An Improved Semi-supervised Fuzzy Clustering Algorithm

Kai Li
School of Mathematics and Computer
Hebei University
Baoding, China
e-mail: likai {at} hbu.edu.cn

Yufei Zhou
School of Mathematics and Computer
Hebei University
Baoding, China

Abstract—**Semi-supervised clustering is an important method which can improve clustering performance by introducing partial supervised information. This paper mainly studies the semi-supervised fuzzy clustering based on Mahalanobis distance and Gaussian Kernel for SCAPC algorithm. Here, we give a new semi-supervised fuzzy clustering objective function. By solving the optimization problem with above objective function, we obtain a semi-supervised fuzzy clustering algorithm F-SCAPC which includes F(M)-SCAPC and F(K)-SCAPC. And we do experimental research for proposed algorithm F-SCAPC using the selected standard data set and the artificial data set. Besides, we compare performance of presented algorithm F-SCAPC with one of FCM, CA, AFFC, KCA, KFCM-F and SCAPC algorithms. From the results, we can see that F-SCAPC is effective in the convergence speed and the clustering accuracy.**

Keywords-Semi-supervised clustering; Pairwise constraints; Mahalanobis distance; Gaussian Kernel

I. INTRODUCTION

Semi-supervised clustering is an important research direction for semi-supervised learning. It mainly uses a small amount of supervision information and a large number of unlabeled samples to learn. In general, the supervision information is given by two kinds of the different forms: one is the pairwise constraints, such as must-link and cannot-link; the other one is directly given by the few labeled samples. Currently, researchers have proposed many different semi-supervised clustering algorithms for two kinds of the supervision information. These methods can be roughly divided into two categories: 1) Introducing the supervision information to the existing clustering algorithm in order to obtain a semi-supervised clustering algorithm. 2) Using the supervision information to learn some metric [1-3]. For example, Bar-Hillel et al. [2] used must-link constraints to give a non-iterative method RCA by learning Mahalanobis metric. After that, Yeung et al. [4] made an extension for RCA method by introducing both must-link constraints and cannot-link constraints. In addition, kernel-based learning is also an important area of research in machine learning and choosing a good kernel function can further improve the classification accuracy. However, how to choose better parameter value for kernel function becomes an important problem. After that, non-parametric kernel learning is presented and Cristianini et al. [5] studied the non-parametric kernel learning. Yang et al. [6] discussed the kernel learning with metric strategy. Sindwani et al. [7] proposed a semi-supervised kernel learning framework which modifies the existing kernel function by unlabeled data to determine the kernel function. Zhang et al.[8] researched FCM algorithm for different kernel-based learning methods and proposed fuzzy clustering algorithm KFCM based on kernel learning. Graves et al. [9] used different kernel functions to map the high-dimensional nonlinear data to the linear data and proposed KFCM-K and KFCM-F based on FCM algorithm. Baghshah [10] proposed a nonlinear kernel matrix metric learning algorithm for semi-supervised learning. It can be seen that for most clustering algorithms, which include unsupervised clustering and semi-supervised clustering, the number of clusters should be artificially given before the clustering process. But in practical applications, it is more difficult to obtain number of clusters in advance. For solving this problem, Frigui et al. [11] proposed CA algorithm, which automatically calculates the appropriate number of clusters by the competition. It is regretful that the accuracy of the clustering results for this method is low. In order to solve this problem, Grira et al. [12,13] proposed the AFCC algorithm by combining the semi-supervised clustering with the CA algorithm. In 2010, Gao et al.[14] further proposed SCAPC algorithm by modifying the objective function in AFCC algorithm. It can be seen that both AFCC and SCAPC algorithm are designed based on the Euclidean distance which mainly applies to spherical data. For non-spherical, high-dimensional nonlinear and the overlapping data, clustering results are not very good using these algorithms. In order to solve these problems, we study semi-supervised fuzzy clustering by introducing Mahalanobis metric and Gaussian kernel into SCAPC's objective function. And we obtain a new objective function and present semi-supervised fuzzy clustering algorithm F-SCAPC.

II. SEMI-SUPERVISED FUZZY CLUSTERING ALGORITHM F-SCAPC

Currently, most algorithms mainly use the Euclidean distance to compute similarity between two samples. Surely, the Euclidean distance has better results during processing spherical data. However, for the high-dimensional, non-spherical, nonlinear or the overlapping data, the error rate will be increased and clustering speed is also very slower. We know that Mahalanobis metric has good properties. In addition, kernel trick can make complex and nonlinear problem to become linearly separable problem in the feature space after nonlinear transform. For better studying semi-supervised fuzzy clustering, we first propose a generalized objective function.

Given data set of N samples X={x$_i$|i ∈ {1,…,N}}, it is divides into C clusters S1,S2,…,S$_C$. Let *V*={v$_k$|k∈{1,…,C}} to represent set of the cluster centers and *U*=(u_{ik}) is a matrix of membership degree, where u_{ik} is the membership of the elements x_i to the cluster center v$_k$. The generalized objective function of algorithm F-SCAPC is written as follows:

$$\begin{aligned} J_{F-SCAPC} &= \sum_{K=1}^{C}\sum_{i=1}^{N}\left(u_{ik}\right)^2 f^2\left(x_i,v_k\right) \\ &+\alpha\left(\sum_{(x_i,x_j)\in \mathrm{M}}\ \sum_{k=1,l=1}^{C}\ \sum_{l\neq k}^{C} u_{ik} f\left(x_i,v_k\right) u_{jl} f\left(x_j,v_l\right)\right) \\ &+\alpha\left(\sum_{(x_i,x_j)\in \mathrm{C}}\ \sum_{k=1}^{C} u_{ik} f\left(x_i,v_k\right) u_{jk} f\left(x_j,v_k\right)\right)-\beta\sum_{k=1}^{C}\left[\sum_{i=1}^{N}u_{ik}\right]^2 \end{aligned} \quad . \ (1)$$

For Mahalanobis metric, $f^2\left(x_i,v_k\right)$ is denoted as $f_m^{\ 2}\left(x_i,v_k\right)$,where $f_m^{\ 2}\left(x_i,v_k\right)=\left(x_i-v_k\right)^T C_k^{-1}\left(x_i-v_k\right)$, C_k is the covariance matrix, namely $C_k=\sum_{i=1}^{N}\left(u_{ik}\right)^2\left(x_i-v_k\right)^T\left(x_i-v_k\right)\Big/\sum_{i=1}^{N}\left(u_{ik}\right)^2$. For the Gaussian Kernel, $f^2\left(x_i,v_k\right)$ is written as $f_k^{\ 2}\left(x_i,v_k\right)$, where

$$\begin{aligned} f_k^{\ 2}\left(x_i,v_k\right) &= \left(\varphi\left(x_i\right)-\varphi\left(v_k\right)\right)^2 \\ &= \varphi\left(x_i\right)^2-2\varphi\left(x_i\right)^T\varphi\left(v_k\right)+\varphi\left(v_k\right)^2 \ , \\ &= K\left(x_i,x_i\right)+K\left(v_k,v_k\right)-2K\left(x_i,v_k\right) \end{aligned}$$

φ is a mapping.

It is seen that when Gaussian kernel is used, $K\left(x,x\right)=1$. So $f_k^{\ 2}\left(x_i,v_k\right)=2-2K\left(x_i,v_k\right)$, where $K\left(x_i,v_k\right)=e^{-(x_i-v_k)^2/\sigma^2}$.

In order to distinguish between above two methods, we denote by F(M)-SCAPC and F(K)-SCAPC using Mahalanobis metric and kernel function, respectively.

Therefore, the improved semi-supervised clustering algorithm is attributed to following optimization problem:

$$\begin{aligned} &\min_{U,V} J_{F-SCAPC} \\ &s.t. \sum_{k=1}^{C} u_{ik}=1 \quad i\in\left\{1,2,\ldots,N\right\} \ . \\ &\quad u_{ik}\geq 0, \quad 1\leq k\leq C \end{aligned} \qquad (2)$$

In the following, we first consider semi-supervised clustering using Mahalanobis metric. By minimizing (2) with respect to λ$_i$ and applying Lagrange multipliers, we obtain the following Lagrangian function:

$$\begin{aligned} J\left(V,U,\lambda\right) &= \sum_{K=1}^{C}\sum_{i=1}^{N}\left(u_{ik}\right)^2 f_m^{\ 2}\left(x_i,v_k\right) \\ &+\ \alpha\left(\sum_{(x_i,x_j)\in \mathrm{M}}\ \sum_{k=1,l=1}^{C}\ \sum_{l\neq k}^{C} u_{ik} f_m\left(x_i,v_k\right) u_{jl} f_m\left(x_j,v_l\right)\right) \\ &+\alpha\left(\sum_{(x_i,x_j)\in \mathrm{C}}\ \sum_{k=1}^{C} u_{ik} f_m\left(x_i,v_k\right) u_{jk} f_m\left(x_j,v_k\right)\right) \\ &-\beta\sum_{k=1}^{C}\left[\sum_{i=1}^{N}u_{ik}\right]^2-\sum_{k=1}^{N}\lambda_i\left(\sum_{i=1}^{N}u_{ik}-1\right) \end{aligned} \qquad (3)$$

For i ∈ {1,…,N} and k∈{1,…C}, with respect to u$_{rs}$ and λ$_i$, setting $\frac{\partial J(V,U,\lambda)}{\partial\lambda_r}=0$ and $\frac{\partial J(V,U,\lambda)}{\partial u_{rs}}=0$,

namely

$$\frac{\partial J(V,U,\lambda)}{\partial\lambda_r}=\sum_{k=1}^{C}u_{rk}-1=0, \qquad (4)$$

$$\begin{aligned} &2u_{rs} f_m^{\ 2}(x_r,v_s)-2\beta\sum_{i=1}^{N}u_{is}-\lambda_r \\ &+\alpha\sum_{(x_r,x_j)\in \mathrm{M}}\ \sum_{l=1,l\neq s}^{C} f_m(x_r,v_s)u_{jl} f_m(x_j,v_l) \\ &+\alpha\sum_{(x_i,x_j)\in \mathrm{C}} f_m(x_r,v_s)u_{js} f_m(x_j,v_s)=0 \end{aligned} \quad . \qquad (5)$$

We know that $N_k=\sum_{i=1}^{N}u_{ik}$ is the cardinality of cluster k. Using N_k and computing u$_{rs}$ by (5), we have

$$\begin{aligned} u_{rs} &= \frac{2\beta N_s+\lambda_r}{2 f_m^{\ 2}\left(x_r,v_s\right)} \\ &-\alpha\frac{\sum_{(x_r,x_j)\in \mathrm{M}}\ \sum_{l=1,l\neq s}^{C} f_m\left(x_r,v_s\right)u_{jl} f_m\left(x_j,v_j\right)+\sum_{(x_r,x_j)\in \mathrm{C}} f_m\left(x_r,v_s\right)u_{js} f_m\left(x_j,v_s\right)}{2 f_m^{\ 2}\left(x_r,v_s\right)} \end{aligned} \quad . \ (6)$$

Considering the constraint $\sum_{k=1}^{C}u_{ik}=1$, i∈ {1,…,N} and replacing (4) with (6), we have the following equation:

$$\begin{aligned} &\sum_{k=1}^{C}\frac{2\beta N_k+\lambda_r}{2 f_m^{\ 2}\left(x_r,v_k\right)} \\ &-\sum_{k=1}^{C}\alpha\frac{\sum_{(x_r,x_j)\in \mathrm{M}}\ \sum_{l=1,l\neq k}^{C} f_m\left(x_r,v_k\right)u_{jl} f_m\left(x_j,v_l\right)}{2 f_m^{\ 2}\left(x_r,v_k\right)} \quad , \\ &-\sum_{k=1}^{C}\alpha\frac{\sum_{(x_r,x_j)\in \mathrm{C}} f_m\left(x_r,v_k\right)u_{jk} f_m\left(x_j,v_k\right)}{2 f_m^{\ 2}\left(x_r,v_k\right)}=1 \end{aligned}$$

namely

$$\beta\sum_{k=1}^{C}\frac{N_k}{f_m\left(x_r,v_k\right)}+\lambda_r\sum_{k=1}^{C}\frac{1}{2f_m\left(x_r,v_k\right)}$$
$$-\alpha\sum_{k=1}^{C}\frac{\sum\limits_{(x_r,x_j)\in M}\sum\limits_{l=1,l\neq k}^{C}f_m\left(x_r,v_k\right)u_{jl}f_m\left(x_j,v_l\right)}{2f_m\left(x_r,v_k\right)}. \qquad (7)$$
$$-\alpha\sum_{k=1}^{C}\frac{\sum\limits_{(x_r,x_j)\in C}f_m\left(x_r,v_k\right)u_{jk}f_m\left(x_j,v_k\right)}{2f_m\left(x_r,v_k\right)}$$
$$=1$$

By (7), we obtain the following expression λ_r:

$$\lambda_r=\frac{1-\beta\sum\limits_{k=1}^{C}\left(\frac{N_k}{f_m^2\left(x_r,v_k\right)}\right)}{\sum\limits_{k=1}^{C}\frac{1}{2f_m^2\left(x_r,v_k\right)}}$$
$$+\alpha\frac{\sum\limits_{k=1}^{C}\left(\frac{\sum\limits_{(x_r,x_j)\in M}\sum\limits_{l=1,l\neq k}^{C}f_m\left(x_r,v_k\right)u_{jl}f_m\left(x_j,v_l\right)+\sum\limits_{(x_r,x_j)\in C}f_m\left(x_r,v_k\right)u_{jk}f_m\left(x_j,v_k\right)}{2f_m^2\left(x_r,v_k\right)}\right)}{\sum\limits_{k=1}^{C}\frac{1}{2f_m^2\left(x_r,v_k\right)}}$$

By substituting the right side with (6) for λ_r , we obtain new equation for the membership u_{rs}, namely

$$u_{rs}=\frac{\frac{1}{f_m^2\left(x_r,v_s\right)}}{\sum\limits_{k=1}^{C}\frac{1}{f_m^2\left(x_r,v_k\right)}}+\frac{\alpha}{2f_m^2\left(x_r,v_s\right)}\overline{C_{v_r}} \quad .$$
$$-\frac{\alpha}{2f_m^2\left(x_r,v_s\right)}C_{v_{rs}}+\frac{\beta}{f_m^2\left(x_r,v_s\right)}\left(N_s-\overline{N_r}\right)$$

Now, we express membership u_{rs} as u_m, namely

$$u_m=u_{mF}+u_{mC}+u_{mB}, \qquad (8)$$

where

$$u_{mF}=\frac{\frac{1}{f_m^{\,2}\left(x_r,v_s\right)}}{\sum\limits_{k=1}^{C}\frac{1}{f_m^{\,2}\left(x_r,v_k\right)}}\quad,\quad u_{mC}=\frac{\alpha}{2f_m^2\left(x_r,v_s\right)}\left(\overline{C_{v_r}}-C_{v_{rs}}\right)\quad,$$
$$u_{mB}=\frac{\beta}{f_m^2\left(x_r,v_s\right)}\left(N_s-\overline{N_r}\right)\ ,$$

$$C_{v_{rs}}=\sum_{(x_r,x_j)\in M}\sum_{l=1,l\neq s}^{C}f_m\left(x_r,v_s\right)u_{jl}f_m\left(x_j,v_l\right)$$
$$+\sum_{(x_r,x_j)\in C}f_m\left(x_r,v_s\right)u_{js}f_m\left(x_j,v_s\right) \quad ,$$

$$\overline{C_{v_r}}=\frac{\sum\limits_{k=1}^{C}\frac{\sum\limits_{(x_r,x_j)\in M}\sum\limits_{l=1,l\neq k}^{C}f_m\left(x_r,v_k\right)u_{jl}f_m\left(x_j,v_l\right)}{f_m^2\left(x_r,v_k\right)}}{\sum\limits_{k=1}^{C}\frac{1}{f_m^2\left(x_r,v_k\right)}} \quad ,$$
$$+\frac{\sum\limits_{k=1}^{C}\frac{\sum\limits_{(x_r,x_j)\in C}f_m\left(x_r,v_k\right)u_{jk}f_m\left(x_j,v_k\right)}{f_m^2\left(x_r,v_k\right)}}{\sum\limits_{k=1}^{C}\frac{1}{f_m^2\left(x_r,v_k\right)}}$$

and $\overline{N_r}=\dfrac{\sum\limits_{k=1}^{C}\frac{N_k}{f_m^2\left(x_r,v_k\right)}}{\sum\limits_{k=1}^{C}\frac{1}{f_m^2\left(x_r,v_k\right)}}$.

Similarly, we can get the (9) when *f (x)* takes Gaussian kernel.

$$u_k=u_{kF}+u_{kC}+u_{kB}\ , \qquad (9)$$

Where

$$u_{kF}=\frac{\frac{1}{1-f_k^{\,2}\left(x_r,v_s\right)}}{\sum\limits_{k=1}^{C}\frac{1}{1-f_k^{\,2}\left(x_r,v_k\right)}}\quad,\quad u_{kC}=\frac{\alpha}{2\left(1-f_k^{\,2}\left(x_r,v_s\right)\right)}\left(\overline{C_{v_r}}-C_{v_{rs}}\right)\quad,$$
$$u_{kB}=\frac{\beta}{1-f_k^{\,2}\left(x_r,v_s\right)}\left(N_s-\overline{N_r}\right),$$

$$C_{v_{rs}}=\sum_{(x_r,x_j)\in M}\sum_{l=1,l\neq s}^{C}f_k\left(x_r,v_s\right)u_{jl}f_k\left(x_j,v_l\right)+\sum_{(x_r,x_j)\in C}f_k\left(x_r,v_s\right)u_{js}f_k\left(x_j,v_s\right)$$

$$\overline{C_{v_r}}=\frac{\sum\limits_{k=1}^{C}\frac{\sum\limits_{(x_r,x_j)\in M}\sum\limits_{l=1,l\neq k}^{C}f_k\left(x_r,v_k\right)u_{jl}f_k\left(x_j,v_l\right)+\sum\limits_{(x_r,x_j)\in C}f_k\left(x_r,v_k\right)u_{jk}f_k\left(x_j,v_k\right)}{1-f_k^{\,2}\left(x_r,v_k\right)}}{\sum\limits_{k=1}^{C}\frac{1}{1-f_k^{\,2}\left(x_r,v_k\right)}}$$

, and $\overline{N_r}=\dfrac{\sum\limits_{k=1}^{C}\frac{N_k}{1-f_k^{\,2}\left(x_r,v_k\right)}}{\sum\limits_{k=1}^{C}\frac{1}{1-f_k^{\,2}\left(x_r,v_k\right)}}$.

In above objective function, parameter α and β are adjustment factors of the clusters which are computed by (10) and (11) , respectively[13]:

$$\beta(t)=\frac{\eta_0\exp(-|t-t_0|/\tau)}{\sum\limits_{k=1}^{C}\left[\sum\limits_{i=1}^{N}\left(u_{ik}\right)\right]^2}\left[\sum_{k=1}^{C}\sum_{i=1}^{N}\left(u_{ik}\right)^2\ f^{\,2}\left(x_i,v_k\right)\right], \qquad (10)$$

$$\alpha = \frac{N}{M}\frac{\sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})^2 f^2(x_i, v_k)}{\sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})^2}. \tag{11}$$

In order to obtain the cluster centers, we take Mahalanobis distance into the object function $J_{F-SCAPC}$ and get thefollowing expression:

$$\begin{aligned}J_{F(M)-SCAPC} &= \sum_{K=1}^{C}\sum_{i=1}^{N}(u_{ik})^2\sum_{m=1}^{n}(x_{im}-v_{km})^T C_k^{-1}(x_{im}-v_{km})\\
&+\alpha\left(\sum_{(x_i,x_j)\in \mathrm{M}}\sum_{k=1,l=1}^{C}\sum_{l\neq k}^{C}u_{ik}\sqrt{\sum_{m=1}^{n}(x_{im}-v_{km})^T C_k^{-1}(x_{im}-v_{km})}u_{jl}\sqrt{\sum_{m=1}^{n}(x_{jm}-v_{lm})^T C_k^{-1}(x_{jm}-v_{lm})}\right)\\
&+\alpha\left(\sum_{(x_i,x_j)\in \mathrm{C}}\sum_{k=1}^{C}u_{ik}\sqrt{\sum_{m=1}^{n}(x_{im}-v_{km})^T C_k^{-1}(x_{im}-v_{km})}u_{jk}\sqrt{\sum_{m=1}^{n}(x_{jm}-v_{km})^T C_k^{-1}(x_{jm}-v_{km})}\right)\\
&-\beta\sum_{k=1}^{C}\left[\sum_{i=1}^{N}u_{ik}\right]^2\end{aligned}$$

Setting $\frac{\partial J_{F-SCAPC}}{\partial v_{km}} = 0$,we obtain following expression:

$$\begin{aligned}\frac{\partial J_{F(M)-SCAPC}}{\partial v_{km}} &= 2\sum_{i=1}^{N}(u_{ik})^2 C_k^{-1}(x_{im}-v_{km})\\
&+\alpha\left(\sum_{(x_i,x_j)\in \mathrm{M}}\sum_{k=1,l=1}^{C}\sum_{l\neq k}^{C}u_{ik}\frac{\sqrt{C_k^{-1}}(x_{im}-v_{km})}{\sqrt{(x_{im}-v_{km})^2}}u_{jl}\sqrt{\sum_{m=1}^{n}(x_{jm}-v_{lm})^2 C_k^{-1}}\right)\\
&+\alpha\sum_{(x_i,x_j)\in \mathrm{C}}\sum_{k=1}^{C}u_{ik}\frac{\sqrt{C_k^{-1}}(x_{im}-v_{km})}{\sqrt{(x_{im}-v_{km})^2}}u_{jk}\sqrt{C_k^{-1}(x_{jm}-v_{km})^2}\\
&+\alpha u_{ik}\sqrt{C_k^{-1}(x_{im}-v_{km})^2}u_{jk}\frac{\sqrt{C_k^{-1}}(x_{jm}-v_{km})}{\sqrt{(x_{jm}-v_{km})^2}} = 0\end{aligned}$$

.

Letting $\xi_{ik} = \frac{(x_{im}-v_{km})}{\sqrt{(x_{im}-v_{km})^2}}$. It can be seen that and if $x_{im}>v_{km}$ then ξ_{ik} =1 else ξ_{ik} =-1. So cluster center v_{km} is obtained by solving the above equation:

$$\begin{aligned}v_{km} &= \frac{\sum_{i=1}^{N}(u_{ik})^2 C_k^{-1}x_{im} + (\alpha/2)\left(\sum_{(x_i,x_j)\in \mathrm{M}}\sum_{k=1,l=1,l\neq k}^{C}C_k^{-1}u_{ik}u_{jl}\xi_{ik}\,|\,x_{jm}-v_{lm}\,|\right)}{\sum_{i=1}^{N}(u_{ik})^2+\alpha\sum_{(x_i,x_j)\in \mathrm{C}}\xi_{ik}\xi_{jk}u_{ik}u_{jk}}.\\
&+\frac{(\alpha/2)\left(\sum_{(x_i,x_j)\in \mathrm{C}}\xi_{ik}\xi_{jk}C_k^{-1}u_{ik}u_{jk}(x_{im}+x_{jm})\right)}{\sum_{i=1}^{N}(u_{ik})^2+\alpha\sum_{(x_i,x_j)\in \mathrm{C}}\xi_{ik}\xi_{jk}u_{ik}u_{jk}}\end{aligned} \tag{12}$$

Similarly, we can get the cluster center v_k when $f\ (x)$ is Gaussian kernel.

$$\begin{aligned}\frac{\partial J_{F(K)-SCAPC}}{\partial v_k} &= 2\sum_{i=1}^{N}(u_{ik})^2 K(x_i, v_k)\frac{2(x_{ip}-v_{kp})}{\sigma^2}\\
&+\alpha\left(\sum_{(x_i,x_j)\in \mathrm{M}}\sum_{k=1,l=1}^{C}\sum_{l\neq k}^{C}u_{ik}\frac{K(x_i,v_k)}{\sqrt{K(x_i,v_k)}}\frac{2(x_{ip}-v_{kp})}{\sigma^2}u_{jl}\sqrt{K(x_{jp},v_{lp})}\right),\\
&+\alpha\sum_{(x_i,x_j)\in \mathrm{C}}\sum_{k=1}^{C}u_{ik}\frac{K(x_i,v_k)}{\sqrt{K(x_i,v_k)}}\frac{2(x_{ip}-v_{kp})}{\sigma^2}u_{jk}\sqrt{K(x_{jp},v_{kp})}\\
&+\alpha u_{ik}\sqrt{K(x_{xp},v_{kp})}u_{jk}\frac{K(x_i,v_k)}{\sqrt{K(x_i,v_k)}}\frac{2(x_{jp}-v_{kp})}{\sigma^2} = 0\end{aligned}$$

$$\begin{aligned}v_k &= \frac{\sum_{i=1}^{N}(u_{ik})^2 K(x_i,v_k)x_{ip} + (\alpha/2)\left(\sum_{(x_i,x_j)\in \mathrm{M}}\sum_{k=1,l=1,l\neq k}^{C}u_{ik}u_{jl}K(x_i,v_k)(x_{ip}-v_{kp})\right)}{\sum_{i=1}^{N}(u_{ik})^2 K(x_i,v_k)+\alpha\sum_{(x_i,x_j)\in \mathrm{C}}u_{ik}u_{jk}K(x_i,v_k)}\\
&+\frac{(\alpha/2)\left(\sum_{(x_i,x_j)\in \mathrm{C}}u_{ik}u_{jk}K(x_j,v_k)(x_{ip}+x_{jp})\right)}{\sum_{i=1}^{N}(u_{ik})^2 K(x_i,v_k)+\alpha\sum_{(x_i,x_j)\in \mathrm{C}}u_{ik}u_{jk}K(x_i,v_k)}\end{aligned} \tag{13}$$

In the following, we give the detailed algorithm F-SCAPC.

(1) Initialize the maximum number of clusters C, threshold e and ε.

(2) Choose the cluster center and the membership.

(3) Calculate covariance matrix C_k or Gaussian kernel $K(x_i, v_k) = e^{-(x_i-v_k)^2/\sigma^2}$.

(4) Calculate f_k or f_m.

(5) Compute β using (10) and α using (11).

(6) Compute cardinalities $N_k = \sum_{i=1}^{N}u_{ik}\ ,k\in(1,2,\cdots,C)$, if N_k < e then discarding cluster k and updating number of clusters C .

(7) Compute the new membership u_m or u_k using (8) or Eq. (9), and update cluster centers v_{km} or v_k using (12) or (13).

(8) Repeat step 3 to step 7 until the iteration terminal (Two adjacent loop have the same number of clusters and $J_{F-SCAPC}(t) - J_{F-SCAPC}(t+1) < \varepsilon$).

III. EXPERIMENTAL RESULTS AND ANALYSIS

In order to verify the effectiveness of algorithm, we use UCI datasets to test performance with F -SCAPC which includes F(M)-SCAPC and F(K)-SCAPC. The selected data sets are Iris, Diabetes, Breast and Wine from UCI database, respectively. In the meantime, we also generate artificial data set to conduct the experiment. In experiments, we set the cardinalities' threshold e=7 and ε=0.001. And we select 10 constraints from the sample set. According to the known labeled samples, sets of Must-link and Cannot-link are created.

A. Parameter Analysis With Algorithm F(M)-SCAPC

1) Relation between parameters (include α and β) and number of iteration.

According to the objective function (3), parameter α (Alpha) is a balance factor of the F(M)-SCAPC algorithm. It reflects the importance of constraint items and directly affects the adjustment size of the penalty term of the objective function. For iris data set, we conduct the experimental study with parameter α whose results are shown in Fig 1. It may be seen that as increasing with the number of iteration, the value of α first increases and then decreases. Especially, in the early of the iteration, since constraint penalty term is to adjust the membership, classification is not very clear when α increases. And then α gradually stabilizes and the objective function does not basically changes.

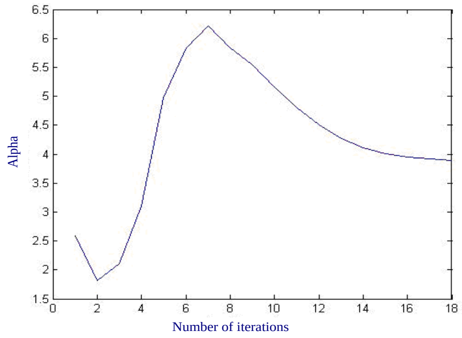


Figure 1. Relation between value of α and number of iteration

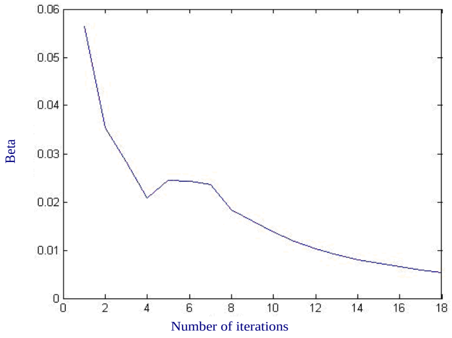


Figure 2. Relation between value of β and number of iteration

Similarly, we also conduct the experiment with β(Beta). Experimental result is given in Fig 2. According to Fig 2, when increasing with the number of iterations, the value of β is gradually decreases. And there are some fluctuations in the middle. In the initial iteration, the value of β is relatively large and the classification is not stable. But as the number of iterations increases, both the number of clusters and the value of β also gradually stabilize.

Moreover, β depends on the selected value of η_0 which will affect the clustering results at some extent and the number of iterations. In Table 1, the experimental results are given.

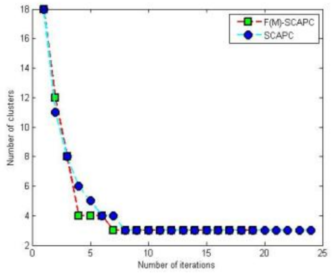
TABLE I. INFLUENCE OF η_0 ON THE NUMBER OF CLUSTERS AND THE NUMBER OF ITERATIONS

η0	No. of clusters	No. of iterations
1	16	21
1.5	14	21
2	13	21
4	3	20
4.5	3	19
5	3	17
7	1	16
8	2	16
9	1	16

Seen from Table 1, the value of η0 will affect the number of clusters and the convergence speed of the algorithm. It can be known that the number of correct cluster is 3 for data set Iris, however, when original number of clusters is initialized as 18 and value of η0 ranges from [1,2], the number of clusters for data set Iris is 16 and 13,which shows that it is hard to get the correct number of clusters. The objective function will be stable after 21 iterations in this case. When η0 is in the range [4,5], the final classification number is 3,which shows that the correct number of clusters is obtained and the number of iterations is about 18 times. When η0 is in the range [7,9], the final number of clusters that is smaller than the correct number of cluster is 1 or 2,which shows that the balance factor is too large and the competition is over normal range, where the number of iterations is about 16 times. These show that the bigger is η0, the smaller is the number of iterations. Test results show that a better result will be got when η0 is in the interval [4,5].

2) Comparison of convergence speed in clustering

In this subsection, we mainly compare the convergence speed with different data set. We selected Iris, Diabetes and Wine data set from UCI database. The experimental parameter's values are as follows: The maximum number of cluster Cmax are initialized as 18, 13, and 18, respectively. ε and η0 is set as 0.001 and 4, respectively. Results are shown in Fig 3with the convergence speed of the SCAPC and F(M)-SCAPC on different data set. According to Fig 3, SCAPC algorithm makes the objective function to stabilize after 24 iterations on the Iris data set, while the algorithm F(M)-SCAPC only uses 17 iterations. Moreover, for F(M)-SCAPC algorithm, correct number of clusters is obtained in the 7th iteration, while SCAPC obtains correct number of clusters in the 8th iteration. For Diabetes and Wine data sets, the similar conclusions can be also obtained. This shows that convergence speed of algorithm F(M)-SCAPC is faster than the convergence speed of algorithm SCAPC.



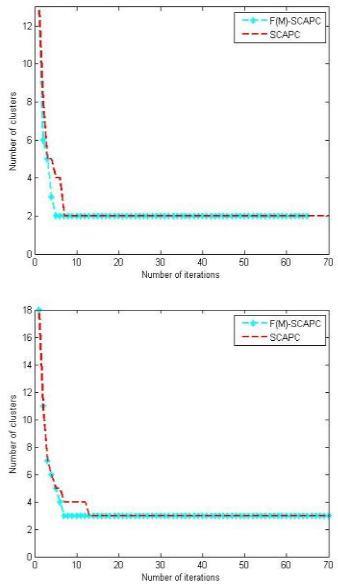


Figure 3. Comparison with convergence speed between SCAPC and F(M)-SCAPC on the Iris, Diabetes and wine dataset

Moreover, to verify performance of the modified algorithm F(M)-SCAPC, we generate an artificial data set Data to test. This dataset is given a three-dimensional data which contains 150 sample points and equally divided the dataset into 3 classes. That is to say, each subset contains 50 sample points. Fig 4 is a representation of the three-dimensional space of the given dataset. In experiment, the initial number of clusters is set as 18 and threshold ε of objective function is set as 0.001. Fig 5 is the clustering convergence speed of this data set which shows that F(M)-SCAPC completes the correct number of classification at the 7th iteration on Data dataset, but SCAPC does in the 9th iteration. This also shows that the convergence speed of F(M)-SCAPC is faster than SCAPC.

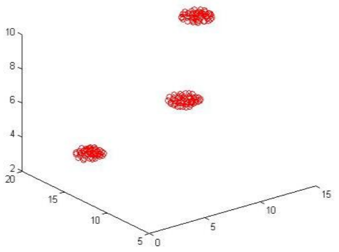


Figure 4. Generated dataset Data

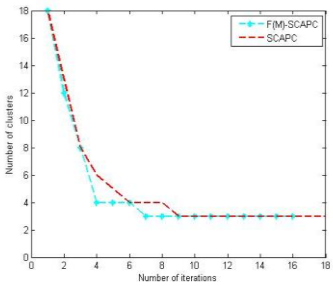


Figure 5. Comparison with convergence speed between SCAPC and F(M)-SCAPC on the Data dataset

3) *Pairwise constraints and clustering performance*

We know by above analysis that if known pairwise data with constraint sets is wrongly classified during the clustering process, the penalty term of objective function will be bigger. Here, the penalty item will be continued to adjust the value of the membership. Thus, by adjusting the penalty term in the objective function, some wrongly clustering samples will be divided into the correct cluster, so the accuracy of clustering result will be higher eventually. To verify above conclusions, for the selected datasets, we test the relation between the given number of constraints and the clustering results. In experiments, η0s are set as 4, 5, 5.5 and 6 with Iris, Diabetes, Wine, Data dataset, respectively. The maximum number of clusters C is initialized as 18 and objective function's threshold ε is set as 0.001. Experimental results are given by Fig 6.

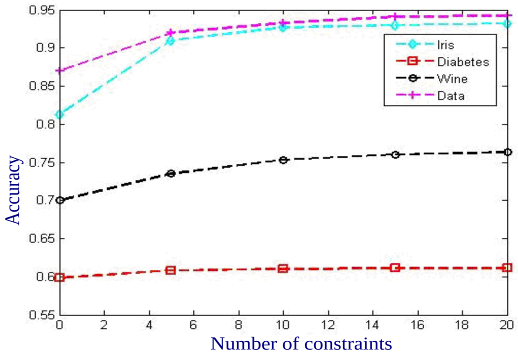


Figure 6. Accuracy with different number of constraints on different dataset

The experiments show that as the number of constraints (must-link and cannot-link) increases, accuracy of F(M)-SCAPC algorithm clustering results will be higher. It also shows that penalty term of the objective function has an impact on adjustment. When the number of constraints is small, accuracy of clustering results is relatively very low. When the number of constraints increases, the accuracy of clustering results have a significant improvement. When the number of constraints reaches a certain number, the curve will be steady gradually where its effect on clustering result will be gradually small.

B. Experimental results with algorithm F(K)-SCAPC

1) Comparison with different algorithms

To validate the effectiveness of presented algorithm F (K)-SCAPC, we selected four data sets from the UCI database as the experimental data objects which are Iris, Diabetes, Wine and Heart. We compare F(M)-SCAPC and F(K)-SCAPC which we proposed with FCM，CA，AFCC，SCAPC，KCA，KFCM-F to the clustering result. And F(M)-SCAPC algorithm is a semi-supervised learning algorithm which based on the Mahalanobis metric, and F (K)-SCAPC is another semi-supervised learning algorithm which based on Gaussian kernel metric. FCM and CA are unsupervised algorithms. AFCC and SCAPC are semi-supervised algorithms based on the Euclidean distance. KCA is an unsupervised clustering algorithm which introduce Gaussian kernel into the CA algorithm. KFCM-F is an unsupervised algorithm which based kernel learning. We use the accuracy of clustering r as an evaluated method which is defined as r= (c/N) ×100%, where c is the correct classification number of points and N is the number of points in the dataset. The value of the parameters we used in the test is as follows: The initial maximum number of clusters Cmax=18. The algorithm terminates threshold value ε=0.001. The cardinality of the threshold value e=7. The number of constraints is different for different data set. In order to obtain a more fair comparison, we performed five times for each algorithm tests which with its best clustering parameters that we have. The clustering results are shown in Table 2 and the parameters we used in the experiment are also list.

From the Table 2 we can see that F-SCAPC has a better performance than the other algorithms in the data sets iris, diabetes, wine and heart. And F (K)-SCAPC algorithm which based on kernel has a good performance to deal with the high

TABLE II. CLUSTERING ACCURACY BY DIFFERENT ALGORITHMS

Data set	**Algorithm**	**Parameter**	**Pair-wise constraints**	**Accuracy（%）**				
				1	***2***	***3***	***4***	***5***
Iris	FCM	—	—	89.3	89.3	89.3	89.3	89.3
	CA	$\eta_{0=4}$	—	72.7	70	81.3	72.7	70.7
	AFCC	$\eta_{0=4}$	5	90.7	90.6	88.7	82.7	90
	SCAPC	$\eta_{0=4}$	5	91.3	91.3	90.7	90.7	91.3
	F(M)-SCAPC	$\eta_{0=4}$	5	92.7	92.7	92.7	91.3	92.7
	KCA	$\eta_{0=4,}$ σ=10	—	82.7	73.3	89.3	82	82.7
	KFCM-F	$\eta_{0=4,}$σ=15	—	90.7	84	89.3	81.3	86
	F(K)-SCAPC	$\eta_{0=4,}$σ=15	5	93.3	94	91.3	92.7	92.7
Diabetes	FCM	—	—	55.9	55.9	55.9	55.9	55.9
	CA	$\eta_{0=4}$	—	55.6	55.6	55.6	55.6	55.6
	AFCC	$\eta_{0=4}$	15	56.6	62.5	56.6	56.6	62.5
	SCAPC	$\eta_{0=4}$	15	58.3	58.3	58.3	58.3	58.3
	F(M)-SCAPC	$\eta_{0=4}$	15	68.8	68.8	68.8	68.8	68.8
	KCA	$\eta_{0=4,}$ σ=25	—	55.9	56.6	56.6	55.9	56.6
	KFCM-F	$\eta_{0=4,}$σ=25	—	65.6	65.4	65.6	52.5	65.8
	F(K)-SCAPC	$\eta_{0=4,}$σ=30	15	68.2	68.2	67.4	69.8	68.2
Wine	FCM	—	—	62.5	62.5	62.5	62.5	62.5
	CA	$\eta_{0=4}$	—	55.6	55.6	55.6	55.6	55.6
	AFCC	$\eta_{0=4}$	10	70.2	70.2	70.2	69.1	70.2
	SCAPC	$\eta_{0=4}$	10	73	73	73	73	73
	F(M)-SCAPC	$\eta_{0=4}$	10	75.3	75.3	75.3	75.3	75.3
	KCA	$\eta_{0=4,}$ σ=30	—	67.4	64.6	67.4	67.4	67.4
	KFCM-F	$\eta_{0=4,}$σ=95	—	70.2	71.9	48.3	70.2	72.5
	F(K)--SCAPC	$\eta_{0=4,}$ σ=30	10	77.5	75.8	74.1	75.8	74.1
Heart	FCM	—	—	59.3	59.3	59.3	59.3	59.3
	CA	$\eta_{0=4}$	—	59.3	59.3	59.3	59.3	59.3
	AFCC	$\eta_{0=4}$	15	60.1	60.1	60.1	60.1	60.1
	SCAPC	$\eta_{0=4}$	15	61.1	60.7	61.5	61.5	61.1
	F(M)-SCAPC	$\eta_{0=4}$	15	63.7	62.6	63.7	63.7	63.7
	KCA	$\eta_{0=4,}$ σ=25	—	59.6	59.6	59.6	59.6	59.6
	KFCM-F	$\eta_{0=4,}$σ=95	—	60.7	60.7	60.7	60.7	60.7
	F(K)-SCAPC	$\eta_{0=4,}$ σ=30	15	66.7	64.8	64.8	66.7	66.7

dimension data sets such as wine and heart. That means F (K)-SCAPC can map the nonlinear high-dimensional data to the feature space and obtain a linearly separable data in the new feature space. The algorithm F(M)-SCAPC which base on Mahalanobis metric can deal with the data sets that with large relatively correlation between samples such as wine data set. We take iris data set as an example. The pairwise constraints we used in the algorithms are 5 for must-link 3 and cannot-link 2. The average accuracy of the five tests for F(M)-SCAPC is 92.42% and the average accuracy of the five tests for F (K)-SCAPC is 92.8%. Both of the two algorithms average accuracy is higher than SCAPC which is 91.0%. In order to obtain the best clustering result to compare, we used different value of parameters for different algorithms. And only the best combinations of the parameters are list in the table. From the table we can see different data set need different parameter in one algorithm to have the best clustering result.

2) Influence of Gaussian kernel parameters σ on the cluster results

Gaussian kernel can be largely affected by the value of σ. Thus, we study value of σ impact on algorithm F(K)-SCAPC's performance. Here, besides above chosen three data sets, we also chose data set Heart from the UCI database. And the range of σ we choose is 0.5 to 40. The initial maximum number of clusters Cmax is set as 18. Threshold ε is set as 0.001. The threshold value e with cardinality is 7 and η0 is 4. The number of constraints we used in this experiment is set as 5. The results are listed in Tab 3, where symbol "—" means that algorithm do not gets the right number of clusters.

TABLE III. INFLUENCE OF σ ON THE CLUSTERING RESULTS (ACCURACY %)

σ	Iris	Wine	Heart	Diabetes
0.5	52.7	33.7	45.2	44.3
1	—	34.8	44.8	—
5	84.3	50.7	65.3	58.2
10	89.3	68.3	55.9	62.7
15	92.7	73.9	59.2	60.6
20	88.7	70.2	60.3	63.8
25	92	69.8	63.2	68.2
30	88.7	62.3	61.7	51.3
35	91.3	65.2	60.8	58.7
40	81.3	60.8	55.1	53.6

From Tab 3, we can see that low accuracies are obtained for most of the data sets when the value of σ is less than 1. What's worse is Iris and Diabetes cannot obtain correct classes in this condition. And when σ ranges from 15 to 35, better clustering results can obtained for most data sets. When the values of σ are 15, 20,25 and 25, the best results can be got for Iris, Wine, Heart and Diabetes, respectively.

IV. CONCLUSION

In this paper, we combine the semi-supervised clustering with Mahalanobis metric and Gaussian kernel to propose a new semi-supervised fuzzy clustering algorithm F-SCAPC. For different data sets including iris, diabetes, wine, Data and heart, respectively, we conduct a series of experiments from clustering speed and accuracy of clustering. Experiments show that the modified algorithm clearly improves the accuracy of clustering and speed of convergence. In the meantime, we also conduct the comparison on performance with FCM, CA, AFCC, KCA, KFCM-F and SCAPC. Experimental results show that the presented algorithm F-SCAPC is a more effective semi-supervised fuzzy clustering algorithm. In future, we further study semi-supervised fuzzy clustering algorithm based on other metric learning.

ACKNOWLEDGMENT

This work is support by Natural Science Foundation of China (No. 61073121) and Nature Science Foundation of Hebei Province (No. F2012201014).

REFERENCES

[1] S. Xiang, F. Nie, C. S. Zhang. "Learning a Mahalanobis distance metric for data clustering a classification," Pattern Recognition, 2008.

[2] Bar-Hillel,A. Hertz T., Shental N.,et al. "Learning a Mahalanobis metric from equivalence constraints," Journal of Machine Learning Research, 2005, 6:937–965.

[3] Yin X. S., Shu T.,Huang Q.., "Semi-supervised fuzzy clustering with metric learning and entropy regularization," Knowledge-Based Systems, 2012, 35:304–311.

[4] Yeung D.Y.,Chang H. "Extending the relevant component analysis algorithm for metric learning using both positive and negative equivalence constraints," Pattern Recognition, 2006, 39:1007-1010.

[5] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. S. Kandola, "On kernel-target alignment," in NIPS, 2001, pp. 367–373.

[6] Yang, L. and Jin, R. "Distance metric learning," A comprehensive survey. 2006

[7] Sindhwani, V., Niyogi, P., & Belkin, M. . "Beyondthe point cloud: From transductive to semisupervised learning," ICML, 2005.

[8] D.Q. Zhang, S.C. Chen, "Fuzzy Clustering Using Kernel Method," International Conference, 2002.

[9] Daniel Graves∗, WitoldPedrycz, "Kernel-based fuzzy clustering and fuzzy clustering," A comparative experimental study Fuzzy Sets and Systems 161 (2010) 522–543

[10] M. S. Baghshah, S. B. Shouraki, "Kernel-based metric learning for semi-supervised clustering," Neurocomputing , 73 (2010) 1352–1361

[11] H. Frigui, R. Krishnapuram, "Clustering by competitive agglomeration," Pattern Recognition 30 (7) (1997) 1109–1119.

[12] N. Grira, M. Crucianu, N. Boujemaa, "Semi-supervised fuzzy clustering with pairwise-constrained competitive agglomeration," in: IEEE International Conference on Fuzzy Systems (Fuzz'IEEE 2005), May 2005.

[13] N. Grira, M. Crucianu, N. Boujemaa, "Active semi-supervised fuzzy clustering," Pattern Recognition, 2008, 41 (5): 1834-1844.

[14] C. F. Gao, X. J. Wu, "A new semi-supervised clustering algorithm with pairwise constraints by competitive agglomeration," Applied Soft Computing, 2011.