Front. Electr. Electron. Eng. China 2011, 6(2): 363–373
DOI 10.1007/s11460-011-0154-y

RESEARCH ARTICLE

**Liang LIAO, Yanning ZHANG**

# MRI image segmentation based on fast kernel clustering analysis



**Abstract** Kernel-based clustering is supposed to provide a better analysis tool for pattern classification, which implicitly maps input samples to a high-dimensional space for improving pattern separability. For this implicit space map, the kernel trick is believed to elegantly tackle the problem of "curse of dimensionality", which has actually been more challenging for kernel-based clustering in terms of computational complexity and classification accuracy, which traditional kernelized algorithms cannot effectively deal with. In this paper, we propose a novel kernel clustering algorithm, called KFCM-III, for this problem by replacing the traditional isotropic Gaussian kernel with the anisotropic kernel formulated by Mahalanobis distance. Moreover, a reduced-set represented kernelized center has been employed for reducing the computational complexity of KFCM-I algorithm and circumventing the model deficiency of KFCM-II algorithm. The proposed KFCM-III has been evaluated for segmenting magnetic resonance imaging (MRI) images. For this task, an image intensity inhomogeneity correction is employed during image segmentation process. With a scheme called preclassification, the proposed intensity correction scheme could further speed up image segmentation. The experimental results on public image data show the superiorities of KFCM-III.

**Keywords** kernel-based clustering, dimensionality reduction, speeding-up scheme, magnetic resonance imaging (MRI) image segmentation, intensity inhomogeneity correction

Received March 24, 2011; accepted April 26, 2011

Liang LIAO (✉), Yanning ZHANG
Shaanxi Provincial Key Laboratory of Speech and Image Information Processing (SAIIP), School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China
E-mail: liaoliangis@126.com

Liang LIAO
School of Electric and Information Engineering, Zhongyuan University of Technology, Zhengzhou 450007, China

## 1 Introduction

Pattern analysis of magnetic resonance imaging (MRI) is important for diagnosing brain illnesses, such as Alzheimer's disease and schizophrenia. Among the MRI analyses, precise measurement of tissue distributions is critical for further treatment. But the precise measurement by means of MRI image segmentation is still challenging due to the task itself and a variety of noise corruptions. Among them, spurious intensity inhomogeneities, known as bias field in MRI data, could even reach 40% of local intensity altitude. Different to common additive noises, the dominated part of the spurious intensity inhomogeneities could be model as a low-frequently varying multiplicative field. Although it is not quite a problem to qualitative clinic diagnosis, the multiplicative field could severely tamper with accurate machine-based image processing and analysis. So the quantitative MRI image analysis needs to seriously reckon with bias field.

For correcting the aforementioned intensity inhomogeneities, historically, the correction approaches could roughly be categorized as the prospective and retrospective methods. The prospective methods are basically to alleviate the effect of bias fields before imaging process, e.g., to use imaging phantom, multiple coils, special imaging sequences for calibrating imaging equipments and conditions. The prospective methods use image processing techniques, such as filtering, surface interpolation, segmentation, histogram, etc., for intensity inhomogeneity correction after imaging process.

In this paper, we discuss the retrospective correction using segmentation results. Due to the helpfulness of intensity correction to image segmentation accuracy, the two parts could be regarded as the reciprocal ones in one process.

For segmentation purposes, a rich variety of algorithms could be used. Here we are interested in

segmentation based on fuzzy kernel-clustering due to the efficacy of fuzziness and kernel trick. Kernel trick is supposed to provide a better tool for pattern analysis, which implicitly maps input samples to a high-dimensional space for improving the pattern separability, making it possible to transform the linear recognition in original space to a nonlinear one in kernel space. Using the kernel trick, which computes the inner product of two mapped data points in original space, kernel-based pattern analysis is believed to elegantly tackle the problem of "curse of dimensionality" and could therefore be used for effectively segmenting MRI medical images.

In this paper, we introduce several kernel-clustering based image segmentation approaches and discuss their merits and drawbacks. Based on the discussion, we introduce a novel kernel clustering algorithm via reduced-set representation for image segmentation purpose. The algorithm uses the reduced sets for representing kernel cluster centers, making it circumvent the unnecessary computational complexity of traditional kernel clustering. Moreover, the algorithm uses a membership-based approach to estimate the well-tuned Gaussian kernel parameter. Based on clustering result, a residual-based bias field correction method is discussed. The correction and clustering processes benefit each other, making the segmentation accuracy further improved.

## 2 Backgrounds of clustering-based image segmentation

Traditionally, fuzzy $C$-means (FCM) clustering has been widely used for image segmentation. For improving its effectiveness, kernel trick and spatial constraints could be adopted. For example, the directly kernelized version of FCM, known as KFCM-I [1] could be used to improve the segmentation accuracy by implicitly mapping data to a higher dimensional kernel space and performing FCM clustering in kernel space, transforming the linear method to a nonlinear one [2,3]. Moreover, using a variety of schemes, KFCM-I could be improved. For example, a kernel-induced distance and spatial penalty term are introduced in the objective function of the algorithm called KFCM-II [1,4,5], which aims to retain the efficacy of kernel tricks as well as to incorporating the spatial continuity in clustering process. Theoretically, both KFCM-I and KFCM-II could be used for image segmentation. Before the discussion of their applications and their possible improvements, it is necessary to briefly introduce some of important backgrounds.

### 2.1 FCM

KFCM-I and KFCM-II are all based on FCM algorithm. FCM aims to extract $C$ clusters from $D$-dimensional data set containing $N$ data points $\boldsymbol{x} \triangleq (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N)^{\mathrm{T}} \in \mathbb{R}^{N\times D}$ by minimizing the following clustering objective function:

$$J_{\mathrm{FCM}} = \sum_{i=1}^{C}\sum_{k=1}^{N} \mu_{i,k}^{m} \left\|\boldsymbol{x}_k - \boldsymbol{v}_i\right\|^2, \tag{1}$$

where $\boldsymbol{v}_i$ is the $i$th cluster center, represented as a linear sum of input data points. $\mu_{i,k}$ (satisfying $\mu_{i,k} \in [0,1]$ and $\sum_{i=1}^{C} \mu_{i,k} = 1$) denotes the membership of $\boldsymbol{x}_k$ belonging to the $i$th cluster. And $m$ is the clustering constant and usually set as 2 for some convenience.

The constrained minimization of Eq. (1) using the condition satisfying $\sum_{i=1}^{C} \mu_{i,k} = 1$ could be solved by the well-known Lagrange multiplier technique. Namely, the constrained minimization of Eq. (1) is equivalent to the minimization of the following equation:

$$J_{\mathrm{Lag}} = \sum_{i=1}^{C}\sum_{k=1}^{N} \mu_{i,k}^{m} \left\|\boldsymbol{x}_k - \boldsymbol{v}_i\right\|^2 + \lambda\left(1 - \sum_{i=1}^{C}\mu_{i,k}\right). \tag{2}$$

The minimization of Eq. (2) could be obtained by taking the first order partial derivative of $J_{\mathrm{Lag}}$ with respect to $f$ membership $\mu_{i,k}$ and setting it zero. So the following equation is obtained ($m > 1$):

$$\frac{\partial J_{\mathrm{Lag}}}{\partial \mu_{i,k}} = m\mu_{i,k}^{m-1}\left\|\boldsymbol{x}_k - \boldsymbol{v}_i\right\|^2 - \lambda\Big|_{\mu_{i,k}=\mu_{i,k}^{*}} = 0. \tag{3}$$

The solution of $\mu_{i,k}^{*}$ is therefore obtained as follows:

$$\mu_{i,k}^{*} = \left(\lambda^{-1} m \left\|\boldsymbol{x}_k - \boldsymbol{v}_i\right\|^2\right)^{1/(1-m)}.$$

Since $\sum_{i=1}^{C} \mu_{i,k}^{*} = 1$, we have

$$\sum_{i=1}^{C}\left(\lambda^{-1} m \left\|\boldsymbol{x}_k - \boldsymbol{v}_i\right\|^2\right)^{1/(1-m)} = 1.$$

Solving for $\lambda$, we have

$$\lambda = m\left(\sum_{i=1}^{C}\left\|\boldsymbol{x}_k - \boldsymbol{v}_i\right\|^{2/(1-m)}\right)^{1-m}.$$

Using the obtained $\lambda$ and solving $\mu_{i,k}^{*}$ from Eq. (3), $\mu_{i,k}^{*}$ could be explicitly written as follows (where $\mu_{i,k}^{*}$ is replaced as $\mu_{i,k}$ for presentation clarity):

$$\mu_{i,k} = \frac{\left\|\boldsymbol{x}_k - \boldsymbol{v}_i\right\|^{-2/(m-1)}}{\sum_{j=1}^{C}\left\|\boldsymbol{x}_k - \boldsymbol{v}_j\right\|^{-2/(m-1)}}. \tag{4}$$

The explicit formulation of $\boldsymbol{v}_i$ could be obtained in similar way. Setting $\delta J_{\mathrm{Lag}}/\delta \boldsymbol{v}_i|_{\boldsymbol{v}_i=\boldsymbol{v}_i^{*}} = 0$, we get

$$\sum_{k=1}^{N} \mu_{i,k}^{m}\left(\boldsymbol{x}_k - \boldsymbol{v}_i\right)\Big|_{\boldsymbol{v}_i=\boldsymbol{v}_i^{*}} = 0. \tag{5}$$

Solving for $\boldsymbol{v}_i^*$, we have

$$\boldsymbol{v}_i = \frac{\sum_{k=1}^{N} \mu_{i,k}^m \boldsymbol{x}_k}{\sum_{k=1}^{N} \mu_{i,k}^m}. \tag{6}$$

After the initialization, FCM alternately iterates Eqs. (4) and (6) until the termination condition is reached and then uses the obtained $[\mu_{i,k}] \in \mathbb{R}^{C\times N}$ to transform the fuzzy clusters to hard ones.

### 2.2 KFCM-I

The direct kernelized version of FCM, known as KFCM-I, is actually quite straightforward. In KFCM-I, $\Phi(\cdot)$ denotes the implicit Mercer map from original data space to higher dimensional kernel space (where the mapped data points are believed more separable), and Eq. (1) could be rewritten as follows:

$$J_{\text{KFCM-I}} = \sum_{i=1}^{C}\sum_{k=1}^{N} \mu_{i,k}^m \left\|\Phi(\boldsymbol{x}_k) - \boldsymbol{v}_i^{\Phi}\right\|^2, \tag{7}$$

where $\boldsymbol{v}_i^{\Phi}$ is the $i$th cluster obtained in kernel space. Formally the iterative formulas could be consequently written as

$$\mu_{i,k} = \frac{\left\|\Phi(\boldsymbol{x}_k) - \boldsymbol{v}_i^{\Phi}\right\|^{-2/(m-1)}}{\sum_{j=1}^{C} \left\|\Phi(\boldsymbol{x}_k) - \boldsymbol{v}_j^{\Phi}\right\|^{-2/(m-1)}}, \tag{8}$$

$$\boldsymbol{v}_i^{\Phi} = \frac{\sum_{k=1}^{N} \mu_{i,k}^m \Phi(\boldsymbol{x}_k)}{\sum_{k=1}^{N} \mu_{i,k}^m}. \tag{9}$$

Due to the implicitness of $\Phi(\cdot)$, Eqs. (8) and (9) could not be directly computed. This dilemma could be avoided by substituting Eq. (9) to Eq. (8). More specifically, if Eq. (9) is rewritten as

$$\boldsymbol{v}_i^{\Phi} = \boldsymbol{\alpha}_i^{\mathrm{T}}\Phi(\boldsymbol{x}) = \sum_{k=1}^{N} \alpha_{i,k}\Phi(\boldsymbol{x}_k), \tag{10}$$

where $\boldsymbol{\alpha}_i \triangleq (\alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,N})^{\mathrm{T}} = \left(\sum_{k=1}^{N} \mu_{i,k}^m\right)^{-1} (\mu_{i,1}^m, \mu_{i,2}^m, ..., \mu_{i,N}^m)^{\mathrm{T}}$ and $\Phi(\boldsymbol{x}) \triangleq (\Phi(\boldsymbol{x}_1), \Phi(\boldsymbol{x}_2), ..., \Phi(\boldsymbol{x}_N))^{\mathrm{T}}$, the squared norm in Eq. (8) could be expanded as follows:

$$\begin{aligned}
&\left\|\Phi\left(\boldsymbol{x}_k\right) - \boldsymbol{v}_i^{\Phi}\right\|^2 \\
&\quad = \left(\Phi\left(\boldsymbol{x}_k\right) - \boldsymbol{v}_i^{\Phi}\right)^{\mathrm{T}} \left(\Phi\left(\boldsymbol{x}_k\right) - \boldsymbol{v}_i^{\Phi}\right) \\
&\quad = \Phi\left(\boldsymbol{x}_k\right)^{\mathrm{T}} \Phi\left(\boldsymbol{x}_k\right) - 2\Phi\left(\boldsymbol{x}_k\right)^{\mathrm{T}} \boldsymbol{v}_i^{\Phi} + \left(\boldsymbol{v}_i^{\Phi}\right)^{\mathrm{T}} \boldsymbol{v}_i^{\Phi} \\
&\quad = \Phi\left(\boldsymbol{x}_k\right)^{\mathrm{T}} \Phi\left(\boldsymbol{x}_k\right) - 2\sum_{k_1=1}^{N} \alpha_{i,k_1}\Phi\left(\boldsymbol{x}_k\right)^{\mathrm{T}} \Phi\left(\boldsymbol{x}_{k_1}\right) \\
&\qquad + \sum_{k_1=1}^{N}\sum_{k_2=1}^{N} \alpha_{i,k_1}\alpha_{i,k_2}\Phi\left(\boldsymbol{x}_{k_1}\right)^{\mathrm{T}} \Phi\left(\boldsymbol{x}_{k_2}\right).
\end{aligned} \tag{11}$$

Due to the kernel trick satisfying $K(\boldsymbol{x},\boldsymbol{y}) = \Phi(\boldsymbol{x})^{\mathrm{T}}\Phi(\boldsymbol{y})$ $(\forall \boldsymbol{x},\boldsymbol{y})$ for the Gaussian kernel $K(\boldsymbol{x},\boldsymbol{y}) \triangleq \exp(-\|\boldsymbol{x}-\boldsymbol{y}\|^2/(2\sigma^2))$, Eq. (11) becomes

$$\begin{aligned}
\left\|\Phi(\boldsymbol{x}_k) - \boldsymbol{v}_i^{\Phi}\right\|^2 &= 1 - 2\sum_{k_1=1}^{N} \alpha_{i,k_1} K\left(\boldsymbol{x}_k, \boldsymbol{x}_{k_1}\right) \\
&\quad + \sum_{k_1=1}^{N}\sum_{k_2=1}^{N} \alpha_{i,k_1}\alpha_{i,k_2} K(\boldsymbol{x}_{k_1}, \boldsymbol{x}_{k_2}).
\end{aligned} \tag{12}$$

Therefore, Eq. (12) could be iteratively computed, so does Eq. (8).

Although the clustering becomes feasible, the computational complexity has been significantly increased. Even without considering the calculation and storage of kernel matrix $[K\left(\boldsymbol{x}_{k_1}, \boldsymbol{x}_{k_2}\right)] \in \mathbb{R}^{N\times N}$, Eq. (12) expands the original one multiplication to at least $N^2$ multiplications, so the complexity increases from $O(1)$ to $O(N^2)$, which is not favourable.

### 2.3 KFCM-II

Due to complexity concern, a modified algorithm, called KFCM-II, is design to solve the problem. It is assumed that the kernel cluster center $\boldsymbol{v}_i^{\Phi}$ $(\forall i)$ could be replaced by $\Phi(\boldsymbol{v}_i)$. In other words, $\boldsymbol{v}_i^{\Phi}$ has a pre-image $\boldsymbol{v}_i$ existing in the original space. Using this assumption, the following equation holds:

$$\left\|\Phi(\boldsymbol{x}_k) - \boldsymbol{v}_i^{\Phi}\right\|^2 = 2\left(1 - K(\boldsymbol{x}_k, \boldsymbol{v}_i)\right). \tag{13}$$

Using Eq. (13), Eqs. (7)–(9) consequently become

$$J_{\text{KFCM-II}} = \sum_{i=1}^{C}\sum_{k=1}^{N} \mu_{i,k}^m \left(1 - K(\boldsymbol{x}_k, \boldsymbol{v}_i)\right), \tag{14}$$

$$\mu_{i,k} = \frac{\left(1 - K(\boldsymbol{x}_k, \boldsymbol{v}_i)\right)^{-1/(m-1)}}{\sum_{j=1}^{C} \left(1 - K(\boldsymbol{x}_k, \boldsymbol{v}_j)\right)^{-1/(m-1)}}, \tag{15}$$

$$\boldsymbol{v}_i = \frac{\sum_{k=1}^{N} \mu_{i,k}^m K(\boldsymbol{x}_k, \boldsymbol{v}_i)\boldsymbol{x}_k}{\sum_{k=1}^{N} \mu_{i,k}^m K(\boldsymbol{x}_k, \boldsymbol{v}_i)}. \tag{16}$$

So in KFCM-II, Eqs. (15) and (16) could be both explicitly computed. Compared with KFCM-I, the computational complexity has been significantly reduced. KFCM-II is effective under following considerations: 1) $1 - K(\boldsymbol{x}_k, \boldsymbol{v}_i)$ is proved a distance defined in the original

space. 2) Compared with the cluster center in Eq. (6), Eq. (16) is a kernel weighted one, but also obtained in the original space. So KFCM-II is actually a degenerated one from KFCM-I in kernel space to input space. Strictly speaking, this degeneration disqualifies KFCM-II as a kernelized clustering. In this sense, KFCM-II is called "fuzzy clustering with kernel-induced distance" [5].

Another important issue, which was not pointed out in Refs. [1,4,5], is that the assumption of the existence of pre-image does not actually hold. This conclusion can be easily derived from the work of Schölkopf et al. that was reported in 1999 [6]. More specifically, the linear sum containing at least two mapped data points does not have an exact pre-image in data space (more details could be found in Schölkopf's work). However, it should be pointed out that Eq. (16) is still effective because it is actually the approximate pre-image reported in Ref. [6]. The approximate pre-image is defined as follows:

$$\boldsymbol{v}_i = \arg\min_{\boldsymbol{x}} \left\| \Phi(\boldsymbol{x}) - \boldsymbol{v}_i^{\Phi} \right\|. \tag{17}$$

The solution of Eq. (17) could be obtained by minimizing $f(\boldsymbol{x}, \boldsymbol{v}_i) = \|\Phi(\boldsymbol{x}) - \boldsymbol{v}_i^{\Phi}\|^2$. Taking the derivative with respect to $\boldsymbol{x}$ and setting it to zeros, the following equation holds for the Gaussian kernel:

$$\begin{aligned}\frac{\delta f}{\delta \boldsymbol{x}} &= \frac{\delta}{\delta \boldsymbol{x}} \Bigg( 1 - 2 \sum_{k_1=1}^{N} \alpha_{i,k_1} K(\boldsymbol{x}, \boldsymbol{x}_{k_1}) \\ &\quad + \sum_{k_1=1}^{N} \sum_{k_2=1}^{N} \alpha_{i,k_1} \alpha_{i,k_2} K(\boldsymbol{x}_{k_1}, \boldsymbol{x}_{k_2}) \Bigg) \Bigg|_{\boldsymbol{x}=\boldsymbol{v}_i^*} \\ &= \frac{2}{\sigma^2} \sum_{k=1}^{N} \alpha_{i,k} K(\boldsymbol{x}, \boldsymbol{x}_k) (\boldsymbol{x} - \boldsymbol{x}_k) \Big|_{\boldsymbol{x}=\boldsymbol{v}_i^*} = 0. \end{aligned} \tag{18}$$

The solution of Eq. (18) is

$$\boldsymbol{v}_i^* = \frac{\sum_{k=1}^{N} \alpha_{i,k} K(\boldsymbol{v}_i^*, \boldsymbol{x}_k) \boldsymbol{x}_k}{\sum_{k=1}^{N} \alpha_{i,k} K(\boldsymbol{v}_i^*, \boldsymbol{x}_k)}.$$

Because $\alpha_{i,k} \triangleq \left( \sum_{k=1}^{N} \mu_{i,k}^m \right)^{-1} \mu_{i,k}^m$, we get Eq. (16). Although it guarantees the clustering effectiveness, the aforementioned approximate pre-image formulation inevitably brings the imperfection to KFCM-II, making it less robust and of accuracy limitation.

## 3 KFCM-III: Kernel fuzzy clustering via reduced-set representation

### 3.1 Reduced sets

In order to tackle the problems of KFCM-I and KFCM-II, a reduced-set representation for kernel clustering could be adopted. More specifically, it is believed unnecessary and less accurate to use all $N$ data points to represent the $C$ cluster centers. Alternatively, we could use the data sets which contain some typical data points with higher memberships belonging to specific cluster, to represent the corresponding cluster centers. We call the collections of these data points reduced sets. More specifically, if denoting the $i$th reduced set as $\boldsymbol{s}_i \triangleq (\boldsymbol{s}_{i,1}, \boldsymbol{s}_{i,2}, ..., \boldsymbol{s}_{i,N_i})^{\mathrm{T}}$, $\{\boldsymbol{s}_{i,1}, \boldsymbol{s}_{i,2}, ..., \boldsymbol{s}_{i,N_i}\} \subset \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$, and the kernelized version as $\Phi(\boldsymbol{s}_i) \triangleq (\Phi(\boldsymbol{s}_{i,1}), \Phi(\boldsymbol{s}_{i,2}), ..., \Phi(\boldsymbol{s}_{i,N_i}))^{\mathrm{T}}$, the $i$th kernel cluster center could be represented as follows:

$$\boldsymbol{v}_i^{\Phi} = \boldsymbol{a}_i^{\mathrm{T}} \Phi(\boldsymbol{s}_i) = \sum_{k=1}^{N_i} a_{i,k} \Phi(\boldsymbol{s}_{i,k}), \tag{19}$$

where $\boldsymbol{a}_i \triangleq (a_{i,1}, a_{i,2}, \ldots, a_{i,N_i})^{\mathrm{T}}$ is $\ell^1$ normalized (i.e., the sum of all elements is equal to 1) coefficient vector with positive elements. The formulation of $\boldsymbol{a}_i$ could be determined by different strategies. One solution is to use the following representation similar to the corresponding one in FCM:

$$\begin{aligned}\boldsymbol{a}_i = &\left( \sum_{k=1}^{N_i} \mu_{i,\mathrm{IND}(\boldsymbol{s}_{i,k})}^m \right)^{-1} \left( \mu_{i,\mathrm{IND}(\boldsymbol{s}_{i,1})}^m, \mu_{i,\mathrm{IND}(\boldsymbol{s}_{i,2})}^m, ..., \right. \\ &\left. \mu_{i,\mathrm{IND}(\boldsymbol{s}_{i,N_i})}^m \right)^{\mathrm{T}}, \end{aligned} \tag{20}$$

where $\mathrm{IND}(\boldsymbol{s}_{i,k})$ denotes the index of $\boldsymbol{s}_{i,k}$ in $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$.

Using Eq. (20), Eq. (12) has been simplified as follows:

$$\begin{aligned}&\left\| \Phi(\boldsymbol{x}_k) - \boldsymbol{v}_i^{\Phi} \right\|^2 \\ &\quad = 1 - 2 \sum_{k_1=1}^{N_i} a_{i,k_1} K(\boldsymbol{x}_k, \boldsymbol{s}_{i,k_1}) \\ &\qquad + \sum_{k_1=1}^{N_i} \sum_{k_2=1}^{N_i} a_{i,k_1} a_{i,k_2} K(\boldsymbol{s}_{i,k_1}, \boldsymbol{s}_{i,k_2}), \ \ \forall k, i. \end{aligned} \tag{21}$$

Thus, the complexity has been reduced from $O(N^2)$ to $O(N_i^2)$ $(N_i < N)$.

Note for each clustering iteration, $\sum_{k_1=1}^{N_i} \sum_{k_2=1}^{N_i} a_{i,k_1} \cdot a_{i,k_2} K(\boldsymbol{s}_{i,k_1}, \boldsymbol{s}_{i,k_2})$ is only computed once because this term is actually a constant when $\boldsymbol{s}_i$ has been determined. Thus, the most complex term is computationally distributed over each $k$. Using this trick, the calculation of Eq. (21) could be further accelerated after a relatively slow iteration initialization. Equation (19) is called as the reduced-set representation of kernel cluster center. So the clustering iterates Eqs. (20), (21) and (8) until the termination condition is reached.

In Fig. 1, we give an illustration of three clusters and their reduced sets. Due to their high memberships in reduced sets, the data points are generally some with the nearest distances to their cluster centers while the

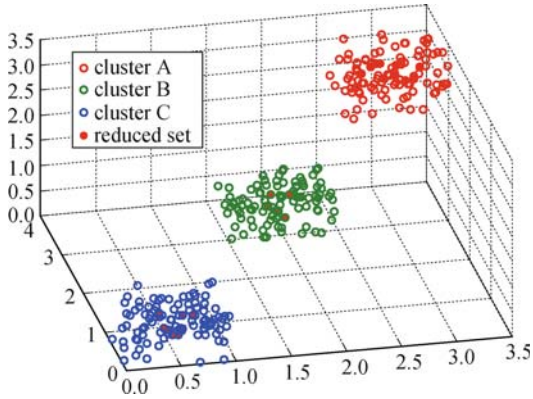


**Fig. 1** An illustration of reduced sets

cluster centers are believed appropriately represented by those data points.

### 3.2 Covariance-based anisotropic kernel parameterization

As aforementioned, the parameterization of the Gaussian kernel, defined as

$$K(\boldsymbol{x},\boldsymbol{y}) \triangleq \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}\right) = \Phi(\boldsymbol{x})^{\mathrm{T}}\Phi(\boldsymbol{y}),$$

is isotropic, because there is only one parameter $\sigma$ globally controlling the separability of the mapped data points. Either large or small $\sigma$ could lead to classification disaster. In other words, if $\sigma$ is small enough, $K(\cdot,\cdot)$ is approximately equal to 0, meaning any two mapped data points are approximately orthogonal and sharing no similarity. On the other hand, if $\sigma$ is large enough, the inter product of any two mapped data points is approximately 1, meaning they are approximately overlapped. Both situations make pattern classification unfeasible. So for good classification (clustering) performance, the kernel parameter(s) should be well-tuned.

The estimation of $\sigma$ is still an open problems despite of some works attempting to give some solutions [7–9]. Despite of the work trying to find optimal $\sigma$ estimation, it is reasonable to assume that even with well-estimated $\sigma$, the clustering performance could still be limited. The reason could be somehow interpreted by the concept of data manifold. Basically, manifold is a geometry concept, which could be roughly described as the hyper-surface spanned by data points in high-dimensional space, which visually reflects the geometric properties of data distributions. Suitable (explicit or implicit) transform should be able to embed data manifold or optimize manifold in the mapped space. From the point of view of manifold embedding, we suggest to replace the traditional isotropic kernel with an anisotropic one, whose parameterization could vary at different geometric locations. Due to the close connection of manifold and generative model, a Gaussian mixture model (GMM) based parameterization could be adopted. More specifically, in GMM, the probability density distribution of observed data $\boldsymbol{x}$ could be given as follows:

$$p(\boldsymbol{x}) = \sum_{i=1}^{C} \pi_k \mathcal{N}\left(\boldsymbol{x}\,|\boldsymbol{v}_i, \boldsymbol{\Sigma}_i\right), \tag{22}$$

where $\pi_i > 0\ (\forall i)$, $\sum_{i=1}^{C}\pi_i = 1$ and $\mathcal{N}\left(\boldsymbol{x}\,|\boldsymbol{v}_i, \boldsymbol{\Sigma}_i\right)$ is the $i$th Gaussian component with the mean and covariance matrix respectively denoted as $\boldsymbol{v}_i$ and $\boldsymbol{\Sigma}_i$. Basically, the $i$th Gaussian component could be used for modeling the corresponding cluster. Using the expectation maximization (EM) and maximum likelihood (ML) methods, $\boldsymbol{v}_i$ and $\boldsymbol{\Sigma}_i$ could be formulated as follows:

$$\boldsymbol{v}_i = \sum_{k=1}^{N} \beta_{i,k}\boldsymbol{x}_k, \tag{23}$$

$$\boldsymbol{\Sigma}_i = \sum_{k=1}^{N} \beta_{i,k}\left(\boldsymbol{x}_k - \boldsymbol{v}_i\right)\left(\boldsymbol{x}_k - \boldsymbol{v}_i\right)^{\mathrm{T}}, \tag{24}$$

where $\left(\beta_{i,1}, \beta_{i,2}, ..., \beta_{i,N}\right)^{\mathrm{T}}$ is an $\ell^1$ normalized coefficient vector with positive elements. In our kernel parameterization, we use Eqs. (23), (24) and $\beta_{i,k} \triangleq \left(\sum_{k=1}^{N}\mu_{i,k}\right)^{-1}\mu_{i,k}$. So the anisotropic kernel is defined as follows:

$$K\left(\boldsymbol{x},\boldsymbol{y}\,|\boldsymbol{\Sigma}_i\right) = \exp\left[-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{y}\right)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}\left(\boldsymbol{x}-\boldsymbol{y}\right)\right]. \tag{25}$$

Note in Eq. (25), $\left(\boldsymbol{x}-\boldsymbol{y}\right)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}\left(\boldsymbol{x}-\boldsymbol{y}\right)$ is actually Mahalanobis distance. If $\boldsymbol{\Sigma}_i = \sigma^2\boldsymbol{I}$ ($\boldsymbol{I}$ is a $D\times D$ identity matrix), $\left(\boldsymbol{x}-\boldsymbol{y}\right)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}\left(\boldsymbol{x}-\boldsymbol{y}\right)$ could be reduced to the Euclidean distance $\sigma^{-2}\left\|\boldsymbol{x}-\boldsymbol{y}\right\|^2$, which is used in the traditional Gaussian kernel.

Before using Eq. (25), some concerns should be clarified. The with-cluster covariance matrix in Eq. (24) could still be problematic: 1) Because the classification information is not available in unsupervised learning, Eq. (24) could not be computed a priori. One strategy is to use a well-tuned estimation (e.g., obtained by FCM) as the classification information for the initialization of sample covariance matrix and iterates it during the clustering procedure. 2) If the number of data points in one cluster is small enough compared to the input dimensionality $D$, the sample covariance matrix could be singular and lead to some clustering instabilities. So the regularization of Eq. (24) is needed. One possible solution is to have matrix diagonalized, i.e., only the diagonal matrix elements are retained and non-diagonal elements are just nullified. The diagonalization of Eq. (24) could be written as follows:

$$\boldsymbol{\Sigma}_i = \mathrm{diag}\left[\left(\sum_{k=1}^{N}\mu_{i,k}\right)^{-1}\sum_{k=1}^{N}\mu_{i,k}\left(\boldsymbol{x}_k-\boldsymbol{v}_i\right)\left(\boldsymbol{x}_k-\boldsymbol{v}_i\right)^{\mathrm{T}}\right]. \tag{26}$$

The diagonalization of Eq. (26) means the features of input data are zero correlated. Basically, data decorrelation is the concern of feature extraction. If the features are not zero correlated, the observed data should be regularized.

If we rewrite Eq. (26) as $\boldsymbol{\Sigma}_i = \mathrm{diag}\left(\sigma_{i,1}^2, \sigma_{i,2}^2, ..., \sigma_{i,D}^2\right)$, the Mahalanobis distance used in Eq. (25) could be rewritten as follows:

$$(\boldsymbol{x}-\boldsymbol{y})^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{y}) = \sum_{k=1}^{D}(x_k-y_k)^2/\sigma_{i,k}^2, \quad (27)$$

where $x_k$ and $y_k$ are the $k$th elements of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively.

Despite of the aforementioned formulations, the discussed kernel anisotropy also exhibits that kernel calculation is cluster dependent. Thus, the anisotropic version of Eq. (21) could be written as follows:

$$\begin{aligned}&\left\|\Phi(\boldsymbol{x}_k)-\boldsymbol{v}_i^{\Phi}\right\|^2 \qquad \forall k,i\\ &\quad = 1-2\sum_{k_1=1}^{N_i}a_{i,k_1}K(\boldsymbol{x}_k,\boldsymbol{s}_{i,k_1}|\boldsymbol{\Sigma}_i)\\ &\qquad +\sum_{k_1=1}^{N_i}\sum_{k_2=1}^{N_i}a_{i,k_1}a_{i,k_2}K\left(\boldsymbol{s}_{i,k_1},\boldsymbol{s}_{i,k_2}|\boldsymbol{\Sigma}_i\right). \quad (28)\end{aligned}$$

### 3.3 Determination of reduced sets

As aforementioned, the reduced sets contain typical data points with highest memberships belonging to corresponding clusters. Using this principle, the reduced sets could be determined. Here we give a reduced-set formulation using on the exponential function $f(\boldsymbol{x}) = \exp\left(-\frac{1}{2}\sum_{k=1}^{D}\frac{x_k^2}{\sigma_k^2}\right)$, whose inflexion points is valued as $(\sigma_1, \sigma_2, ..., \sigma_D)$. As for Eq. (27), the data points of the $i$th reduced set are selected from the data points with the highest membership to the $i$th cluster and subject to the following condition: $\forall \boldsymbol{x}, \boldsymbol{y} \in \{\boldsymbol{s}_{i,1}, \boldsymbol{s}_{i,2}, ..., \boldsymbol{s}_{i,N_1}\}$ and $\boldsymbol{x} \triangleq (x_1, x_2, ..., x_D)^{\mathrm{T}}$, $\boldsymbol{y} \triangleq (y_1, y_2, ..., y_D)^{\mathrm{T}}$, then

$$|x_k - y_k| \leqslant 2\sigma_{i,k}, \quad \forall k = 1, 2, ..., D. \quad (29)$$

### 3.4 Connection between KFCM-I, KFCM-II and KFCM-III (via reduced-set representation)

Generally speaking, KFCM-III (via reduced-set representation) could be regarded as a generalized version of KFCM-I and KFCM-II. KFCM-I and KFCM-II respectively corresponds to KFCM-III with the largest and smallest reduced sets.

Apparently, if all reduced sets are chosen as the whole input set $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$, and the diagonal elements in $\boldsymbol{\Sigma}_i = \mathrm{diag}\left(\sigma_{i,1}^2, \sigma_{i,2}^2, ..., \sigma_{i,D}^2\right)$ are identical, i.e., $\boldsymbol{\Sigma}_i \triangleq \mathrm{diag}\left(\sigma^2, \sigma^2, ..., \sigma^2\right)$ $(\forall i)$, KFCM-III therefore becomes KFCM-I. On the other hand, $\boldsymbol{\Sigma}_i \triangleq \mathrm{diag}\left(\sigma^2, \sigma^2, ..., \sigma^2\right)$, but setting all reduced sets respectively have only one element, KFCM-III is therefore reduced to KFCM-II. So the idea of KFCM-III is to tune the sizes of each reduced set to well balance the accuracy and efficiency of clustering.

## 4 Spurious intensity inhomogeneity correction

Bias field is usually modeled as a low-frequently varying multiplicative field. To be more specific, let $Y_k$ denote the $k$th observed pixel intensity, $X_k$ the bias-field-free intensity, $B_k$ the bias field, then pixel-wise observed data could be represented as follows:

$$Y_k = B_k X_k, \quad \forall k = 1, 2, ..., N. \quad (30)$$

With the pixel intensity extracted as input data ($N$ denoting the pixel number), according to Ref. [10], a mathematically graceful method to correct pixel intensities could be used. The correction method was first reported for spatially constrained FCM clustering. In our study, spatially constrained methods are not our concerns. So with a small amount of mathematical manipulations, the correction method could be recapitulated as follows.

The logarithm on Eq. (30) transforms the multiplicative field to the addictive one. To be more specific, the transformed additive one could be rewritten as follows:

$$y_k = b_k + x_k, \quad \forall k = 1, 2, ..., N. \quad (31)$$

For FCM algorithm considering bias field estimation, the modified objective function could be given as follows:

$$J_{\mathrm{m}} = \sum_{i=1}^{C}\sum_{k=1}^{N}\mu_{i,k}^m (y_k - b_k - v_i)^2. \quad (32)$$

Bias field could be estimated by taking the derivative of $J_{\mathrm{m}}$ with respect to $b_k$ and setting the derivative to zero. Thus, we have the following equation:

$$\sum_{i=1}^{C}\frac{\partial}{\partial b_k}\sum_{k=1}^{N}\mu_{i,k}^m (y_k - b_k - v_i)^2\Big|_{b_k=b_k^*} = 0. \quad (33)$$

With a small amount of mathematical manipulations, the left side of Eq. (33) could be expanded and we have

$$y_k\sum_{i=1}^{C}\mu_{i,k}^m - b_k\sum_{i=1}^{C}\mu_{i,k}^m - \sum_{i=1}^{C}\mu_{i,k}^m v_i\Big|_{b_k=b_k^*} = 0. \quad (34)$$

The solution of $b_k^*$ gives the bias field estimator as

follows:

$$b_k^* = y_k - \frac{\sum_{i=1}^{C} \mu_{i,k}^m v_i}{\sum_{i=1}^{C} \mu_{i,k}^m}. \tag{35}$$

The above equation also implies that the corrected intensity could be represented as a linear sum of cluster centers. The residual, represented as the subtraction of observed intensity and the linear sum of cluster centers could be used for modelling bias field.

The idea could be extended in kernel fuzzy clustering for intensity correction. Namely, the logarithm-transformed intensity $x_k$ has been corrected as follows:

$$x_k = \left( \sum_{i=1}^{C} \mu_{i,k}^m \right)^{-1} \sum_{i=1}^{C} \mu_{i,k}^m v_i, \tag{36}$$

where $v_i$ is the $i$th logarithm-transformed cluster center. Due to the implicitness of kernerlized cluster center of KFCM-III, $v_i$ could be represented as the log-transformed approximate pre-image of kernerlized cluster center, which is determined by Eq. (16).

Using this idea, the aforementioned formulation could also be extended for the multivariate input data by respectively applying the estimator to each dimension of input vectors and therefore be applicable to kernel-clustering based image segmentation.

## 5 Feature extraction and pre-classification scheme

### 5.1 Feature extraction, spatial constraints and random feature

Feature extraction that used for data representation plays an important role for pattern analysis. Of a variety of possible features, pixel intensity is apparently the most distinctive feature for clustering-based image segmentation.

Besides the intensity of target pixel, spatial constraints of target pixel are usually expected to be incorporated as part of input features. Traditionally for clustering-based image segmentation, spatial constraints are usually incorporated in the clustering objective function as a penalty term weighted by a coefficient, which is usually determined empirically.

Apparently, how to tune the coefficient balancing between the penalty term and the other part of objective function is another difficult problem to reckon with.

Therefore, in this study, we use a simple neighbourhood-based data feature, stacking the pixel intensities in a neighbourhood (say 5×5 neighbourhood) as the input vector of target pixel.

Basically, the neighbourhood size could influence on clustering performance. Large neighbourhood means higher dimensionality of input vector, which will inevitably bring higher computational complexity. On the other hand, small neighbourhood might reduce the pattern separability. Due to the piece-wise smoothness of image, the features from smaller neighbourhood have a higher probability to overlap and are therefore less separable. In other words, smaller neighbourhood corresponds to fewer unique (not identical) input vectors. For the extreme case, the smallest neighbourhood only contains the target pixel, the input vector degenerates to a scalar (i.e., the target pixel intensity). For grayscale image with the intensity represented in 1 byte, the unique number of input data is not larger than 255. So the parsimony of effective data (unique data) could lead to recognition limitation.

To reckon with this dilemma between computational complexity and effectiveness of data representation, the recently-emerged compressive sampling (CS) theory could be used. Briefly speaking, in compressive sampling, a higher dimensional vector could be projected on underdetermined random matrix which satisfies some condition of CS theory and usually could be obtained by Gaussian distribution. Thus, a lower dimensional vector could be obtained. The lower dimensional vector is therefore called compressed measurements.

Let $\boldsymbol{y} \in \mathbb{R}^{D\times 1}$ denote the original vector, $\boldsymbol{x} \in \mathbb{R}^{d\times 1}$ the compressed data and $\boldsymbol{\Psi} \in \mathbb{R}^{d\times D}$ $(d < D)$ the underdetermined i.i.d. Gaussian matrix, the undersampling scheme could be written as $\boldsymbol{x} = \boldsymbol{\Psi y}$. Intriguingly, it has been mathematically proved that the sampling rate of CS could be much lower than the required one of classical Nyquist-Shannon sampling theory without losing any information of original data.

Therefore, the compressed measurements could be used for exact data reconstruction [11–13]. Besides exact data reconstruction, it has also been proved that the data manifold could be stably embedded in compressed domain, in which the pattern classification (known as compressive recognition) is equivalently practicable.

Thus, using compressive sampling technique and without losing the recognition effectiveness, features extracted from large neighbourhood could be dimension reduced. For example, we could first use a 5×5 neighbourhood for feature extraction, getting a 25-dimensional vector, and then use an random sampling matrix with i.i.d. Gaussian distribution to compress feature vectors to 9-dimensional for the following clustering-based image segmentation. We call the compressed measurements the random features.

### 5.2 Pre-classification scheme

Besides the dimension reduction brought by compressive sampling, the image piecewise-smoothness could be exploited for clustering acceleration because some of the input features, from the whole image domain perspective, are actually identical.

Therefore, a pre-classification scheme that is similar to the trick reported in Refs. [14,15] could be used for this purpose. The scheme is actually quite straightforward: before clustering, all identical (overlapped) vectors are classified as one unique pattern bestowed with a weight equal to the number of identical data. Due to the piecewise smoothness of image, the pre-classification scheme could significantly reduce the effective data points to be clustered. For example, the input vectors corresponding to $N$ pixels could be pre-classified as $M$ unique vectors. For example, for input vector $x_k$, correspondingly there are other $w_k - 1$ identical vectors $(k = 1, 2, ..., M, \sum_k w_k = N)$.

Let $\left\{\boldsymbol{x}_1^{(u)}, \boldsymbol{x}_2^{(u)}, ..., \boldsymbol{x}_M^{(u)}\right\}$ denote the $M$ unique data and $\{w_1, w_2, \ldots, w_M\}$ the corresponding weights. So the clustering objective function could be rewritten as

$$J_{\text{pre}} = \sum_{i=1}^{C}\sum_{k=1}^{M} \mu_{i,k}^{m} w_k D\left(\boldsymbol{x}_k^{(u)}, \boldsymbol{\theta}_i\right), \tag{37}$$

where $\boldsymbol{\theta}_i$ is the parameter(s) of the $i$th cluster and $D(\boldsymbol{x}, \boldsymbol{\theta}_i) \geqslant 0$ denotes the dissimilarity measurement of $x$ to the $i$th cluster. In our study, $\boldsymbol{\theta}_i$ represents the $i$th cluster center $(\boldsymbol{v}_i)$ and $D(\cdot, \cdot)$ the squared $\ell^2$ (or kernelized) norm. Because $M \ll N$, the computational complexity has been significantly reduced. Thus, using the pre-classification scheme for image segmentation, the discussed KFCM-I, KFCM-II, KFCM-III and even FCM could be further accelerated.

## 6 Experimental verification

In this section, we present the experiments on Brain-Web MRI phantom data, a publicly available database for studies of brain MRI image analysis. The dataset is generated by a computerized simulator, with a pixel-wise segmentation ground-truth, making the quantitative comparison convenient [16–18]. We use the normal brain MRI phantom for our experiments. The dataset is given as a 217×181×181 3D volume, with 8-bit voxel intensity and imaging resolution of 1 $\text{mm}^3$. The experiment is to segment regions of interest (ROIs) of the 2D slices to different tissues. More specifically, in the experiment, we segment the interested regions to gray matter (GM), white matter (WM) and cerebrospinal fluid (CSF). The uninterested regions have been removed by other method in advance. The clustering constant $m$ in the objective function is set as 2.

### 6.1 Segmentation performance of kernel clustering algorithms over $\sigma$

In the experiments, we compare the performances of different kernel clustering algorithms over isotropic Gaussian kernel parameter $\sigma$. This setting means for the aforementioned KFCM-III with $\boldsymbol{\Sigma}_i$ parameterization, it satisfies $\boldsymbol{\Sigma}_i = \sigma \boldsymbol{I}$ (where $\boldsymbol{I}$ is an identity matrix).

Using different $\sigma$, this experiment gets a series of accuracy curves of FCM, KFCM-I, KFCM-II and KFCM-III over $\sigma$ for segmenting a coronal slice respectively corrupted by 0%, 1% and 3% noise. One note should be given that the clustering accuracy of FCM does not depend on $\sigma$. Thus, its segmentation curve is represented as a horizontal line. Figure 2 shows those segmentation accuracy curves. The statistics are obtained in the experiments using different algorithms over different values of $\sigma$, ranging from 10 to 150 with the step of 5. It is easy to find that the segmentation accuracy of all kernel-based clustering (KFCM-I, KFCM-II and KFCM-III) are all dependent on $\sigma$ and there exists relative stable range that the tuned $\sigma$ is located in. Outside the range, the accuracy dramatically deteriorates. The experimental results show that generally, the discussed KFCM-III outperforms its counterparts when using tuned $\sigma$.

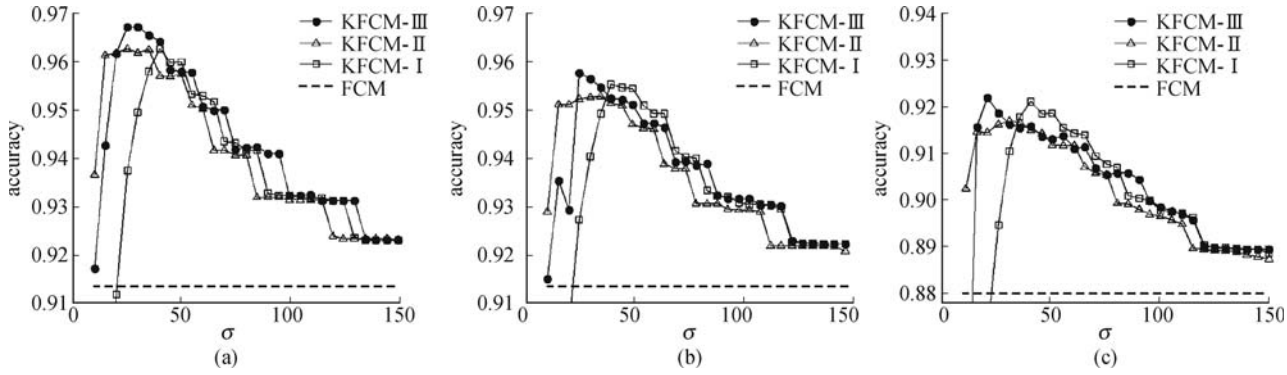


**Fig. 2** Typical segmentation accuracy curves of FCM, KFCM-I, KFCM-II and KFCM-III (via reduced-set representation) over different values of $\sigma$ on a phantom slice respectively corrupted by different noise levels. (a) Results on image slice corrupted by 0% noise; (b) results on image slice corrupted by 1% noise; (c) results on image slice corrupted by 3% noise

### 6.2 Segmentation comparison

Next, in an experiment on a coronal slice, the visual segmentation comparison of FCM, KFCM-I, KFCM-II and KFCM-III is given in Fig. 3. The segmentation results have been pseudo-coloured, with the red region corresponding to GM, the green region corresponding to CSF and blue region corresponding to WM.

In Fig. 3, the optimal segmentations obtained by different algorithms using tuned Gaussian kernel parameter $\sigma$ are given. It is not difficult to find that the result obtained by KFCM-III is the closest to the ground-truth.

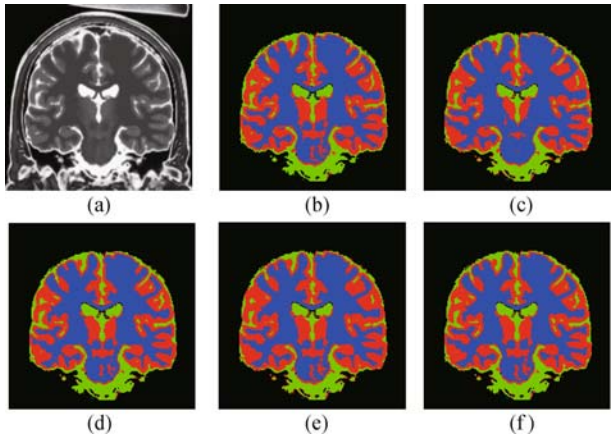


**Fig. 3** Pseudo-coloured segmentation results of FCM, KFCM-I, KFCM-II and KFCM-III on a phantom slice with red region corresponding to GM, green region corresponding to CSF and blue region corresponding to WM. (a) Original image; (b) ground-truth segmentation; (c) result of FCM; (d) optimal result of KFCM-I with tuned parameter $\sigma$=40, segmentation accuracy is 96.28%; (e) optimal result of KFCM-II with tuned parameter $\sigma$=25, segmentation accuracy is 96.28%; (f) optimal result of KFCM-III via reduced-set representation, segmentation accuracy is 96.73%

### 6.3 Reduced sets and bias field estimation

For the segmentation experiment corresponding to the results presented in Fig. 3, we give the pseudo-coloured reduced-set representation of KFCM-III in Fig. 4.

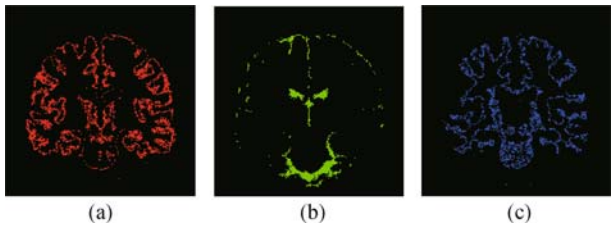


**Fig. 4** Pseudo-coloured reduced sets of KFCM-III on a phantom slice with $\sigma$=25. (a) Reduced set for GM; (b) reduced set for CSF; (c) reduced set for WM

Figure 4 shows the pseudo-coloured pixels of reduced sets for the three clusters, with red region, green region and blue region corresponding to GM, CSF and WM, respectively. As aforementioned, the elements of reduced set are basically some pixels with the highest memberships belonging to specific cluster. It is also easy to find from Fig. 4 that the reduced sets are apparently the subsets of corresponding hard clusters.

We also present the bias field estimation and corrected image in this experiment corresponding to the results shown in Fig. 2 and Fig. 3.

Figure 5 shows the original target image, corrected image and the bias field estimation. It is apparent that the corrected image is more piece-wise smooth than uncorrected one. The estimated bias field is shown in Fig. 5(c). Besides the low frequencies reflecting the true bias field, the obtained bias field also contains some high frequencies, spatially occurring on the borders of different tissues. High frequencies could further be removed for more accurate bias field estimation. There are a lot of high-frequency filtering methods available. Due to the clustering methods this study focuses, we leave the filtering techniques without further discussion.

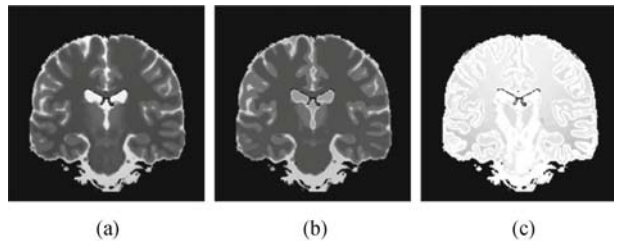


**Fig. 5** Bias field estimation and image correction. (a) Uncorrected image; (b) corrected image; (c) bias field estimation

Note that the discussed image correction (or equivalently bias field estimation) and the aforementioned preclassification scheme could benefit each other, making a fast segmenting convergence. Using the discussed intensity correction scheme, the number of effective (unique) input data points $M$ in Eq. (37) has been an approximate exponentially decreasing function of clustering iterations. Figure 6 gives the curve of $M$ over the clustering iterations, where the legend symbol pn$X$rf$Y$ (e.g.,

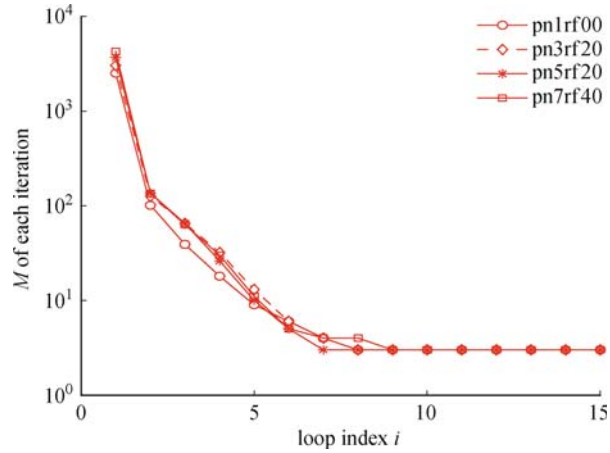


**Fig. 6** Curves of number of effective input data points over clustering iterations with image intensity correction

pn1rf40) means the experimental slice is corrupted by $X\%$ noise and $Y\%$ bias field. For clarity purpose, only four curves have been given in Fig. 6 although the experiments with other image corruption combinations had given the similar curves like the four ones shown in Fig. 6.

Due to the decrease of $M$ in clustering iteration, the computational complexity has also iteratively reduced. In this sense, the introduction of intensity correction does not bring additional computational complexity while improving the segmentation accuracy if correction process is well harnessed.

## 7 Conclusions

In this paper, we have analyzed the merits and deficiencies of KFCM-I/KFCM-II, and pointed out the connections of these two algorithms. In our study, we have proven that KFCM-II could be regarded as the approximate pre-image version of KFCM-I, while the latter, as a direct kernelized version of classic FCM algorithm, suffers the unnecessary computational complexity. Due to their deficiencies, we have proposed a novel fast kernel clustering algorithm, called KFCM-III, via reduced-set representation. In this work, we have contended that the parsimony for representing cluster centers could be exploited. Using this idea, we have proposed the use of some data points with highest memberships, included in a subset of input data called reduced set for specific cluster, to represent the corresponding cluster center.

The proposed KFCM-III via reduced-set representation could be regarded as a generalized version of KFCM-I and KFCM-II, and combines the strength of both when using appropriate parameterization. The proposed algorithm could be favourably used for dealing with large scale clustering problems. The application on MRI image segmentation with a bias field estimation formulated by clustering residuals shows its effectiveness and superiority to its kernelized counterparts.

Future work includes an improved kernel parameterization, either anisotropic or isotropic, which is easily tuned to optimal clustering accuracy and robust to clustering initialization.

**Acknowledgements** This work was partially supported by the National Natural Science Foundation of China (Grant Nos. 60872145, 60902063), the National High Technology Research and Development Program of China (Grant No. 2009AA01Z315), the Cultivation Fund of the Key Scientific and Technical Innovation Project, Ministry of Education of China (No. 708085), and the Henan Research Program of Foundation and Advanced Technology (No. 082300410090).

## References

1. Zhang D Q. Kernel-based associative memories, clustering algorithms and their applications. Dissertation for the Doctoral Degree. Nanjing: Nanjing University of Aeronautics and Astronautics, 2004 (in Chinese)
2. Girolami M. Mercer kernel-based clustering in feature space. IEEE Transactions on Neural Networks, 2002, 13(3): 780–784
3. Müller K R, Mika S, Rätsch G, Tsuda K. An introduction to kernel-based learning algorithms. IEEE Transactions on Neural Networks, 2001, 12(2): 181–202
4. Zhang D Q, Chen S C. A novel kernelized fuzzy C-means algorithm with application in medical image segmentation. Artificial Intelligence in Medicine, 2004, 32(1): 37–40
5. Chen S C, Zhang D Q. Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure. IEEE Transactions on System, Man and Cybernetics, Part B, 2004, 34(4): 1907–1916
6. Schölkopf B, Mika S, Burges C J C, Knirsch P, Müller K, Rätsch G, Smola A J. Input space versus feature space in kernel-based methods. IEEE Transactions on Neural Networks, 1999, 10(5): 1000–1017
7. Bi L P, Huang H, Zheng Z Y, Song H T. New heuristic for determination Gaussian kernel's parameter. In: Proceedings of 2005 International Conference on Machine Learning and Cybernetics. 2005, 7: 4299–4304
8. Wang W J, Xu Z B, Lu W Z. Determination of the spread parameter in the Gaussian kernel for classification and regression. Neurocomputing, 2003, 3(4): 643–663
9. Zhang D Q, Chen S C, Zhou Z H. Learning the kernel parameters in kernel minimum distance classifier. Pattern Recognition, 2006, 39(1): 133–135
10. Ahmed M, Yamany S, Mohamed N, Farag A A, Moriarty T. A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data. IEEE Transactions on Medical Imaging, 2002, 21(3): 193–199
11. Candes E J, Romberg J, Tao T. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on Information Theory, 2006, 52(2): 489–509
12. Candes E J, Tao T. Near-optimal signal recovery from random projections: universal encoding strategies? IEEE Transactions on Information Theory, 2006, 52(12): 5406–5425
13. Donoho D L. Compressed sensing. IEEE Transactions on Information Theory, 2006, 52(4): 1289–1306
14. Szilagyi L, Benyo Z, Szilagyi S M, Adam H S. MR brain image segmentation using an enhanced fuzzy C-means algorithm. In: Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 2003, 1: 724–726
15. Cai W L, Chen S C, Zhang D Q. Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation. Pattern Recognition, 2007, 40(3): 825–838
16. Aubert-Broche B, Evans A C, Collins L. A new improved version of the realistic digital brain phantom. Neuroimage, 2006, 32(1): 138–145
17. Collins D L, Zijdenbos A P, Kollokian V, Sled J G, Kabani N J, Holmes C J, Evans A C. Design and construction of a realistic digital brain phantom. IEEE Transactions on Medical Imaging, 1998, 17(3): 463–468
18. Cocosco C A, Kollokian V, Kwan R K S, Evans A C. BrainWeb: online interface to a 3-D MRI simulated brain database. NeuroImage, 1997, 5(4): S425

Liang LIAO received his B.S., M.S. and Ph.D degrees respectively from Northwestern Polytechnical University, Southwest Jiaotong University and South China University of Technology, all in Electrical Engineering and Information Technology. Currently, he is a postdoctoral researcher at Shaanxi Provincial Key Laboratory of Speech and Image Information Processing. His research interests include pattern recognition, computer vision, machine learning and intelligent signal processing. He has published more than 20 papers.

Yanning ZHANG received her B.S. degree from Dalian University of Science and Engineering in 1988, and her M.S. and Ph.D degrees from Northwestern Polytechnical University in 1993 and 1996, respectively. She is currently the assistant dean and professor of School of Computer Science, Northwestern Polytechnical University. In 2009, she served as the organizing chair of the Ninth Asian Conference on Computer Vision (ACCV2009). She has won two first class awards from Shaanxi province and several science and technology innovation awards from the Commission of Science, Technology and Industry for National Defense. Her research interests focus on signal and image processing, computer vision, and pattern recognition. She has published more than 200 papers in international journals, conference proceedings, and Chinese key journals.