

# Soft Subspace Clustering with Competitive Agglomeration

Lin Zhu

Institute of Image Processing and  
Pattern Recognition  
Shanghai Jiao Tong University  
Shanghai, China  
Email: zhulin@sjtu.edu.cn

Longbing Cao\*

Faculty of Engineering and IT,  
QCIS  
University of Technology, Sydney  
Sydney, Australia  
Email: longbing.cao-1@uts.edu.au

Jie Yang

Institute of Image Processing and  
Pattern Recognition  
Shanghai Jiao Tong University  
Shanghai, China  
Email: jieyang@sjtu.edu.cn

**Abstract**—In this paper, two novel soft subspace clustering algorithms, namely fuzzy weighting subspace clustering with competitive agglomeration (FWSCA) and entropy weighting subspace clustering with competitive agglomeration (EWSA), are proposed to overcome the problems of the unknown number of clusters and the initialization of prototypes for soft subspace clustering. The main advantage of FWSCA and EWSA lies in the fact that they effectively integrate the merits of soft subspace clustering and the good properties of fuzzy clustering with competitive agglomeration. This makes it possible to obtain the appropriate number of clusters during the clustering progress. Moreover, FWSCA and EWSA algorithms can converge regardless of the initial number of clusters and initialization. Substantial experimental results on both synthetic and real data sets demonstrate the effectiveness of FWSCA and EWSA in addressing the two problems.

**Keywords**—subspace clustering; competitive agglomeration; fuzzy weighting subspace clustering; entropy weighting subspace clustering; optimal number of clusters

## I. INTRODUCTION

Clustering has long been a hot research topic in various disciplines. It has been widely utilized as an efficient tool for data analysis and visualization in areas such as machine learning and data mining [1]. Clustering partitions the unlabeled input vectors into different groups, called clusters, such that data points within a cluster are more similar to each other than they are to those belonging to different clusters, i.e., by maximizing the intra-cluster similarity while minimizing the inter-class similarity [2]. However, clustering high dimensional data has become an emerging research field, since dimensions might have locally varying relevance for different groups of objects in high dimensional spaces [3]. For example, given a subset of data points identified in a cluster, it is possible that these points exhibit different characteristics from others when varied subsets of dimensions are observed.

The difficulty that traditional clustering algorithms encounter in dealing with high dimensional data sets inspired the invention of subspace clustering or projected clustering [4], which has been studied extensively in recent years [3-13]. The goal of subspace clustering is to locate clusters embedded in different subspaces of the original data space with their own associated dimensions. Based on the ways that the subspaces of clusters are determined, subspace clustering algorithms can be generally classified into two main categories: hard subspace

clustering and soft subspace clustering [5]. Hard subspace clustering methods identify the exact subspaces for different clusters, which can be divided into bottom-up [4, 6] and top-down [7] subspace methods according to their subspace searching methods. Soft subspace clustering methods can measure the importance of each dimensionality to a particular cluster in the clustering process by automatically assigning different weightings to different dimensions of clusters embedded in subspaces. In this paper, we focus on soft subspace clustering. Many different soft subspace clustering algorithms have been proposed recently [8-13] and the resulting algorithms can be grouped into two main categories [9, 11]: fuzzy weighting subspace clustering (FWSC) and entropy weighting subspace clustering (EWSA). The final partition of both the FWSC and EWSA approaches assumes that the number of clusters is known prior to clustering. However, this is rarely the case and the optimal number of clusters is hard to determine.

In order to deal with the above problem for traditional clustering algorithms, Frigui and Krishnapuram introduced a regularization term to the original objective function of FCM and derived the fuzzy clustering with competitive agglomeration algorithm (FCA) [14]. In FCA, the initial partition has an over-specified number of clusters, which is progressively reduced until an optimum is reached. Zhang et al. revisited and reformulated the FCA algorithm from the point of view of entropy minimization, and proposed four generalized competitive clustering algorithms: RECA, SECA, CA-p and RECA-p [15]. Grira et al. presented a new algorithm, active fuzzy constrained clustering, in the categorization of image database by minimizing a competitive agglomeration cost function with fuzzy terms corresponding to pairwise constraints provided by the user [16]. Recently, Jeng et al. proposed an interval competitive agglomeration clustering algorithm, which uses the theory of competitive agglomeration for clustering the symbolic interval-values data [17]. However, all these competitive agglomeration-based clustering algorithms are applied in full dimensional space. It is anticipated that the performance of clustering can be further enhanced by including subspace selection in the clustering process.

By considering the advantages of competitive agglomeration, clustering algorithms can be rapidly converged in a few iterations regardless of the initial number of clusters, and can also converge to the same optimal partition regardless of its initialization. Inspired by this fact of competitive agglomeration, we propose two novel soft subspaces clustering

\* Corresponding author: email (longbing.cao-1@uts.edu.au)

algorithms: fuzzy weighting subspace clustering with competitive agglomeration (FWSCA) and entropy weighting subspace clustering with competitive agglomeration (EWSCA). FWSCA and EWSCA not only inherit the merits of soft subspace clustering, but also receive the beneficial properties of fuzzy clustering with competitive agglomeration. Our experiment results demonstrate that FWSCA and EWSCA are generally effective in subspace clustering and can achieve superior performance over traditional soft subspace clustering and competitive agglomeration-based clustering algorithms in most cases. Furthermore, a link between FWSCA, EWSCA algorithms and FWSC, EWSC algorithms is established in Section III.A. Thus, the FWSC and EWSC algorithms are regarded as a special case of the proposed FWSCA and EWSCA algorithms. The rest of this paper is organized as follows. In Section II, a brief overview of existing algorithms for soft subspace clustering is provided. The proposed FWSCA and EWSCA algorithms are presented and their properties are explained in Section III. In Section IV, the clustering performances of FWSCA and EWSCA, as compared with other algorithms, on artificial and UCI data sets are reported. Finally, conclusions and future work are presented in Section V.

## II. SOFT SUBSPACE CLUSTERING

Soft clustering aims at grouping a set of given data points  $X = \{x_1, x_2, \dots, x_n\} \subset R^d$  into a set of clustering prototypes  $V = \{v_i, 1 \leq i \leq c\}$ . Let  $u_{ij}$  denote the membership degree of  $x_j$  belonging to  $v_i$ , then we can define the fuzzy  $c$ -partition  $U$  of the given data set, i.e.,  $U = \{u_{ij} \mid 1 \leq i \leq c, 1 \leq j \leq n\}$ . To discover clusters from different subspaces, it is important that a soft clustering algorithm has the capability of clustering data points in a subspace by automatically weighting features in the clustering process. Based on the importance of the  $k$ -th dimension on the formation of  $i$ -th cluster, a weighting  $w_{ik}$  is assigned to each dimension in soft subspace clustering. The subspaces of different clusters can be identified by the weight matrix  $W = \{w_{ik} \mid 1 \leq i \leq c, 1 \leq k \leq d\}$  after clustering [11, 12].

### A. Fuzzy weighting subspace clustering

Fuzzy weighting subspace clustering (FWSC) aims to find clusters from fuzzy weighting subspaces [8, 10, 12]. In all the fuzzy weighting subspace clustering algorithms, a fuzzy weight  $w_{ik}^m$  is assigned to the features of different clusters with a fuzzy weighting index  $m$ . The objective function of FWSC is generally formulated as:

$$J_{FWSC} = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^2 \sum_{k=1}^d w_{ik}^m (x_{jk} - v_{ik})^2 \quad (1)$$

$$\text{s. t. } 0 \leq u_{ij} \leq 1, \sum_{i=1}^c u_{ij} = 1, 0 \leq w_{ik} \leq 1, \sum_{k=1}^d w_{ik} = 1.$$

By minimizing (1) using Lagrangian optimization, the updating equation for estimating center  $v_i$ , fuzzy weighting  $w_{ik}$  and membership  $u_{ij}$  can be derived.

### B. Entropy weighting subspace clustering

The entropy concept, which is used to represent the certainty of dimensions in the identification of a cluster, is introduced into soft subspace clustering [9, 11, 13]. Because the weights in extended subspace clustering algorithms are controllable by entropy, these types of algorithms are referred to as entropy weighting subspace clustering (EWSC). In general, the objective function of EWSC can be defined as:

$$J_{EWSC} = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^2 \sum_{k=1}^d w_{ik} (x_{jk} - v_{ik})^2 \quad (2)$$

$$+ \gamma \sum_{i=1}^c \sum_{k=1}^d w_{ik} \log w_{ik}$$

$$\text{s. t. } 0 \leq u_{ij} \leq 1, \sum_{i=1}^c u_{ij} = 1, 0 \leq w_{ik} \leq 1, \sum_{k=1}^d w_{ik} = 1.$$

The Lagrange multipliers method is also applied to minimize the objective function in (2). It can result in the equations for updating the estimations of center  $v_i$ , entropy weighting  $w_{ik}$  and membership  $u_{ij}$ . In (1) and (2), to cater for the fuzzy partition, we specify  $0 \leq u_{ij} \leq 1$ , while  $u_{ij} \in \{0, 1\}$  is used in the K-means framework-based FWSC [8, 10, 12] and EWSC algorithms [11, 13] for hard partition.

By observing the existing soft subspace clustering algorithms, we find that the final partition of both FWSC and EWSC is based on the assumption that the number of clusters is known prior to clustering. However, this is rarely the case, and the optimal number of clusters is difficult to determine. As a result, the clustering algorithms are sensitive to the initialization. An effective way to overcome the above problems is to extend the concept of the competitive agglomeration (CA) in FWSC and EWSC, since CA can help to determine the initialization and the number of clusters during the clustering progress.

## III. SOFT SUBSPACE CLUSTERING WITH COMPETITIVE AGGLOMERATION

### A. Objective function based on the theory of competitive agglomeration

How to find an “appropriate” number of clusters is an important issue for partition-based clustering. Fuzzy clustering with competitive agglomeration (FCA) was introduced to solve the problem by adding a regularization term to the cost function [14]. The regularization term can avoid overfitting of a data set with too many prototypes, make clusters compete for membership of data points, and result in the number of clusters gradually being reduced until an optimum is reached.

To incorporate the idea of competitive agglomeration into FWSC and EWSC, we add a term to agglomerate all data points into one cluster in the objective functions to generate fuzzy weighting and entropy weighting-based subspace clustering with competitive agglomeration, namely FWSCA, EWSCA, which minimize:

$$J_{FWSCA} = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^2 \sum_{k=1}^d w_{ik}^m (x_{jk} - v_{ik})^2 - \alpha \sum_{i=1}^c \left[ \sum_{j=1}^n u_{ij} \right]^2 \quad (3)$$

and

$$J_{EWSCA} = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^2 \sum_{k=1}^d w_{ik} (x_{jk} - v_{ik})^2 + \gamma \sum_{i=1}^c \sum_{k=1}^d w_{ik} \log w_{ik} - \alpha \sum_{i=1}^c \left[ \sum_{j=1}^n u_{ij} \right]^2 \quad (4)$$

under the same constraint of (1) or (2).

By comparing the objective functions (3) and (4) for FWSCA and EWSCA and those in (1) and (2) for FWSC and EWSC, the first terms of objective functions for FWSCA and EWSCA are similar to those of FWSC and EWSC. The first term of FWSCA is the sum of the squared distances to the prototypes weighted by the constrained feature weightings and fuzzy memberships; and the first term of EWSCA is the sum of the intra-cluster dispersions and the negative weight entropy. Both components allow us to control the shapes and sizes of the clusters and lead to the compact clusters. The global minimum of these components is achieved when the number of clusters  $c$  is equal to the number of data points  $n$ . The last terms of (3) and (4) are the sum of squares of the cardinalities of the clusters, which allows us to control the number of clusters. By minimizing this term, all data points are agglomerated into one cluster, as shown by Theorem 1.

*Theorem 1:* Given  $0 \leq u_{ij} \leq 1$  and  $\sum_{i=1}^c u_{ij} = 1$ , the

$$\text{membership constraint function } f(u_{11}, u_{12}, \dots, u_{cn}) = - \sum_{i=1}^c \left[ \sum_{j=1}^n u_{ij} \right]^2$$

reaches its minimum, i.e.  $-n^2$ , when  $u_{ij}$  is equal to 1 and  $u_{kj}$  are equal to 0 for all other  $k \neq i$  ( $1 \leq j \leq n$ ).

Therefore, the parameter  $\alpha$  in (3) and (4) controls the balance between the two components in order to find an optimal number of clusters  $c$ . It should be noted that  $c$  is dynamically updated during the process of FWSCA and EWSCA. Fig. 1 shows the relationship between the objective function and the number of clusters for FWSCA and EWSCA on a synthetic data set used in Section IV. To show the dynamics of three curves simultaneously, all values are normalized into the interval  $[3, 15]$ , since the optimal number of clusters and the initial number of clusters are 3 and 15, respectively. It shows that the first term of the FWSCA and EWSCA objective function increases as the number of clusters  $c$  decreases. On the other hand, the last term of the objective functions becomes smaller when the number of clusters  $c$  decreases. The dynamics of the objective function shows that, if  $\alpha$  is properly chosen, adjacent clusters will compete for the data points; those clusters lost in competition will gradually be discarded and disappear. Consequently, an appropriate number of clusters is obtained.

## B. Fuzzy weighting subspace clustering with Competitive Agglomeration (FWSCA)

To minimize the objective function of FWSCA in (3), we apply Lagrange multipliers and obtain the  $V$  and  $W$  updating equation based on the theorems below.

*Theorem 2:* Given that  $U$  and  $W$  are fixed, and  $m > 1, \alpha > 0$ ,  $J_{FWSCA}$  reaches the local minimum if and only if  $V$  meets the following condition:

$$v_{ik} = \frac{\sum_{j=1}^n u_{ij}^2 x_{jk}}{\sum_{j=1}^n u_{ij}^2} \quad (5)$$

*Theorem 3:* Given that  $U$  and  $V$  are fixed, and  $m > 1, \alpha > 0$ ,  $J_{FWSCA}$  reaches the local minimum if and only if  $W$  meets the following condition:

$$w_{ik} = \frac{1}{\sum_{s=1}^d \left[ \frac{\sum_{j=1}^n u_{ij}^2 (x_{jk} - v_{ik})^2}{\sum_{j=1}^n u_{ij}^2 (x_{js} - v_{is})^2} \right]^{\frac{1}{m-1}}} \quad (6)$$

To obtain the updating equation  $u_{ij}$ ,  $J_{FWSCA}$  is minimized with respect to  $U$  for fixed  $V$  and  $W$ , and both are set to zero. We apply Lagrange multipliers to (3) and obtain:

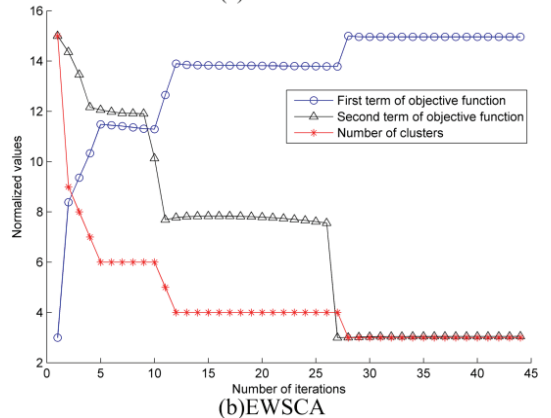
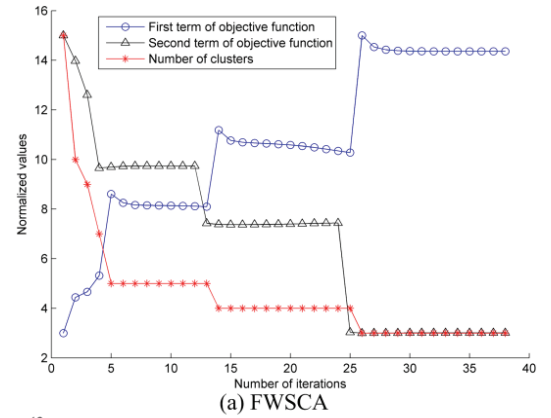


Fig. 1 The change curve of the objective function and the number of clusters. (a) FWSCA, (b) EWSCA

$$\begin{aligned} \phi(u_{ij}) = & \sum_{i=1}^c \sum_{j=1}^n u_{ij}^2 \sum_{k=1}^d w_{ik}^m (x_{jk} - v_{ik})^2 \\ & - \alpha \sum_{i=1}^c \left[ \sum_{j=1}^n u_{ij} \right]^2 - \sum_{j=1}^n \lambda_j^u \left( \sum_{i=1}^c u_{ij} - 1 \right) \end{aligned} \quad (7)$$

With (7), we can compute the optimal value of  $u_{ij}$  by setting  $\partial\phi/\partial u_{ij} = 0$ . These derivatives yield the following relationship:

$$\frac{\partial\phi}{\partial u_{ij}} = 2u_{ij} \left( \sum_{k=1}^d w_{ik}^m (x_{jk} - v_{ik})^2 \right) - 2\alpha \left( \sum_{j=1}^n u_{ij} \right) - \lambda_j^u = 0 \quad (8)$$

The solution of (8) can be simplified by assuming that the membership values do not change significantly from one iteration to another. With this assumption, and by computing the cardinality of cluster  $i$ ,  $N_i = \sum_{j=1}^n u_{ij}$  using the membership values from previous iteration, (8) is converted to

$$u_{ij} = \frac{2\alpha N_i + \lambda_j^u}{2 \left( \sum_{k=1}^d w_{ik}^m (x_{jk} - v_{ik})^2 \right)} \quad (9)$$

Using the membership constraint  $\sum_{s=1}^c u_{sj} = 1$ , we obtain

$$\lambda_j^u = \frac{1 - \alpha \sum_{s=1}^c \left[ N_s / \left( \sum_{k=1}^d w_{sk}^m (x_{jk} - v_{sk})^2 \right) \right]}{\sum_{s=1}^c \left[ 1 / \left( 2 \sum_{k=1}^d w_{sk}^m (x_{jk} - v_{sk})^2 \right) \right]} \quad (10)$$

Substituting (10) in (9), we obtain the following updated equation for the membership of data point  $x_j$  to center  $v_i$  after rearranging the terms.

$$u_{ij} = u_{ij}^{FCM} + u_{ij}^{Bias} \quad (11)$$

where

$$u_{ij}^{FCM} = \frac{1}{\sum_{s=1}^c \left[ \left( \sum_{k=1}^d w_{ik}^m (x_{jk} - v_{ik})^2 \right) / \left( \sum_{k=1}^d w_{sk}^m (x_{jk} - v_{sk})^2 \right) \right]} \quad (12)$$

and

$$u_{ij}^{Bias} = \alpha \frac{N_i - \bar{N}_j}{\sum_{k=1}^d w_{ik}^m (x_{jk} - v_{ik})^2} \quad (13)$$

In (13),  $\bar{N}_j$  is defined as

$$\bar{N}_j = \frac{\sum_{s=1}^c \left[ 1 / \left( \sum_{k=1}^d w_{sk}^m (x_{jk} - v_{sk})^2 \right) N_s \right]}{\sum_{s=1}^c \left[ 1 / \left( \sum_{k=1}^d w_{sk}^m (x_{jk} - v_{sk})^2 \right) \right]} \quad (14)$$

which is simply a weighted average of the cluster cardinalities.

The updated function (11) for FWSCA brings about several benefits. The first component in (11) is the same as that in FCM, and considers only the relative distances of a data point

to all clusters. The second component contributes to the reduction of the cardinality of spurious clusters, which are discarded when their cardinality drops below a threshold. Indeed,  $u_{ij}^{Bias}$  is also a signed bias term that depends on the difference between the cardinality  $N_i$  of the cluster  $i$  and the weighted average of the cardinalities  $\bar{N}_j$  from  $x_j$ . For the clusters with cardinality lower than the average value, the bias term is negative. The membership value is then decreased.

Further,  $\alpha$  plays an important role in FWSCA in (13) for balancing the two components of (11). To make FWSCA independent of the distance measure,  $\alpha$  should be chosen such that both terms in (11) are of the same order of magnitude [14, 16]. Thus,  $\alpha$  should be proportional to the ratio of the two terms. In this paper,  $\alpha$  is chosen as

$$\alpha(itr) = \tau(itr) \frac{\sum_{i=1}^c \sum_{j=1}^n u_{ij}^2 \sum_{k=1}^d w_{ik}^m (x_{jk} - v_{ik})^2}{\sum_{i=1}^c \left[ \sum_{j=1}^n u_{ij} \right]^2} \quad (15)$$

where  $itr$  is an iteration index. The best choice for  $\tau$  is the exponential factor defined in [16]

$$\tau(itr) = \tau_0 \exp(-|itr - itr_0| / \xi) \quad (16)$$

In our experiments, the initial value  $\tau_0$ , the time constant  $itr_0$ , and  $\xi$  in (16) are set as 0.8, 10 and 20, respectively.

Finally, let us discuss the agglomeration process of FWSCA. As discussed in [16], FCA [14] discards spurious clusters whose cardinalities drop below a threshold; however the clustering is sensitive to the threshold parameter. In this paper, we investigate an alternative process to improve the agglomeration. The details of the process are given as follows.

We first compute the average cardinality of clusters at the  $itr$ -th iteration below.

$$Card_{avg}(itr) = \frac{\sum_{i=1}^c \sum_{j=1}^n u_{ij}}{num\_of\_cluster(itr)} = \frac{n}{c(itr)} \quad (17)$$

where  $itr$  is an iteration index,  $n$  is the number of data points and  $c(itr)$  is the number of clusters at the  $itr$ -th iteration. So the minimum cardinality threshold ( $MCT$ ) can be defined by

$$MCT(itr) = \eta \times Card_{avg}(itr) \quad (18)$$

where  $\eta$  is the agglomeration threshold parameter. When the cardinality of a cluster reduces below the  $MCT$  at the  $itr$ -th iteration, we discard the cluster, and update the number of clusters. The selection of  $\eta$  will be discussed in the experiments in Section IV.

Next, we compute the distances between pairs of prototypes  $d(r)$ , for  $r = 1, \dots, R$ ,  $R = c(itr) \times (c(itr) - 1) / 2$ . If  $min$  is the index corresponding to the pair having the minimal distance

### Procedures of FWSCA clustering algorithm

- Step1.** Set the maximum number of clusters  $c=c_{max}$  ( $2 \leq c_{max} \leq n$ ), the fuzzy weighting index  $m$  and the agglomerate threshold parameter  $\eta$ , initialize the fuzzy partition  $u_{ij}^{(1)}$  and fix the iterative index  $itr=1$ ;
- Step2.** Compute the clustering centers matrix  $V(itr+1)$  using Eq.(5);
- Step3.** Compute the weight matrix  $W(itr+1)$  using Eq. (6);
- Step4.** Compute the cardinality  $N_i$  for  $1 \leq i \leq c$ ;
- Step5.** Update  $\alpha(itr)$  using Eq. (15) and Eq. (16);
- Step6.** Compute the partition matrix  $U(itr+1)$  using Eq. (11);
- Step7.** If  $N_i$  is smaller than minimum cardinality threshold (Eq. (18)) or the minimal distance between pairs of prototypes is satisfied Eq. (19) then discard the cluster;
- Step8.** Update the number of clusters  $c(itr+1)$ ;
- Step9.** If prototype parameters are stabilized, then output the clustering result and terminate, otherwise  $itr=itr+1$ , go to Step 2.

Figure 2. Description of the proposed FWSCA algorithm

$d_{min}$ , then we discard the cluster, and update the number of clusters when the following criterion is satisfied:

$$d_{min} < 0.2 \times \sum_{r=1}^R d(r)/R \quad (19)$$

where  $d_{min} = \min\{d(r) | r = 1, \dots, R\}$  and  $\sum_{r=1}^R d(r)/R$  is the average distances between pairs of prototypes at the  $itr$ -th iteration.

Fig. 2 shows the procedure of FWSCA algorithm.

### C. Entropy weighting subspace clustering with Competitive Agglomeration (EWSCA)

To minimize the objective function of EWSCA defined in (4), the Lagrange multipliers method is applied, and we obtain the following Theorems 4 and 5, which can help the optimization of the EWSCA procedure.

**Theorem 4:** Given that  $U$  and  $W$  are fixed, and  $\gamma > 0, \alpha > 0$ ,  $J_{EWSCA}$  reaches the local minimum if and only if  $V$  meets the following condition:

$$v_{ik} = \frac{\sum_{j=1}^n u_{ij}^2 x_{jk}}{\sum_{j=1}^n u_{ij}^2} \quad (20)$$

**Theorem 5:** Given that  $U$  and  $V$  are fixed, and  $\gamma > 0, \alpha > 0$ ,  $J_{EWSCA}$  reaches the local minimum if and only if  $W$  meets the following condition:

$$w_{ik} = \frac{\exp(-\sum_{j=1}^n u_{ij}^2 (x_{jk} - v_{ik})^2 / \gamma)}{\sum_{s=1}^d \exp(-\sum_{j=1}^n u_{ij}^2 (x_{js} - v_{is})^2 / \gamma)} \quad (21)$$

Similar to the deriving procedures of FWSCA, we obtain the updating equation  $u_{ij}$  of EWSCA:

$$u_{ij} = u_{ij}^{FCM'} + u_{ij}^{Bias'} \quad (22)$$

where

$$u_{ij}^{FCM'} = \frac{1}{\sum_{s=1}^c \left[ \left( \sum_{k=1}^d w_{ik} (x_{jk} - v_{ik})^2 \right) / \left( \sum_{k=1}^d w_{sk} (x_{jk} - v_{sk})^2 \right) \right]} \quad (23)$$

$$u_{ij}^{Bias'} = \alpha \frac{N_i - \bar{N}_j}{\sum_{k=1}^d w_{ik} (x_{jk} - v_{ik})^2} \quad (24)$$

### Procedures of EWSCA clustering algorithm

- Step1.** Set the maximum number of clusters  $c=c_{max}$  ( $2 \leq c_{max} \leq n$ ), the entropy weighting index  $\gamma$  and the agglomerate threshold parameter  $\eta$ , initialize the fuzzy partition  $u_{ij}^{(1)}$  and fix the iterative index  $itr=1$ ;
- Step2.** Compute the clustering centers matrix  $V(itr+1)$  using Eq.(20);
- Step3.** Compute the weight matrix  $W(itr+1)$  using Eq. (21);
- Step4.** Compute the cardinality  $N_i$  for  $1 \leq i \leq c$ ;
- Step5.** Update  $\alpha(itr)$  using Eq. (15) and Eq. (16);
- Step6.** Compute the partition matrix  $U(itr+1)$  using Eq. (22);
- Step7.** If  $N_i$  is smaller than minimum cardinality threshold (Eq. (18)) or the minimal distance between pairs of prototypes is satisfied Eq. (19) then discard the cluster;
- Step8.** Update the number of clusters  $c(itr+1)$ ;
- Step9.** If prototype parameters are stabilized, then output the clustering result and terminate, otherwise  $itr=itr+1$ , go to Step 2.

Figure 3. Description of the proposed EWSCA algorithm.

In (24),  $\bar{N}_j$  is defined as

$$\bar{N}_j = \frac{\sum_{s=1}^c \left[ 1 / \left( \sum_{k=1}^d w_{sk} (x_{jk} - v_{sk})^2 \right) N_s \right]}{\sum_{s=1}^c \left[ 1 / \left( \sum_{k=1}^d w_{sk} (x_{jk} - v_{sk})^2 \right) \right]} \quad (25)$$

which is simply a weighted average of the cluster cardinalities.

The selection of  $\alpha$  and the agglomeration process of EWSCA are similar to that of FWSCA. Accordingly, we describe the procedures of EWSCA as shown in Fig. 3.

### D. Convergence and computational complexity of FWSCA and EWSCA

When (5), (6) and (11) are adopted,  $J_{FWSCA}$  will certainly converge, because FWSCA degenerates into FWSC when  $\alpha$  takes zero, and  $J_{FWSCA}$  with  $\alpha > 0$  can easily be proved to be convergent using the similar proof in [10]. Similarly, the proposed EWSCA algorithm is indeed an extension of EWSC. By employing a strategy similar to that used in EWSC [11], it can be inferred that the proposed EWSCA algorithm, at least along a subsequence, converges to either a local optimal solution or a saddle point of its objective function.

Thus, FWSC and EWSC can be regarded as special cases of FWSCA and EWSCA respectively. Consequently, FWSCA and EWSCA have the same upper bound of the computational complexity as FWSC and EWSC, i.e.,  $O(ncdT)$  where  $n$  denotes the number of all data points,  $c$  denotes the number of the cluster centers which varies from  $c_{max}$  to  $c_{final}$ ,  $d$  denotes the dimensionality of the data points, and  $T$  denotes the maximal number of iterations. As the numbers of clusters for FWSCA and EWSCA are decreased during the clustering progress, they actually require more computational time, and converge a little more slowly than FWSC and EWSC.

## IV. EXPERIMENTS AND EVALUATION

Two sets of experiments are conducted to compare the clustering results of FWSCA and EWSCA with FCA, FWSC and EWSC. All data sets are preprocessed by normalizing the feature in each dimension into the interval  $[0, 1]$ . The initial fuzzy partitions for FCA, FWSCA and EWSCA are obtained by performing the FCM algorithm in the first ten iterations respectively. Normalized mutual information (NMI) [18] and

Rand Index [19] are used to evaluate the performance of five clustering algorithms.

#### A. Performance evaluation on synthetic data sets

In this experiment, we adopt a simulated data set to examine whether FWSCA and EWSCA converge to the appropriate partition and evaluate the clustering performance of five algorithms. The data set consists of three clusters in 2-D space which satisfies the normal distribution: the centers of three clusters are (28, 22), (60, 30) and (45, 38) respectively. For the first two clusters, the number of data points per cluster is 120 and their respective covariance matrices are  $\begin{bmatrix} 100 & 0 \\ 0 & 9 \end{bmatrix}$

and  $\begin{bmatrix} 9 & 0 \\ 0 & 144 \end{bmatrix}$ , respectively. For the last cluster centered at (45, 38), there are 140 data points and its covariance matrix is  $\begin{bmatrix} 9 & 0 \\ 0 & 9 \end{bmatrix}$ . Fig.4 plots this data set, which is used to test the

clustering results, since the data points belonging to three clusters locate at different subspaces. In this experiment, we tried many different values of  $c_{max}$  ranging from 9 to 21, and set  $m=3$  in FWSC and FWSCA,  $\gamma=1$  in EWSC and EWSCA, and the agglomeration threshold parameter  $\eta$  of 0.75 in FCA, FWSAC and EWSCA.

First, we analyze the independence of the proposed algorithms from the selection of the initial number of the clusters  $c_{max}$ . By setting different  $c_{max}$  value (9, 12, 15, 18 and 21), we reduce the number of clusters to observe whether FWSCA and EWSCA converge to the same final partition. As shown in Fig. 5 (a) and (b), we find that FWSCA and EWSCA are insensitive to the initial number of clusters; both algorithms can converge in a few iterations as long as  $c_{max}$  is higher than the optimal number of clusters.

Second, we analyze the insensitivity of FWSCA and EWSCA to the initial clustering centers. Fig. 6 (a) and (b) show the relationship between the selection of the initial clustering centers and the number of iterations. The initial clustering centers are randomly selected with  $c_{max}=15$ . The results show that both algorithms converge to the same appropriate partition regardless of the initialization of centers.

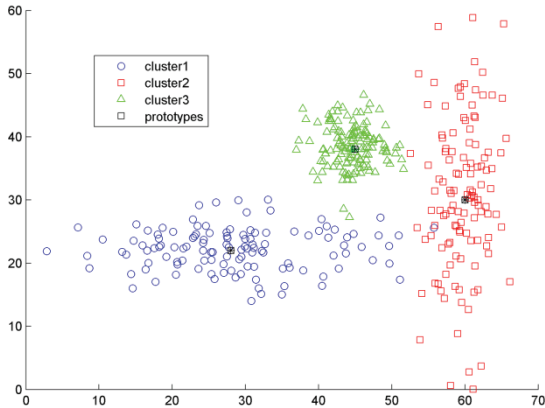


Figure 4. Synthetic data set.

Finally, we compare the clustering performance of FWSCA, EWSCA, FCA, FWSC and EWSC on the synthetic data set. Experiments are conducted by running each algorithm 20 times. The clustering results are evaluated in terms of the means and standard deviations of the NMI and Rand Index values, are shown in Table I. From the comparison, we observe that: (1) soft subspace clustering algorithms outperform FCA, since they integrate subspace feature weighting into the clustering process. (2) FWSCA and EWSCA outperform FWSC and EWSC by achieving better and more stable results. This is due to the fact that FWSC and EWSC are sensitive to the initialization, and they can easily get into a local minimum. (3) As we discussed in Section III.D, FWSCA and EWSCA have a slightly higher average number of iterations and average CPU running time (in seconds) than FWSC and EWSC as shown in Table II.

#### B. Performance evaluation on UCI data sets

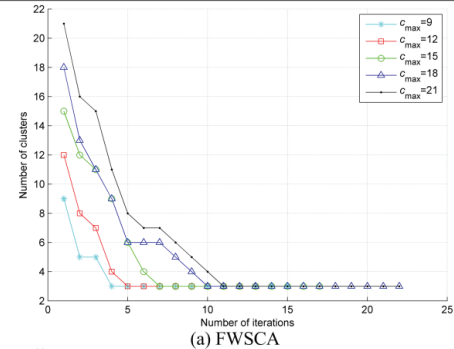
We further evaluate the performance of FWSCA and EWSCA against FCA, FWSC and EWSC on six UCI data sets [20] as shown in Table III. In our experience, FWSCA and EWSCA algorithms are independent of the initial number of

TABLE I. CLUSTERING RESULTS

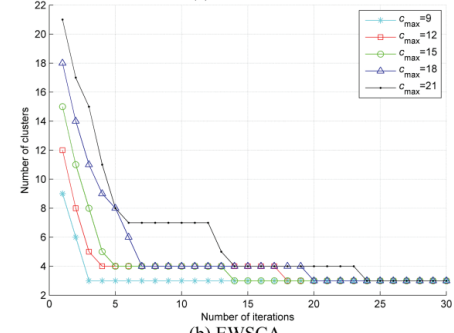
| Dataset    | FWSCA  | EWSCA         | FCA    | FWSC   | EWSC   |
|------------|--------|---------------|--------|--------|--------|
| NMI        |        |               |        |        |        |
| Mean       | 0.8882 | <b>0.9110</b> | 0.6889 | 0.8422 | 0.7710 |
| Std        | 0.0001 | 0.0001        | 0.0001 | 0.1418 | 0.0332 |
| Rand Index |        |               |        |        |        |
| Mean       | 0.9611 | <b>0.9731</b> | 0.8572 | 0.9373 | 0.8867 |
| Std        | 0.0001 | 0.0001        | 0.0001 | 0.0735 | 0.0203 |

TABLE II. ITERATION NUMBER AND CPU RUNNING TIME (SECONDS)

| Dataset           | FWSCA  | EWSCA  | FCA    | FWSC   | EWSC   |
|-------------------|--------|--------|--------|--------|--------|
| Iterations number |        |        |        |        |        |
| Mean              | 31.10  | 46.75  | 74.00  | 26.50  | 34.10  |
| Std               | 6.55   | 5.08   | 0.50   | 8.70   | 5.36   |
| Running time      |        |        |        |        |        |
| Mean              | 0.0853 | 0.1144 | 0.1197 | 0.0327 | 0.0386 |
| Std               | 0.0430 | 0.0410 | 0.0087 | 0.0176 | 0.0243 |



(a) FWSCA



(b) EWSCA

Figure 5. The effect of reducing the number of clusters on convergence with different  $c_{max}$  values for (a) FWSCA and (b) EWSCA.

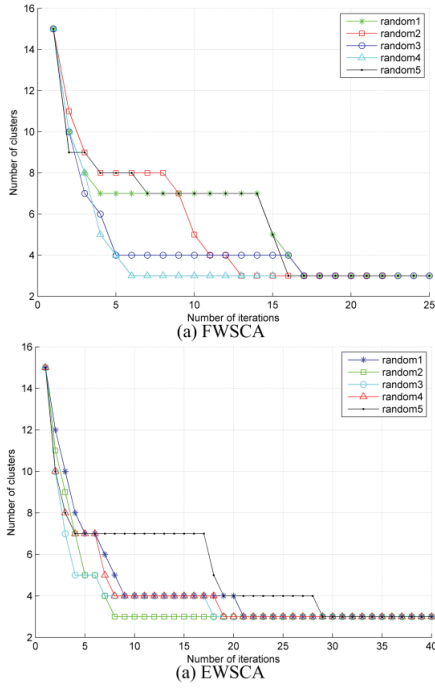


Figure 6. The effect of reducing the number of clusters on convergence with different choices of clustering centers for (a) FWSCA and (b) EWSCA.

clusters  $c_{max}$  when  $c_{max}$  is chosen to be much larger than the expected number of clusters and the agglomerate threshold parameter  $\eta$  is appropriately chosen. Our proposed algorithms can produce a sequence of partitions with a decreasing number of clusters and can finally find the appropriate number of clusters. In this paper, we display the clustering results when  $c_{max}$  is chosen to be 15.

First, we test the sensitivity of FWSCA and EWSCA on the agglomeration threshold  $\eta$ . For all six UCI data sets, experiments are performed for different values of  $\eta$  ranging from 0.6 to 0.9. For each  $\eta$ , the percentage of ‘match’ is calculated on the data sets by running each algorithm 20 times. Here ‘match’ means that the number of clusters identified by FCA, FWSCA and EWSCA is equal to the true number of clusters in a data set. Fig. 7 presents the match percentage on datasets iris and wine for FCA, FWSCA and EWSCA, under a series of agglomeration thresholds  $\eta$ .

As shown in Fig. 7, the match percentage of FWSCA and EWSCA is bigger than FCA, and the effective region of FWSCA and EWSCA is wider than that of FCA. Therefore, FWSCA and EWSCA are less sensitive to  $\eta$ .

Second, we compare the performance of FWSCA and EWSCA against FCA, FWSC and EWSC on the six UCI data sets. We evaluate the cluster validity index NMI and Rand Index under different  $\eta$  values when the algorithms find the true number of clusters. Experiments are conducted by running each algorithm 20 times. The clustering results are shown in terms of mean and standard deviation of NMI and Rand Index,

TABLE III UCI DATA SETS USED IN THE EXPERIMENT

| Data sets information | Iris | Wine | Heart | Australian | Blood | Pid |
|-----------------------|------|------|-------|------------|-------|-----|
| Size of data set      | 150  | 178  | 270   | 690        | 748   | 768 |
| Number of dimensions  | 4    | 13   | 13    | 14         | 4     | 8   |
| Number of clusters    | 3    | 3    | 2     | 2          | 2     | 2   |

as in Tables IV and V respectively. They show that EWSCA performs better than all other algorithms in most cases.

Finally, Tables VI and VII show the average number of iterations and the average CPU running time (in seconds) of the five algorithms on the UCI data sets. We can see that the iteration numbers and CPU running time of FWSCA and EWSCA are higher than those of FWSC and EWSC. This is due to the fact that the number of clusters for FWSCA and EWSCA decreases during the clustering progress. We also find that the computational complexity of FWSCA and EWSCA is sometimes lower than FCA, since the selection of different subspaces can accelerate the clustering convergence.

In summary, experimental results on both synthetic and real data sets demonstrate that FWSCA and EWSCA have the following advantages. First, both FWSCA and EWSCA outperform the classic soft subspace clustering and competitive agglomeration-based clustering algorithms in most cases. Second, FWSCA and EWSCA can obtain the appropriate number of clusters during the clustering progress regardless of the initial number of clusters. Third, FWSCA and

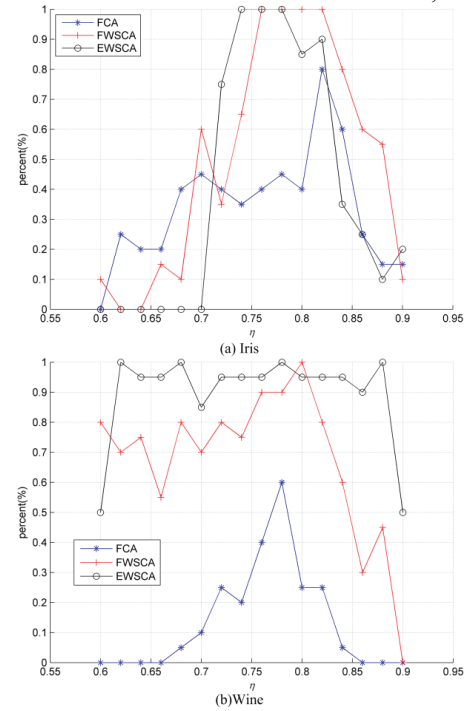


Figure 7. Percentage of the identified number of clusters by an algorithm versus the true number of clusters for different  $\eta$  (a) Iris and (b) Wine.

TABLE IV. CLUSTERING RESULTS IN TERMS OF NMI.

| Dataset    | FWSCA         | EWSCA         | FCA           | FWSC   | EWSC   |
|------------|---------------|---------------|---------------|--------|--------|
| Iris       |               |               |               |        |        |
| Mean       | 0.8623        | <b>0.8801</b> | 0.7433        | 0.8246 | 0.8025 |
| Std        | 0.0047        | 0.0034        | 0.0026        | 0.1685 | 0.0328 |
| Wine       |               |               |               |        |        |
| Mean       | 0.7586        | 0.8216        | <b>0.8336</b> | 0.7586 | 0.8216 |
| Std        | 0.0064        | 0.0083        | 0.0053        | 0.0164 | 0.0078 |
| Heart      |               |               |               |        |        |
| Mean       | 0.3438        | <b>0.3446</b> | 0.2671        | 0.3268 | 0.3095 |
| Std        | 0.0019        | 0.0105        | 0.0090        | 0.0032 | 0.0075 |
| Australian |               |               |               |        |        |
| Mean       | <b>0.3891</b> | 0.3432        | 0.3228        | 0.3829 | 0.3094 |
| Std        | 0.0020        | 0.0105        | 0.0157        | 0.0039 | 0.0047 |
| Blood      |               |               |               |        |        |
| Mean       | 0.8587        | <b>0.8654</b> | 0.8287        | 0.8587 | 0.8627 |
| Std        | 0.0024        | 0.0104        | 0.0048        | 0.0018 | 0.0075 |
| Pid        |               |               |               |        |        |
| Mean       | 0.7798        | <b>0.8080</b> | 0.7608        | 0.7770 | 0.7793 |
| Std        | 0.0016        | 0.0174        | 0.0138        | 0.0021 | 0.0212 |

TABLE V. CLUSTERING RESULTS IN TERMS OF RAND INDEX.

| Dataset    | FWSCA         | EWSCA         | FCA           | FWSC   | EWSC   |
|------------|---------------|---------------|---------------|--------|--------|
| Iris       |               |               |               |        |        |
| Mean       | 0.9495        | <b>0.9575</b> | 0.8797        | 0.9318 | 0.9132 |
| Std        | 0.0023        | 0.0015        | 0.0026        | 0.0795 | 0.0605 |
| Wine       |               |               |               |        |        |
| Mean       | 0.8987        | 0.9265        | <b>0.9331</b> | 0.8987 | 0.9215 |
| Std        | 0.0013        | 0.0026        | 0.0034        | 0.0278 | 0.0196 |
| Heart      |               |               |               |        |        |
| Mean       | 0.7012        | <b>0.7102</b> | 0.6717        | 0.7012 | 0.7063 |
| Std        | 0.0024        | 0.0110        | 0.0062        | 0.0021 | 0.0036 |
| Australian |               |               |               |        |        |
| Mean       | <b>0.7385</b> | 0.6984        | 0.6810        | 0.7372 | 0.6691 |
| Std        | 0.0137        | 0.0142        | 0.0020        | 0.0010 | 0.0090 |
| Blood      |               |               |               |        |        |
| Mean       | 0.5096        | <b>0.5191</b> | 0.4926        | 0.5096 | 0.5154 |
| Std        | 0.0011        | 0.0164        | 0.0038        | 0.0018 | 0.0224 |
| Pid        |               |               |               |        |        |
| Mean       | 0.5368        | 0.5113        | <b>0.5450</b> | 0.4996 | 0.5242 |
| Std        | 0.0017        | 0.0134        | 0.0014        | 0.0019 | 0.0154 |

TABLE VI. ITERATION NUMBER OF DIFFERENT ALGORITHMS

| Dataset    | FWSCA  | EWSCA | FCA   | FWSC  | EWSC   |
|------------|--------|-------|-------|-------|--------|
| Iris       |        |       |       |       |        |
| Mean       | 36.40  | 46.65 | 77.00 | 39.25 | 25.40  |
| Std        | 7.47   | 12.05 | 2.31  | 19.34 | 2.60   |
| Wine       |        |       |       |       |        |
| Mean       | 42.80  | 63.15 | 74.00 | 29.85 | 24.75  |
| Std        | 2.03   | 8.86  | 2.50  | 3.68  | 3.12   |
| Heart      |        |       |       |       |        |
| Mean       | 87.70  | 56.70 | 98.10 | 98.05 | 60.65  |
| Std        | 6.13   | 24.14 | 4.40  | 3.21  | 9.07   |
| Australian |        |       |       |       |        |
| Mean       | 100.00 | 23.85 | 65.00 | 38.90 | 100.00 |
| Std        | 2.89   | 12.42 | 3.50  | 3.50  | 3.22   |
| Blood      |        |       |       |       |        |
| Mean       | 36.40  | 45.65 | 90.00 | 40.05 | 28.60  |
| Std        | 1.23   | 6.03  | 2.43  | 2.58  | 3.63   |
| Pid        |        |       |       |       |        |
| Mean       | 85.85  | 30.15 | 85.00 | 23.9  | 20.5   |
| Std        | 1.84   | 12.49 | 1.57  | 2.14  | 2.91   |

TABLE VII. CPU RUNNING TIME (SECONDS) OF DIFFERENT ALGORITHMS

| Dataset    | FWSCA  | EWSCA  | FCA    | FWSC   | EWSC   |
|------------|--------|--------|--------|--------|--------|
| Iris       |        |        |        |        |        |
| Mean       | 0.0631 | 0.0928 | 0.0938 | 0.0391 | 0.0245 |
| Std        | 0.0108 | 0.0399 | 0.0078 | 0.0320 | 0.0105 |
| Wine       |        |        |        |        |        |
| Mean       | 0.1051 | 0.1396 | 0.1207 | 0.7326 | 0.0245 |
| Std        | 0.0267 | 0.0246 | 0.0290 | 0.0533 | 0.0044 |
| Heart      |        |        |        |        |        |
| Mean       | 0.2183 | 0.1646 | 0.1714 | 0.0930 | 0.0634 |
| Std        | 0.0304 | 0.0610 | 0.0287 | 0.0178 | 0.0223 |
| Australian |        |        |        |        |        |
| Mean       | 0.4012 | 0.1535 | 0.2029 | 0.1573 | 0.1587 |
| Std        | 0.0346 | 0.0572 | 0.0185 | 0.0243 | 0.0360 |
| Blood      |        |        |        |        |        |
| Mean       | 0.1057 | 0.1447 | 0.1711 | 0.0424 | 0.0339 |
| Std        | 0.0196 | 0.0306 | 0.0329 | 0.0120 | 0.0132 |
| Pid        |        |        |        |        |        |
| Mean       | 0.3042 | 0.1908 | 0.2063 | 0.0353 | 0.0290 |
| Std        | 0.0296 | 0.0684 | 0.0385 | 0.0135 | 0.0094 |

EWSCA are insensitive to the initialization of the clusters, and can avoid trapping into the local minimum. Furthermore, although both FWSCA and EWSCA assign a weight to each dimension in the clustering process to measure the contribution of the dimension in forming a particular cluster, the experiment results show that EWSCA is better than FWSCA in most case. We attempt to merge the results of both algorithms and see if new insights do emerge in the near future.

## V. CONCLUSION

Clustering algorithms face two typical challenges: the unknown number of clusters and the initialization of prototypes in the clustering. This paper has proposed two algorithms to address these problems. First, two novel objective functions have been proposed by extending the fuzzy and entropy subspace clustering objective functions, and integrating the theory of competitive agglomeration. Second, the two soft subspace clustering algorithms with competitive

agglomeration, namely FWSCA and EWSCA, have been developed. Extensive experiments on artificial and UCI data sets have shown that FWSCA and EWSCA outperform the classic algorithms FCA, FWSC and EWSC in terms of identifying the number of clusters and the initialization of prototypes in the clustering.

While FWSCA and EWSCA hold great potential for soft subspace clustering, we are further enhancing them, for instance, to improve detection of the true number of clusters and the convergence speed, and to conduct a theoretical study on the sensitive parameters setting of FWSCA and EWSCA algorithms.

## REFERENCES

- [1] D. T. Anderson, J. C. Bezdek, M. Popescu, and J. M. Keller, "Comparing Fuzzy, Probabilistic, and Possibilistic Partitions," *IEEE Transactions on Fuzzy Systems*, vol. 18, pp. 906-918, 2010.
- [2] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, pp. 651-666, 2010.
- [3] E. Muller, S. Gunnemann, I. Assent, and T. Seidl, "Evaluating clustering in subspace projections of high dimensional data," *VLDB 2009*, pp. 1270-1281.
- [4] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data," *Data Mining and Knowledge Discovery*, vol. 11, pp. 5-33, 2005.
- [5] P. Lance, H. Ehteshami, and L. Huan, "Subspace clustering for high dimensional data: a review," *SIGKDD Explor. Newsl.*, vol. 6, pp. 90-105, 2004.
- [6] C. Chun-Hung, F. Ada Waichee, and Z. Yi, "Entropy-based subspace clustering for mining numerical data," *SIGKDD 1999*, pp. 84-93.
- [7] C. C. Aggarwal and P. S. Yu, "Finding generalized projected clusters in high dimensional spaces," *SIGMOD 2000*, pp. 70-81.
- [8] C. Domeniconi, D. Papadopoulos, D. Gunopulos, and S. Ma, "Subspace clustering of high dimensional data," *SDM 2004*, pp. 517-522.
- [9] Z. H. Deng, K. S. Choi, F. L. Chung, and S. T. Wang, "Enhanced soft subspace clustering integrating within-cluster and between-cluster information," *Pattern Recognition*, vol. 43, pp. 767-781, 2010.
- [10] G. Gan and J. Wu, "A convergence theorem for the fuzzy subspace clustering (FSC) algorithm," *Pattern Recognition*, vol. 41, pp. 1939-1947, 2008.
- [11] L. Jing, M. K. Ng, and J. Z. Huang, "An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data," *IEEE Transactions on knowledge and data engineering*, pp. 1026-1041, 2007.
- [12] E. Y. Chan, W. K. Ching, M. K. Ng, and J. Z. Huang, "An optimization algorithm for clustering using weighted dissimilarity measures," *Pattern recognition*, vol. 37, pp. 943-952, 2004.
- [13] C. Domeniconi, D. Gunopulos, S. Ma, B. Yan, M. Al-Razgan, and D. Papadopoulos, "Locally adaptive metrics for clustering high dimensional data," *Data Mining and Knowledge Discovery*, vol. 14, pp. 63-97, 2007.
- [14] H. Frigui and R. Krishnapuram, "Clustering by competitive agglomeration," *Pattern Recognition*, vol. 30, pp. 1109-1119, 1997.
- [15] D. Zhang, S. Chen, and Z. H. Zhou, "Entropy-Inspired Competitive Clustering Algorithms," *Int J Software Informatics*, vol. 1, pp. 67-84, 2007.
- [16] N. Gira, M. Crucianu, and N. Boujmaa, "Semi-supervised fuzzy clustering with pairwise-constrained competitive agglomeration," *FUZZ-IEEE 2005*, pp. 867-872.
- [17] C. C. Chuang, C. W. Tao, and J. T. Jeng, "Robust clustering algorithm for the symbolic interval-values data with outliers," *FUZZ-IEEE 2009*, pp. 1232-1237.
- [18] A. Strehl and J. Ghosh, "Cluster ensembles - a knowledge reuse framework for combining multiple partitions," *The Journal of Machine Learning Research*, vol. 3, pp. 583-617, 2003.
- [19] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical association*, vol. 66, pp. 846-850, 1971.
- [20] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," 1998.