satisfies $a_{ij} \geq 0; a_{ij}{=}a_{ji}$. Where $||x_i - x_j||^2$ can be Euclidean distance, City Block distance, Minkowski distance, or Mahalanobis distance and so on. The degree

of vertex x_i is the sum of all the vertex weights adjacent to x_i, which can be defined as $D_{ii} = \sum_{j=1}^{n} a_{ij}, i = 1, 2, \ldots, n$. A diagonal matrix $D = \begin{bmatrix} D_{11} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & D_{nn} \end{bmatrix}$ can be obtained using the degree of vertices. The matrix $L = D - A$ is called Laplacian matrix. The most commonly used Laplacian matrixes are summarized in Tab. II. In order to simplify the calculation the unnormalized graph Laplacian matrix is used.

Unnormalized	$L = D - A$
Symmetric	$L_{Sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$
Asymmetric	$L_{As} = D^{-1} L = I - D^{-1} W$

Tab. II *Laplacian matrixes types.*

Spectral clustering can be interpreted by several different theories, such as figure cut set theory, random migration point and the perturbation theory [19]. But no matter what theory is used, spectral clustering can been converted to the eigenvector problem of Laplacian matrix, and then the eigenvectors are clustered.

The goal of spectral clustering is to partition the data $\{x_i\}_{i=1}^{n}, x_i \in R^d$ into k disjoint classes $\{C_1, C_2, \ldots, C_k\}$, such that each x_i belongs to one and only one class, which means

$$\begin{cases} C_1 \cup C_2 \cup \ldots \cup C_k = \{x_i\}_{i=1}^{n}, x_i \in R^d \\ C_i \cap C_j = \emptyset, i, j = 1, 2, \ldots, k, i \neq j. \end{cases}$$

Different spectral clustering algorithms formalize this partitioning problem in different ways [5], [20], [21], [22]. In this paper, the following spectral clustering algorithm is used (Algorithm 1):

2.2 Hierarchical clustering algorithm

Hierarchical clustering algorithm organizes the data into different groups at different levels, and forms a respective tree of clustering. It can be further categorized into agglomerative (bottom-up) method and divisive (top-down) method. The agglomerate algorithms treat data points or data set partitions as sub-clusters in the beginning, and then merge the sub-clusters iteratively until a stop condition is met; Divisive methods begin with a single cluster which contains all the data points, and then partition the clusters based on the dissimilarity recursively until some stop condition is reached [23]. In this paper, we use the agglomerate methods until a specific number k of clusters calculated by spectral algorithm is reached.

In the hierarchical clustering procedure, to determine whether to merge two clusters into a new one, the distance between the two clusters is defined, $||C_i - C_j|| = \min\{||x_a - x_b|| : x_a \in C_i, x_b \in C_j\}$. The distance matrix $dist\_mat_{num \times num}$ denotes the distances between every pair of clusters, where num indicates the number of clusters in the current stage.

Algorithm 1: The Spectral Clustering Combined with k-means.

Input:

$\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ - the dataset of data points.
α - the number of eigenvectors ;
k - the number of clusters

Output:

$\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ - the k clusters

1. Construct the similarity matrix A;
2. Calculate the diagonal degree matrix D;
3. Compute the Laplacian matrix: L=D-A;
4. Calculate α largest eigenvectors of L and construct feature vector space $T = (t_1, t_2, \ldots, t_\alpha) \in R^{n \times \alpha}$;
5. Normalize the row vectors of T;
6. Using the k-means algorithm to cluster the normalized row vectors into k clusters.
7. The original data x_i is grouped into the j^{th} cluster if and only if the i^{th} row vector of T is assigned to the j^{th} cluster in Step 6. Output the clustering results $\{C_1, C_2, \ldots, C_k\}$.

$dist\_mat_{num \times num} =$

$$\begin{bmatrix} \infty & ||C_1 - C_2|| & \cdots & \cdots & ||C_1 - C_{\text{num}}|| \\ ||C_2 - C_1|| & \infty & \cdots & \cdots & ||C_2 - C_{num}|| \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \infty & ||C_{num-1} - C_{num}|| \\ ||C_{num} - C_1|| & ||C_{num} - C_2|| & \cdots & ||C_{num} - C_{num-1}|| & \infty \end{bmatrix}$$

Initially, the number of clusters num in this stage is set to n. Each data point is dispatched to a different cluster, namely $x_i \in C_i$ for each $i = 1, 2, \cdots, n$. Then, calculate the distance matrix $dist\_mat_{num \times num}$.

The Single-Linkage method [24] is used to find the two most similar clusters and they are merged as a single cluster. The number of clusters is decreased to $num = num - 1$. Update the distance matrix $dist\_mat_{num \times num}$. Repeat this step until the number of clusters num reaches k. The clusters $\{C_1, C_2, \ldots, C_k\}$ is the final results.

Fig. 2 show a simple illustrative example of the hierarchical clustering algorithm. Nine candidate data points are intended to be clustered into 2 groups, i.e. $P1$, $P2$, $P3$, $P4$, $P5$, $P6$, $P7$, $P8$ and $P9$ located at (1.0, 1.0), (2.0, 1.0), (2.0, 3.0), (3.0, 2.0), (3.0, 4.0), (6.0, 1.0), (6.0, 2.0), (7.0, 1.0) and (7.0, 2.0) respectively. They are grouped into 9 initial clusters, $C_1 = \{P1\}$, $C_2 = \{P2\}$, ..., $C_9 = \{P9\}$. The distance matrix $dist\_mat_{9 \times 9}$ is calculated. Herein the minimal value is the distance between C_1 and C_2. The two clusters C_1 and C_2 are grouped into one cluster. The total number of clusters is then decreased to 8, $C_1 = \{P1, P2\}$, $C_2 = \{P3\}$, ..., $C_8 = \{P9\}$. A new distance matrix $dist\_mat_{8 \times 8}$ needs to be updated. And then the two clusters with the minimal value in this distance matrix, $C_5 = \{P6\}$ and

Algorithm 2: The Hierarchical Clustering Algorithm

Input:

$\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ – the dataset of data points.

k – the number of clusters

Output:

$\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ – the k clusters

1: $num = n$; //Initialize n clusters
2: **FOR EACH** $x_i \in \mathcal{X}$
3: $C_i = \{x_i\}$; //Each point belongs to a different cluster
4: **END FOR**
4: $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$
5: **WHILE** (num>k) **DO**
6: Calculate the distance matrix $dist\_mat_{num \times num}$;
7: //Find the two clusters with the minimal distance in the distance matrix
8: $(C_i, C_j) = \{(C_i, C_j) : min\{dist\_mat[i, j] : 1 \leq i, j \leq num\}\}$;
9: //Merge the two clusters
10: $C_i = C_i \cup C_j$;
11: $\mathcal{C} = \mathcal{C} - C_j$;
12: $num = num - 1$;
13: **END WHILE**

$C_6 = \{P7\}$, are selected to be grouped. After this stage, there are 7 clusters, $C_1 = \{P1, P2\}$, $C_2 = \{P3\}$, $C_3 = \{P4\}$, $C_4 = \{P5\}$, $C_5 = \{P6, P7\}$, $C_6 = \{P8\}$, $C_7 = \{P9\}$. Repeat to create the new distance matrix and group the two clusters with the shortest distance into one until only 2 clusters are remained. Finally, all the data points are classified into two clusters: $C_1 = \{P1, P2, P3, P4, P5\}$ and $C_2 = \{P6, P7, P8, P9\}$.

This hierarchical clustering algorithm can discover clusters of arbitrary shapes and sizes, but cannot perform well when clusters are overlapping.

3. Spectral Clustering with Hierarchical Clustering

We combine the advantages of spectral clustering and hierarchical clustering algorithms, and present a novel clustering algorithm called HSC. In the first part of HSC, the spectral clustering algorithm is used to obtain a normalized row vectors. In the second part, the hierarchical clustering algorithm is used to find a set of clusters. The HSC clustering algorithm can identify clusters having non-spherical shapes with different sizes.

The HSC clustering algorithm consists of the following steps:

Step 1. Preprocess the raw dataset and obtain a normalized row vectors:

Create a similarity matrix A from the raw dataset. Calculate the diagonal degree matrix D and the Laplacian matrix $L = D - A$. Find the α largest eigenvector of the L. Construct feature vector space and normalize the row vectors of T,

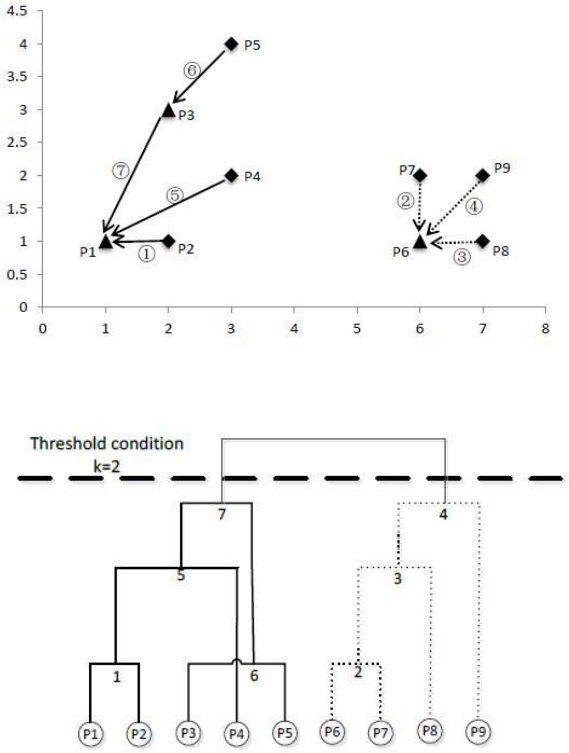


Fig. 2 *Example of the clustering procedure by using the hierarchical clustering algorithm. The number on line indicates the clustering steps. The dotted line and the solid line represent different clustering. The threshold condition $k = 2$ is the final number of clusters.*

$$T = \begin{bmatrix} t_{1,1} & t_{2,1} & \cdots & t_{\alpha,1} \\ t_{1,2} & t_{2,2} & \cdots & t_{\alpha,2} \\ \vdots & \vdots & \ddots & \vdots \\ t_{1,n} & t_{2,n} & \cdots & t_{\alpha,n} \end{bmatrix}.$$

Step 2. Cluster the data points:

(a) For each feature vector $y_i = (t_{1,i}, t_{2,i}, \ldots, t_{\alpha,i})$, where $1 \leq i \leq n$, y_i represents the original data point x_i. y_i is used to be the input data point in α dimension for Algorithm 2, instead of using x_i directly. Hence, the distance between two data point y_i and y_j is then defined as $||y_i - y_j|| = \sqrt{\Sigma_{p=1}^{\alpha}(t_{p,i} - t_{p,j})^2}$. According to this definition, the distance between two clusters can be calculated, and the distance matrix *dist_mat* can be created as well.

(b) Find two clusters with the minimal distance in the distance matrix *dist_mat* and group them into one cluster. Update the clusters and their corresponding distance matrix *dist_mat*. Repeat this step until k clusters are remained.

Algorithm 3: HSC clustering algorithm
Input:
$\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ – the dataset of data points.
α – the number of eigenvectors ;
k – the indicated number of clusters
Output:
$\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ – the k clusters
1. Construct the similarity matrix A from $\mathcal{X}$, and produce the normalized feature vector space $T = (t_1, t_2, \ldots, t_\alpha) \in R^{n \times \alpha}$ by using Algorithm 1.
2. **FOR EACH** $i \in \{1, 2, \ldots, n\}$
3. $y_i = (t_{1,i}, t_{2,i}, \ldots, t_{\alpha,i})$;
4. **END FOR**
5. Find k clusters by using Algorithm 2 of which the input dataset of data points is $\{y_1, y_2, \ldots, y_n\}$.
6. **RETURN** $\{C_1, C_2, \ldots, C_k\}$.

4. Experimental Results And Analysis

In order to evaluate HSC algorithm, we selected a number of datasets that contain points in 2D space, and contain clusters of different shapes, densities, sizes, and noise. Similar data sets can be downloaded from UNIVERSITY OF EASTERN FINLAND (http://cs.joensuu.fi/sipu/datasets/). We compared the results and performances of HSC with other six well known clustering algorithms, k-means, DBSCAN [25], KAP [27], GM_EM [3], HC (Hierarchical Clustering algorithm), SCKM (Spectral Clustering algorithms based on K-Means).

4.1 Datasets

We use five artificial datasets in our experiment, the properties of each data set described as follows: The Path-based data set consists of a circular cluster with an opening near the bottom and two Gaussian distributed clusters inside. Each cluster contains 100 data points. The 3-spiral data set consists of 312 points and these points are divided into 3 clusters. Both the Path-based data set and the 3-spiral data set were used in [28]. The Aggregation dataset consists of seven perceptually distinct groups of points and the total number of these points is 788. In fact, these datasets containing the features that are known to create difficulties for the selected algorithms, e.g., narrow bridges between clusters, uneven-sized clusters, etc, are also used in many previous works [29] as a benchmark.

In addition, two real datasets in higher dimensions called Vehicle Silhouette data set (with 846 points in 18 dimensions) and Balance-scale data set (with 625

points in 4 dimensions) from the UC Irvine Machine Learning Repository [18] are also used in this experiment.

Datasets		Number of clusters	Size
Synthetic datasets	Path-based	3	300
	3-Spiral	3	312
	Jain's toy	2	373
	Circle	2	134
	Aggregation	7	788
Real datasets	Vehicle silhouette	4	846
	Balance-scale	3	625

Tab. III *Datasets.*

4.2 Evaluation criteria

There are usually two types of validation indices for evaluating the clustering results: one for measuring the quality by examining the similarities within and between clusters, and the other for comparing the clustering results against an external benchmark.

As a well-known first type index, the DB-Index [30] is used in our experiments. It is defined as

$$\mathrm{DB-Index} = \frac{1}{K}\sum_{i=1}^{K}\max\frac{\sigma_i+\sigma_j}{dist(C_i,C_j)}, i \neq j,$$

where K is the number of clusters, σ_i is the square root of the intra-cluster inertia of cluster C_i and $dist(C_i, C_j)$ is the distance between the centroids of cluster C_i and C_j.

$$\sigma_i = \sqrt{\sum_{j\in C_i}\frac{dist(j,C_i)^2}{N_i}},$$

where $dist(j, C_i)$ is the distance between data point j and the centroid of cluster C_i, and N_i is the number of data points in cluster C_i. Usually the clustering results with low intra-cluster distances and high inter-cluster distances will produce a low DB-Index. When computing the DB-index, we take the mean position of all the members of a cluster as its centroids instead of using the cluster center given by the algorithm. This will ensure a better comparison because different algorithms usually have different schemes for deciding the centers of clusters. The original cluster centers determined by all the algorithms are shown as filled circles in the figures of the clustering results for reference. If there is only one cluster, a trivial value of zero is given as the DB-Index. On the other extreme, when the number of clusters is close to the number of data points, some clusters will only have one member, and the σ will be zero for these clusters. As a result, a small DB-Index will be produced. So the DB-Index is more meaningful when the number of clusters is greater than one and much smaller than the number of data points.

In order to compare the clustering results with a given benchmark, the Adjusted Rand Index (or simply ARI) [31] is also used in the evaluation. Given a set S of n elements, and two groups of these points, namely $X = \{X_1, X_2, \ldots, X_r\}$and $Y = \{Y_1, Y_2, \ldots, Y_s\}$,the overlap between X and Y can be summarized in a contingency table $[n_{ij}]$ where each entry n_{ij} denotes the number of objects in common between X_i and Y_j : $n_{ij} = |X_i \cap Y_j|$.

$X \setminus Y$	Y_1	Y_2	$\cdots$	Y_s	*Sums*
X_1	n_{11}	n_{12}	$\ldots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\ldots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\ldots$	n_{rs}	a_r
Sums	b_1	b_2	$\cdots$	b_s	

The adjusted form of the Rand Index, ARI, is defined as

$$\begin{aligned} AdjustedIndex &= \frac{Index - ExpectedIndex}{MaxIndex - ExpectedIndex} = \\ &= \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]\Big/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]\Big/\binom{n}{2}}, \end{aligned}$$

where $n_{ij}, a_i, b_j$are values from the contingency table. The ARI can yield a value between -1 and $+1$ [32]. The maximum value of the ARI is 1, which means that the two clustering results are exactly the same. When the two partitions are picked at random which corresponds to the null model, the ARI is 0 [33].

Either a lower DB-Index or a higher ARI indicates a better clustering result. When the benchmark is available, both of the indices are measured, otherwise only the DB-Index is used in our experiments.

4.3 Implementation details

To evaluate the HSC method, it is compared with six popular clustering methods: k-means, DBSCAN, KAP,GM_EM, HC, and SCKM. The HSC algorithm was implemented in Matlab R2010a. The others came from Matlab toolbox, or from the Matlab Central website (http://www.mathworks.com/matlabcentral/). All the experiments were executed on a desktop computer with an Inter(R) Pentium(R) CPU G620 @2.60GHz and 4GB RAM.

The computational time of a single execution, the number of clusters produced, the number of iterations, and the validation indices are recorded in Tab. VI−XI. For all of the clustering, we set the number of clusters in Tab. III, and for the rest parameters, if any, we used Matlab's defaults. Because k-means, GM_EM, and SCKM are all randomized algorithms, they are executed 100 times, and the results with the best ARI were selected. For DBSCAN, KAP, HC, HSC, the results are the same when given the same parameters. However, all of these algorithm were

executed for 100 times. The computational time of each algorithm was estimated to be the average time of these 100 trials. The minimum and the average value of the DB-Index, as well as the maximum and the average value of the Adjusted Rand Index are shown in the following tables.

4.4 Parameter settings

Almost all of the existing clustering algorithms are required to set a number of parameters, which might lead to different outcomes. As such, we conducted an experiment that used various parameter configurations in order to find the parameter settings with the best clustering results for the comparisons in this experiments.

The number of clusters k have to be set for all of the algorithms expect DBSCAN. We assume that k is known in advance(see Tab. III). We will discuss the case that k is unknown in section 5. Therefore, k-means, KAP, GM_EM, HC and FCM have no more parameter to be configured.

The parameter α, the number of eigenvectors, is required for SCKM and HSC. We iteratively searched the parameter α in the integer space ranging from $[0, 30]$. The α with the maximal ARI for each dataset was selected respectively. Tab. IV shows the settings of the parameter α on these two algorithms for each dataset.

	SCKM	HSC
Path-based	5	5
3-Spiral	3	3
Jain's toy	2	2
Circle	2	2
Aggregation	2	7
Vehicle silhouette	19	15
Balance-scale	3	1

Tab. IV *The parameter α selected for SCKM and HSC on different datasets.*

DBSCAN do not need to set the number of clusters explicitly. It requires two parameters to be indicated: Eps – the neighborhood radius and $MinPts$ – the number of objects in a neighborhood of an object. DBSCAN is a density-based clustering methods that the densities between different data sets have different effects on the clustering performance. We determined the parameters Eps and MinPts with the best ARI according to a simple but effective heuristic method proposed by [25].The parameter settings are shown in Tab. V.

4.5 Results and Analysis

4.5.1 Experiments using synthetic data sets

For Dataset Path-based containing three clusters of a circular cluster with an opening near the bottom and two Gaussian distributed clusters inside it, the results are shown in Tab. VI and Fig. 3. It can be seen that k-means, KAP cannot recognize the circular cluster. Although GM_EM and HC are possible to recognize the

Datasets	Eps	MinPts
Path-based	2	9
3-Spiral	3	5
Jain's toy	2.5	3
Circle	0.09	2
Aggregation	1.5	5

Tab. V *The parameters Eps and MinPts selected for DBSCAN on different datasets.*

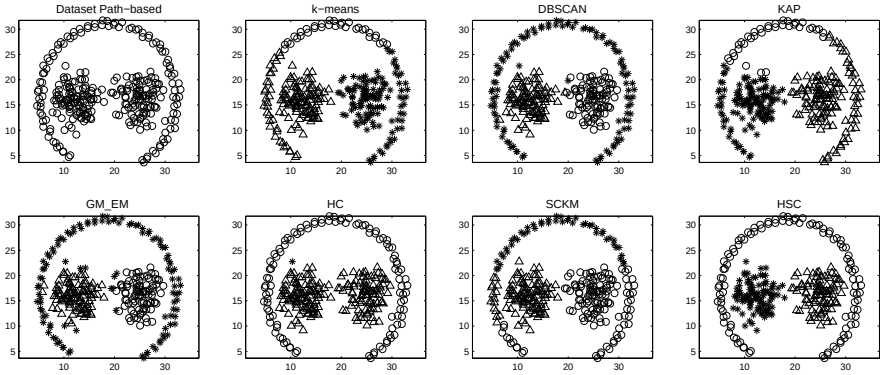


Fig. 3 *Clustering results of Dataset Path-based.*

Dataset	Algorithms	#Clusters[a] (#Iter.)[b]	ARI		DB-Index		Run time(sec)
			MAX.	AVE.	MIN.	AVE.	
	k-means	3(100)	0.4922	0.4920	0.7531	0.7546	0.0028
	DBSCAN	3(1)	0.9598		2.4786		0.0110
	KAP	3(1)	0.4755		0.7882		1.8239
Path-based	GM_EM	3(100)	0.9195	0.8957	2.3876	2.4095	0.5341
	HC	3(1)	0.5875		4.9253		0.0062
	SCKM	3(100)	1	0.8732	1.0623	2.3094	0.0988
	HSC	3(1)	1		2.5443		0.2945

[a] The number of clusters.
[b] The number of iterations.

Tab. VI *Clustering results on Dataset Path-based by different Clustering Algorithms.*

peripheral annular, they could not recognize the two Gaussian distributed clusters which are close to each other. HSC got a prefect clustering result. DBSCAN algorithm got an excellent ARI score and is able to separate the circular from the other two parts inside it. However, there are still few data points in the two Gaussian distributed clusters cannot be recognized. GM_EM and SCKM can obtain good results with a high ARI. However, it is uncertain to obtain good clustering results

for these two randomness-based methods. It is shown that HSC obtained a much better ARI than all the other methods, but its DB-Index is very high. This is because the cluster shape is far beyond globular. The DB-Index is probably not a good qualified criterion for such case.

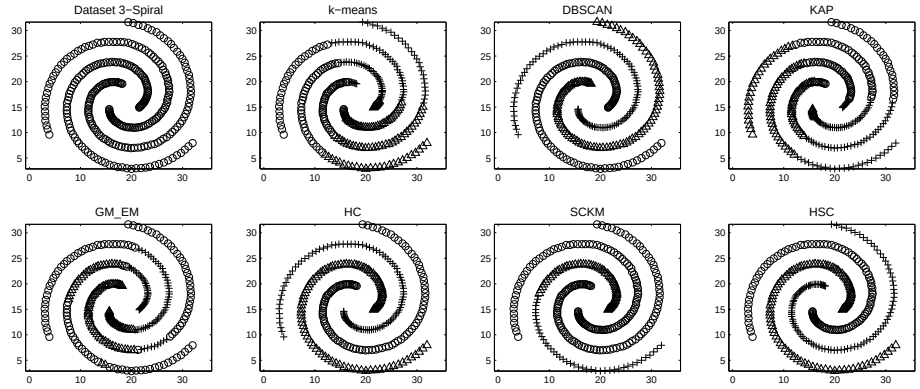


Fig. 4 *Clustering results of Dataset 3-Spiral.*

Dataset	Algorithms	#Clusters (#Iter.)	ARI		DB-Index		Run time(sec)
			MAX.	AVE.	MIN.	AVE.	
3-Spiral	k-means	3(100)	-0.0055	-0.0059	0.9489	0.9546	0.0059
	DBSCAN	3(1)	1		6.1355		0.0149
	KAP	3(1)	-0.0060		0.9527		1.8998
	GM_EM	3(100)	0.0628	0.0541	4.6454	5.0097	1.2336
	HC	3(1)	1		6.1355		0.0087
	SCKM	3(100)	1	0.8382	3.8434	5.7591	0.1138
	HSC	3(1)	1		6.1355		0.3289

Tab. VII *Clustering results on Dataset 3-Spiral by different Clustering Algorithms*

For Dataset 3-Spiral containing three clusters with the same size, Tab. VII and Fig. 4 shows that k-means, KAP, and GM_EM methods obtained poor clustering results. Although the maximum ARI value of SCKM reaches 1, the great difference between the maximum and the minimum ARI from the 100 trials indicates the uncertainty of the methods. DBSCAN, HC and HSC methods obtained excellent results for this dataset.

For Dataset Jain's toy containing two meniscus clusters, the results are shown in Tab. VIII and Fig. 5. We can see that only SCKM and HSC clustering algorithm obtained the best clustering results for the Jain's toy dataset. DBSCAN obtained a relatively high ARI value. However, it failed to separate the upper cluster as two groups. It is sensitive to the density changing in the upper cluster. The ARI values obtained by other algorithms are not very high. They cannot separate the two meniscus clusters. Although the HC method gets the minimum DB-Index value, its clustering accuracy was very poor because of the large distance between each point and the clustering center.

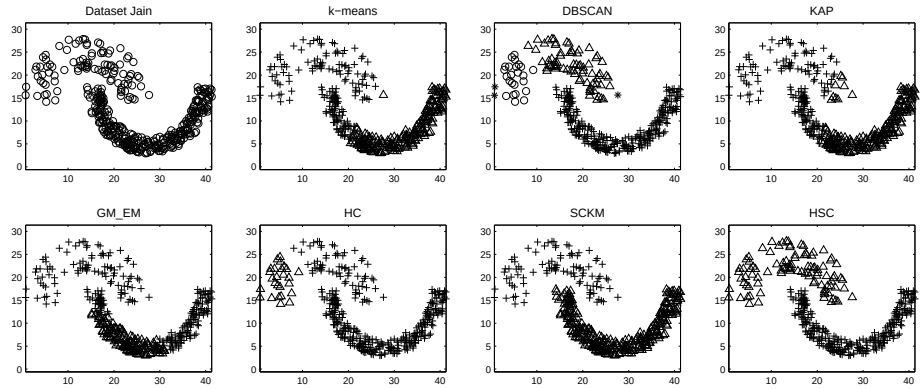


Fig. 5 *Clustering results of Dataset Jain's toy.*

Dataset	**Algorithms**	**#Clusters (#Iter.)**	**ARI**		**DB-Index**		**Run time**(sec)
			MAX.	AVE.	MIN.	AVE.	
Jain's toy	k-means	2(100)	0.3181		0.8583		0.0027
	DBSCAN	3(1)	0.9411		2.1823		0.0163
	KAP	2(1)	0.2488		0.8730		3.1794
	GM_EM	2(100)	0.0838	0.0213	1.4218	1.8968	0.3774
	HC	2(1)	0.2563		0.6419		0.0086
	SCKM	2(100)	1		0.9745		0.1893
	HSC	2(1)	1		0.9745		0.5427

Tab. VIII *Clustering results on Dataset Jain's toy by different Clustering Algorithms.*

For Dataset Circle containing two clusters of the same size, the results are shown in Tab. IX and Fig. 6. DBSCAN, KAP, GM_EM, HC clustering algorithm cannot get the correct clustering results, other methods got excellent clustering results. Regardless of the parameter setting, DBSCAN which is sensitive to the density change failed to separate this dataset into six circular clusters.

From Fig. 7, it can be seen that only the HSC method can identify the variety of shapes in this complex dataset, while all the other methods fail to identify the optimal result. The DBSCAN, GM_EM and HC algorithm could not distinguish the narrow bridges between clusters in the right parts of this dataset. Tab. X shows that other algorithms got relatively high ARI values. The HSC method got the maximum ARI value and the smallest DB-Index.

4.5.2 Experiments using two real data set

We examined all clustering methods except DBSCAM on the real datasets. DBSCAN was not used in this experiment because the selection of the two parameters is relatively difficult in high dimensional datasets. Since the benchmark is not available for these two real datasets, only the DB-Index was used in this study.

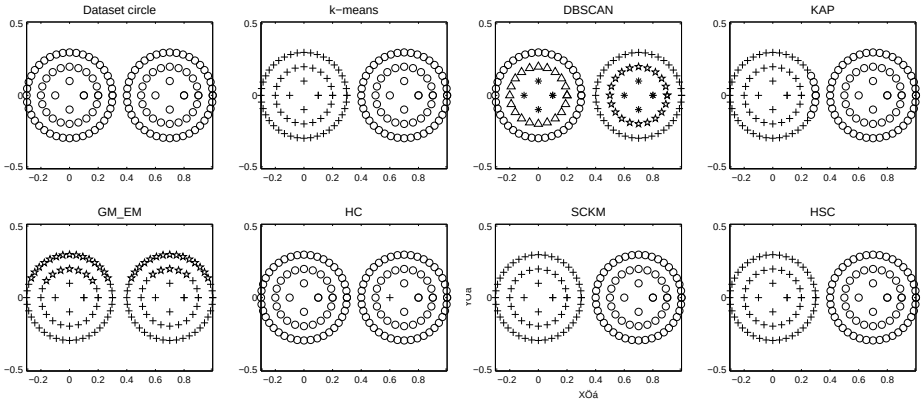


Fig. 6 *Clustering results of Dataset Circle.*

Dataset	Algorithms	#Clusters (#Iter.)	ARI		DB-Index		Run time(sec)
			MAX.	AVE.	MIN.	AVE.	
	k-means	2(100)	1		0.7466		0.0015
	DBSCAN	5(1)	0.4666		181.5419		0.0440
	KAP	2(1)	0.7738		0.7719		0.5827
Circle	GM_EM	2(100)	0.9118	0.5958	0.7546	1.0247	0.0751
	HC	2(1)	0		1.8032		0.0018
	SCKM	2(100)	1		0.7466		0.0163
	HSC	2(1)	1		0.7466		0.0399

Tab. IX *Clustering results on Dataset Circle by different Clustering Algorithms*

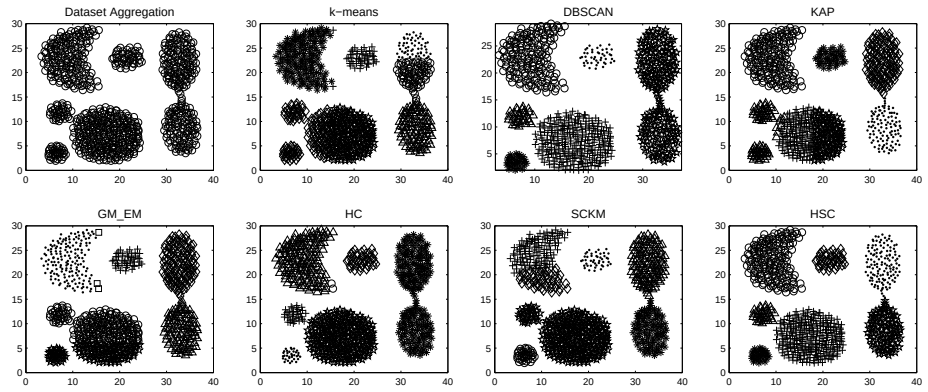


Fig. 7 *Clustering results of Dataset Aggregation .*

Dataset	**Algorithms**	**#Clusters (#Iter.)**	**ARI**		**DB-Index**		**Run time**(sec)
			MAX.	AVE.	MIN.	AVE.	
Aggregation	k-means	7(100)	0.7782	0.7262	0.7411	0.7991	0.0075
	DBSCAN	7(1)	0.8824		0.6247		0.0440
	KAP	7(1)	0.7763		0.7760		32.6706
	GM_EM	7(100)	0.9840	0.8159	0.5822	1.2308	15.1347
	HC	7(1)	0.8795		0.6351		0.0451
	SCKM	7(100)	0.9971	0.8284	0.5414	0.8014	2.8497
	HSC	7(1)	1		0.5372		7.5160

Tab. X *Clustering results on Dataset Aggregation by different Clustering Algorithms.*

Tab. XI shows that the DB-Index of the HSC results are the smallest compared with all other methods. It indicates that HSC got better clustering performance for the two real datasets.

Dataset	**Evaluation parameters**		**Algorithms**					
			k-means	KAP	GM_EM	HC	SCKM	HSC
vehicle	k **(#Iter.)**		4(100)	4(1)	4(100)	4(1)	4(100)	4(1)
	DB-Index	MIN.	1.1589	1.8643	1.4523	0.9324	2.7400	0.6438
		AVE.	1.4959		1.9912		2.9029	
	Run time(sec)		0.0186	22.3412	0.3724	0.0953	2.7370	7.6009
balance	k **(#Iter.)**[a]		2(100)	2(1)	2(100)	2(1)	2(100)	2(1)
	DB-Index	MIN.	1.7480	N/A[a]	1.7818	0.8546	3.1167	0.7552
		AVE.	1.7769		1.8520		3.2337	
	Run time(sec)		0.0136	63.3676	0.0862	0.2409	0.7481	2.3452

[a] K-AP algorithm is not applicable for this dataset.

Tab. XI *Clustering results on Real Datasets by different clustering algorithms.*

4.5.3 Computational time

The experimental results in Tab. VI– XI and Fig. 8– 9 show that HSC needs more computational time than other methods except K-AP.

The time complexity of the HSC algorithms is $O(n^3)$, where n is the number of data points. The time complexity of HSC is mainly dependent on the spectral clustering with its computational complexity of $O(n^3)$ in general.

For other clustering algorithms, the time complexity of k-means is $O(knt)$, where n is the number of objects and t is the number of iterations. The worst-case time complexity of the DBSCAN and HC is $O(n^2)$. The time complexity can decrease to $O(nlogn)$ if the spatial index is used in DBSCAN. The time complexity of GM_EM is linearity relation with the number of features, the number of objects and the number of iterations.

In summary, it indicates that the HSC algorithm could be used for the applications that do not have much real time requirement. However, there are also many

works to improve the computational time complexity of the spectral clustering. A fast spectral clustering method with k-means was presented to reduce the time complexity to the boundary of $O(k^3) + O(knt)$ [34].

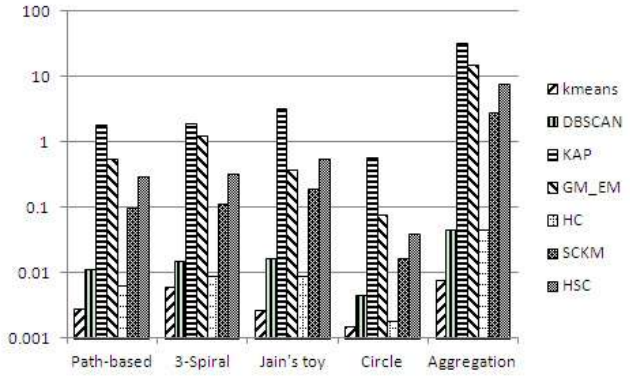


Fig. 8 *The comparison of computational time on the synthetic datasets.*

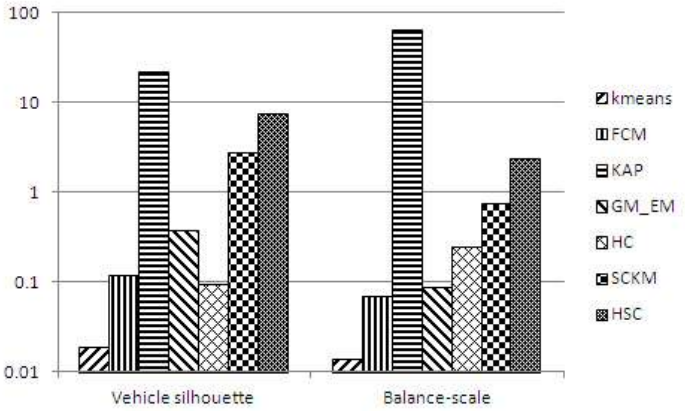


Fig. 9 *The comparison of computational time on the real datasets.*

5. Discussion

5.1 Estimation of number of clusters

The number of clusters k is unknown in most practical clustering applications. However, it is widely believed that determining the number of clusters automatically is one of the challenges in unsupervised machine learning. No theoretically optimal method for finding the number of clusters inherently present in the data

has been proposed so far. In literatures, three kinds of methods were presented to find the number of clusters, stability-based, model-fitting-based and metric-based.

During the study of our experiments, we observed an interesting result that the optimal number of clusters can be found by iteratively searching the space of k with the best ARI obtained from HSC. The estimation of k exactly matches the actual number of clusters on all these datasets expect the last one. We also examined the k with the best ARI calculated by other clustering methods, and found that only HSC can get the excellent estimation of the number of clusters overall.

Tab. XII shows the comparison of the estimated number of clusters with the best ARI calculated by different clustering methods. From these empirical studies, searching the space of k with the best ARI by using HSC could be one of the effective methods to estimate the number of clusters. Since ARI measures the similarity between two data clusterings and the chance of grouping data points, it is also an evidence of that HSC could separate different clusters accurately. However, we do not go into the details of theoretical principles that is out of scope of this study, but in future works.

	k	k-means	DBSCAN	KAP	GM_EM	HC	SCKM	HSC
Path-based	3	5(+2)	3(+0)	3(+0)	2(-1)	6(+3)	8(+5)	3(+0)
3-Spiral	3	7(+4)	3(+0)	11(+8)	4(+1)	3(+0)	3(+0)	3(+0)
Jain's toy	2	2(+0)	4(+2)	3(+1)	2(+0)	5(+3)	2(+0)	2(+0)
Circle	2	2(+0)	5(+3)	2(+0)	2(+0)	1(-1)	2(+0)	2(+0)
Aggregation	7	6(-1)	7(+0)	5(-2)	8(+1)	6(-1)	6(-1)	7(+0)
Vehicle silhouette	4	7(+3)	-	7(+3)	4(+0)	1(-3)	6(+2)	4(+0)
Balance-scale	3	3(+0)	-	9(+6)	4(+1)	2(-1)	2(-1)	2(-1)

Tab. XII *The comparison of the estimated number of clusters on the different datasets.*

5.2 Number of eigenvectors

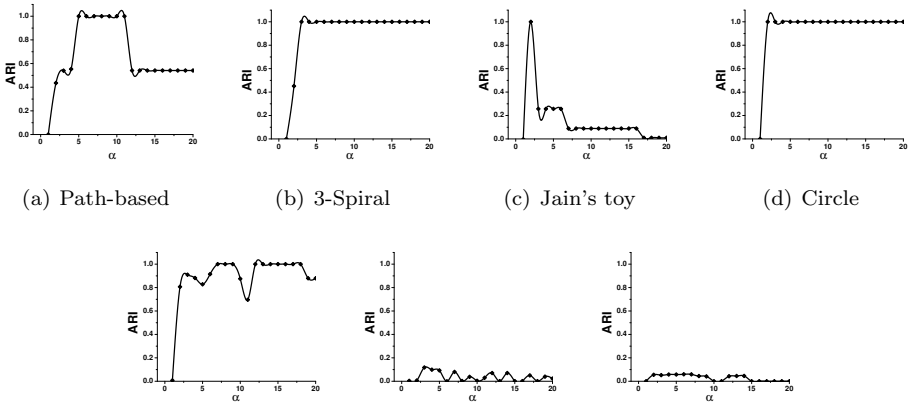
Eigenvectors is significant because using uninformative/irrelevant eigenvectors could lead to poor clustering results. An analysis of the characteristics of eigenspace showed that not every eigenvectors of a data affinity matrix is informative and relevant for clustering and the corresponding eigenvalues cannot be used for relevant eigenvector selection given a realistic data set.[35] Therefore, we investigated the influence of the number of eigenvectors of HSC on the clustering results in this study.

Fig. 10 shows that there is not general rules for all of the datasets. However, the number of eigenvectors with the best ARI values are between 2 and 5 for all of the datasets. Besides, the ARI moves towards stabilization in the best ARI when the number of eigenvectors increases for all of synthetic datasets except Jain's toy for which it moves towards stabilization but in the worst ARI. On the other hand, ARI seems fluctuating periodically for the real datasets.

Fig. 11 shows the influence of the number of eigenvectors on DB-Index. The trend is similar with that of ARI. The number of eigenvectors with the best DB-Index values are between 2 and 5 for all of the datasets. And it seems there are the

stabilization trends for all the synthetic datasets and the periodically fluctuation for the real datasets.

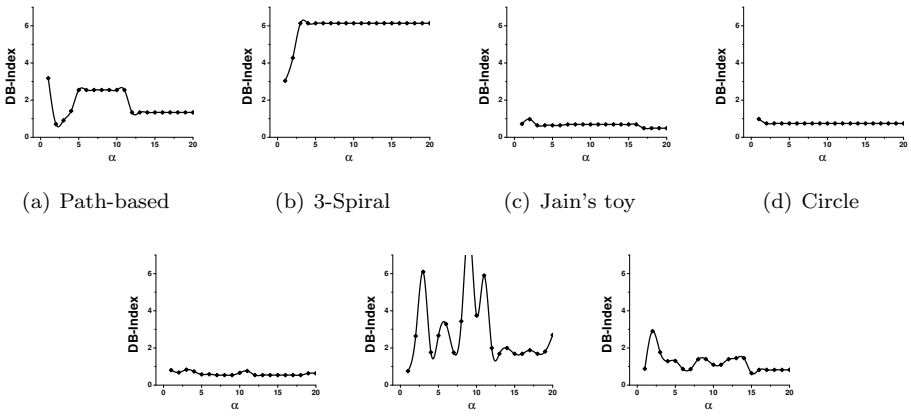
Although the explanations of the relationship between the number of eigenvectors and the clustering results are not given in this study which is believed that it is hardly possible to produce a completely survey on all datasets, the eigenvectors has the impact on performing effective clustering given noisy neighboring data. From our empirical studies, the number of eigenvectors between 2 to 5 is acceptable for all the databases according to the evaluation metrics of ARI and DB-Index.



(a) Path-based (b) 3-Spiral (c) Jain's toy (d) Circle

(e) Aggregation (f) Vehicle silhouette (g) Balance-scale

Fig. 10 *The influence of the number of eigenvectors on ARI.*



(a) Path-based (b) 3-Spiral (c) Jain's toy (d) Circle

(e) Aggregation (f) Vehicle silhouette (g) Balance-scale

Fig. 11 *The influence of the number of eigenvectors on DB-Index.*

6. Conclusion and Future Work

In this paper we presented a novel spectral clustering method based on hierarchical clustering. The main idea is to use the hierarchical clustering instead of the k-means in a traditional spectral clustering to eliminate the misleading information of the noisy neighboring data points. Experiments on real and synthetic datasets showed that the HSC method outperforms overall commonly used methods when considering the evaluations of accuracy and computational time. Besides, We observed that HSC could be used for finding the number of clusters. Furthermore, the HSC also has the advantage in practical implementation. It is not a randomized method, and thus the result can be repeated for the same dataset with the same settings. And it can be easily applied to different kinds of clustering problems, including multidimensional dataset clustering.

An extension of this work include the selection of parameters α which has a significant impact on the performance of HSC. Artificial intelligence approaches could be able to optimize the parameter to find the best evaluation metrics. Neural network, evolutionary programming and particle swarm optimization have been enormously successful to optimize the parameters in many fields. We would use these methods to adjust α to further improve the performance of HSC algorithm in our future work. Another issue of HSC is the time complexity. Fast algorithms for approximate spectral clustering with a lower computational time complexity could be incorporated in HSC. Furthermore, our future research will also focus on analyzing the theoretical principles of the phenomenon that HSC outperforms other clustering algorithms to estimate the number of clusters by iteratively searching the best ARI.

Acknowledgement

The authors thank the editor and the anonymous reviewers for their valuable comments and constructive comments. Their suggestions have led to a major improvement of this paper.

This work was partially supported by the National Natural Science Foundation of China (grant no.61003240), the Scientific Research Foundation for the Returned Overseas Chinese Scholars(grant order no.44th), and the Opening Project of Guangdong Province Key Laboratory of Computational Science at the Sun Yat-sen University (grant year 2012).

References

[1] Qiu H., Hancock E. R.: Graph matching and clustering using spectral partitions. Journal of the Pattern Recognition Society, 39(1): 22-24, January 2006.

[2] Lloyd S. P.: Least squares quantization in PCM. IEEE Transactions on Information Theory, 28(2):129-137, March 1982.

[3] Dempster A. P., Laird N. M., Rubin D. B.: Maximum Likelihood from Incomplete Data via the EM Algorithm[J]. Journal of the Royal Statistical Society-Series B (Methodological), Vol. 39 (1): pp. 1-38, 1977.

[4] Gao Y., Gu S., Tang J.: Research on Spectral Clustering in Machine Learning. Computer Science, 34(2): 201-203, 2007.

[5] Ng A. Y., Jordan M., Weiss Y.: On Spectral Clustering: Analysis and an algorithm. In Advances in Neural Information Processing Systems (NIPS), 2002.

[6] Ding S., Zhang L., Zhang Y.: Research on Spectral Clustering Algorithms and Prospects. The 2nd International Conference on Computer Engineering and Technology (ICCET), 6: 149-153, April 2010.

[7] Von Luxburg U.: A tutorial on Spectral Clustering[J]. Statistics and Computing. Vol. 17(4), pp. 395-416, Dec. 2007.

[8] Wang C., Wang J., Zhen J.: Application of Spectral Clustering in Image Retrieval. Computer Technology and Development, vol.19 (1), pages 207-210, January 2009.

[9] White S., Smyth P.: A spectral clustering approach to finding communities in graph. In: Proceedings of the 5th SIAM International Conference on Data Mining (SDM 2005), pp. 76-84, 2005.

[10] Bach F. R., Jordan M. I.: Spectral clustering for speech separation. Automatic Speech and Speaker Recognition: Large Margin and Kernel Methods. pp. 221-253, Jan. 2009.

[11] Zhang Z., Jordan M. I.: Multiway Spectral Clustering: A Margin-Based Perspective, Statistical science, Vol. 23 (3): pp. 383-403, 2008.

[12] Jiang Y., Tang C. etc.: CTSC: Core-Tag oriented Spectral Clustering Algorithm on Web2.0 Tags. The Sixth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 09),1:460-464, August 2009.

[13] Alzate C., Suykens J. A. K.: Multiway Spectral Clustering with Out-of-Sample Extensions through Weighted Kernel PCA. IEEE Transactions on Pattern Analysis and Machine Intelligence,32(2):335-347, February 2010.

[14] Li Z., Sun W.: A New Method to Calculate Weights of Attributes in Spectral Clustering Algorithms. 2011 International Conference on Information Technology, Computer Engineering and Management Sciences (ICM), Vol.2, pp.58-60, Sep.2011.

[15] Alzate C., Suykens J. A. K.: Locally-Scaled Spectral Clustering using Empty Region Graphs, KDD '12 Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp.1330-1338, 2012.

[16] Ekin A., Pankanti S., Hampapur A.: Initialization-independent Spectral Clustering with applications to automatic video analysis. IEEE International Conference on Acoustics, Speech and Signal Processing,3:641-644, May 2004.

[17] Wang H., Chen J., Guo K.: A Genetic Spectral Clustering Algorithm. Journal of Computational Information Systems 7(9): 3245-3252,2011.

[18] [Online] Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[19] Tian Z., Li X., Ju Y.: The perturbation analysis of the Spectral clustering. Chinese Science, 37(4):527-543, 2007.

[20] Shi J., Malik J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence,22 (8):888-905, August 2000.

[21] Meila M., Shi J.: Learning segmentation with random walk. In Advances in Neural Information Processing Systems (NIPS),pages 470-477,2001.

[22] Yu S., Shi J. B.: Multiclass Spectral Clustering. Ninth IEEE International Conference on Computer Vision,1:313-319, October 2003.

[23] Qian W., Zhou A.: Analyzing Popular Clustering Algorithms from Different Viewpoints. Journal of Software,13(8):1382-1394, 2002.

[24] Gower J. C., G. J. S.: Minimum Spanning Trees and Single Linkage Cluster. Journal of the Royal Statistical Society. Series C (Applied Statistics), 18(1):54-64, 1969.

[25] Ester M., Kriegel H. P., Sander J., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise[C] KDD-96 Proceedings. 96: 226-231, 1996.

[26] Bezdek J. C., Ehrlich R., Full W.: FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences, 10(2-3):191-203, 1984.

[27] Zhang X. etc. K-AP: Generating Specified K Clusters by Efficient Affinity Propagation. IEEE 10th International Conference on Data Mining (ICDM), pages 1187-1192, 2010.

[28] Hong C., Yeung D. Y.: Robust path-based spectral clustering. Pattern Recognition, 41 (1): 191-203, January 2008.

[29] Aristides Gionis, Heikki Mannila, Panayiotis Tsaparas: Clustering Aggregation. 21st International Conference on Date of Conference, pages 341-352, April 2005.

[30] Davies D. L., Bouldin D. W.: A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence 1:224-227,1979.

[31] Hubert L. J., Arabie P.: Comparing partitions. Journal of Classification,2:193-218,1985.

[32] [Online] Available: http://en.wikipedia.org/wiki/Rand_index.

[33] Lu Y., Wan Y..: Clustering by Sorting Potential Values (CSPV): A novel potential-based clustering method. Pattern Recognition,45(9):3512-3522, September 2012.

[34] Yan D., Huang L., Jordan M. I.: Fast approximate spectral clustering[C]//Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009: 907-916.

[35] Xiang T., Gong S.: Spectral clustering with eigenvector selection[J]. Pattern Recognition, 2008, 41(3): 1012-1029.