Knowledge-Based Systems 89 (2015) 641–653

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A manifold learning framework for both clustering and classification

Weiling Cai *

Department of Computer Science & Technology, Nanjing Normal University, Nanjing 210097, PR China

ARTICLE INFO

Article history:
Received 12 February 2015
Received in revised form 4 September 2015
Accepted 8 September 2015
Available online 14 September 2015

Keywords:
Pattern recognition
Clustering learning
Classification learning
Bayesian theory
Manifold Learning

ABSTRACT

In recent years, a great deal of manifold clustering algorithms was presented to identify the subsets of the manifolds data. Meanwhile, numerous classification algorithms were also developed to classified data shaped in the form of manifold. However, nearly none of them pay attention to the statistical relationship between the manifold structures and class labels, thus failing to discover the knowledge concealed in data. In this paper, a manifold learning framework for both clustering and classification is presented, which involves two steps. In the first step, the clustering through ranking on manifolds is executed to explore structures in data; in the second step, the class posterior probability is calculated by using the Bayesian rule. The core of this framework lies in employing the Bayesian theory to establish the relationship between manifolds and classes thus creates a bridge between clustering learning and classification learning. Our new manifold learning framework is interesting from a number of perspectives: (1) our algorithm can perform manifold clustering learning which can auto-determine the clustering parameters without manual determining; (2) our algorithm can perform manifold classification learning which models the posterior probabilities $p(\omega_l|x_i)$ by using the Bayesian rule; (3) our algorithm can provide the statistical relationship between the manifold structure and the given classes. Encouraging experimental results are obtained on 2 artificial and 16 real-life benchmark datasets.

1. Introduction

Data mining [1–3] is the discovery of interesting relationships and characteristics that may exist implicitly in data. Clustering and classification are two primary data-mining techniques [4,5]. The clustering approaches such as K-Means [6], Fuzzy C-Means (FCM) [7] and Gaussian Mixture Model [8] are widely utilized to discover the hidden structure in data. Whereas the classification approaches such as Multi-layer Perceptron (MLP) [9] and Support Vector Machines (SVM) [10,11] are successfully applied to determine the class labels of unseen samples. To fuse the advantages of clustering and classification together, numerous researchers studied on how to design a single approach for both clustering and classification. To bridge clustering and classification, Setnes and Babuŝka [12] proposed Fuzzy Relational Classifier (FRC) which attempted to utilize the fuzzy composite operators to construct the relationship between the cluster structures and classes. To enhance the robustness of FRC, in one of our previous works, we developed Robust Fuzzy Relational Classifier (RFRC) [13] by replacing FCM and hard class labels with Kernelized FCM (KFCM) [14,15] and soft labels, respectively. Another famous classifier is Radial Basis Function neural networks (RBFNN) [16,17] which extracts significant information from the observed data to construct its hidden layer.

However, all above algorithms are relatively suitable for the data shaped in the form of point clouds (group), but unsuitable for those data in the form of manifold structure. In real-life world, there are quite a number of data that form paths through a high-dimensional and expose manifold structure. For instance, motion segmentation problem in computer vision, the point correspondences in a dynamic scene can generally be represented as manifold; in classification of face images, the faces of person lie on the manifold. For these data exhibiting manifold structure rather than compact shape, a considerable number of clustering algorithms such as Spectral Clustering [18,19] have been presented to identify the subsets of the manifolds data. Numerous research studies proved that incorporating the structure information into a classifier can enhance its generalization ability, and this research finding is consistent with the famous No Free Lunch (NFL) theorem [20]. In the last decade, a number of manifold or subspace classification algorithms such as Plane-Gaussian Function Networks (PGFN) [21], Laplacian Regularized Least Square Classification (LapRLSC) [22,23] and Laplacian SVM (LapSVM) [24] were presented. These algorithms only attempt to integrate the manifold or subspace distribution information into the classification model.

* Tel./fax: +86 25 84441669.
E-mail address: caiwl@nuaa.edu.cn

http://dx.doi.org/10.1016/j.knosys.2015.09.010
0950-7051/© 2015 Elsevier B.V. All rights reserved.

However, nearly none of them pay attention to the underlying relationship between the manifold distribution and given classes, thus unable to discover the knowledge concealed in data. As a result, an open and challenging problem is to design a framework for manifold data with the goal of combining the advantages of clustering and classification and meanwhile revealing the statistical relationship between manifolds and classes.

In this paper, we propose a manifold learning framework for both clustering and classification (MCC). MCC aims to discover the manifold structure hidden in data, design an effective and transparent classification mechanism and meanwhile exploit the relationship between manifolds and classes. To achieve these goals, our framework treats the manifold clustering learning and classification learning in a two-step *sequential* manner. In the first step, the clustering through ranking on manifolds is performed to explore structures in data; in the second step, by using the Bayesian rule, the class posterior probability is calculated to give class labels for unseen samples. It is worth mentioning that the number of manifolds (i.e. clusters) has a significant influence on the result of manifold clustering [25–27]. To auto-determine this parameter in our algorithm, the inter-cluster mean distance by ranking on manifolds is maximized and while the intra-cluster mean distance is minimized. As a result, our algorithm can auto-determine the clustering parameters without manual determining. Another key of this framework is to connect the multi-manifold with the given classes employ, and then establish a relationship between them. This relationship creates a bridge between clustering learning and classification learning. Based on such relationship, our framework cannot only group multi-manifold into different clusters, but also make classification decisions for unseen samples. More importantly, this relationship can successfully reflect the probability and statistics meaning between manifold structures and given classes, so that we gain some meaningful insights to make MCC prone to be transparent.

The new manifold learning framework for both clustering and classification is interesting from a number of perspectives:

(1) Our algorithm can perform manifold clustering learning which can auto-determine the clustering parameters without manual determining.
(2) Our algorithm can perform manifold classification learning which models the posterior probabilities $p(\omega_l|x_i)$ by using the Bayesian rule.
(3) Our algorithm can provide the statistical relationship between the manifold structure and the given classes.

The experimental results on both synthetic and real-life datasets all demonstrate the effectiveness and potential of MCC.

The rest of this paper is organized as follows: Section 2 reviews the related works. Section 3 describes the proposed manifold learning framework for both clustering and classification. Preliminary experimental results are shown in Section 4. Finally, we give concluding remarks and future work in Section 5.

2. Related works

There have been several recent related works to inherit the merits of both clustering and classification learning. We review the main works as follows.

2.1. Fuzzy relational classifier

Fuzzy Relational Classifier (FRC) [12] was proposed to provide a transparent alternative to the black-box techniques such as neural networks. As show in Fig. 1, in FRC, FCM is firstly adopted as the clustering criterion to discover the natural structure in data, and its objective function is as follows:

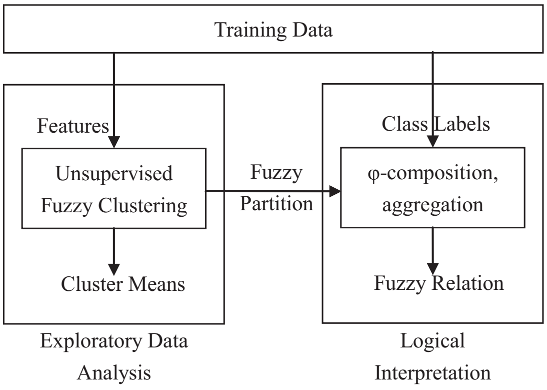


Fig. 1. Training process of FRC and RFRC.

$$J_{FCM}(U,V) = \sum_{j=1}^{c}\sum_{i=1}^{n} u_{ji}^{2}\|\mathbf{x}_i - \mathbf{v}_j\|^2, \tag{1}$$

where $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ and $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c\}$ are the training samples and cluster centers, respectively; and u_{ji} is the fuzzy memberships of $\mathbf{x}_i$ to $\mathbf{v}_j$. By definition, each sample $\mathbf{x}_i$ satisfies the constraint $\sum_{j=1}^{c} u_{ji} = 1$. And then, a relation matrix $\mathbf{R}$ is computed for the obtained fuzzy partition and the given hard class labels. In FRC, FCM is unable to group the datasets consisting of the non-spherical clusters, so that the interpretation of the clustering or classification results may be biased.

Afterwards, we have presented Robust FRC (RFRC) [13] to improve both clustering and classification performance of FRC in our previous work. Specifically, in the clustering phase, the robust Kernelized FCM (KFCM) [14,15] is adopted to replace FCM which can be described as below:

$$J_{KFCM}(U,V) = \sum_{j=1}^{c}\sum_{i=1}^{n} u_{ji}^{m}\|\phi(\mathbf{x}_i) - \phi(\mathbf{v}_j)\|^2, \tag{2}$$

where ϕ is an implicit nonlinear map from the input space to a rather high dimensional feature space. Compared to FCM, KFCM based on RBF kernel is a robust estimator according to M-estimator and is more flexible for clustering non-spherical data. Next, in the classification phase, the soft class label motivated by the fuzzy *k*-nearest-neighbor [28] is employed to replace the hard class label. With the incorporation of both KFCM and the soft class labels, RFRC makes the constructed relation matrix $\mathbf{R}$ more really reflect the relationship between the classes and clusters, and thus significantly boosts the performance of FRC.

It is worth to point out that in FRC and RFRC, the entries in the relation matrix $\mathbf{R}$ lack the statistical meaning, thus it is difficult to judge whether the obtained relationship is really reliable.

2.2. Radial basis function neural networks

Radial Basis Function neural networks (RBFNN) [16,17], as shown in Fig. 2, is a feed-forward multi-layer network. It usually consists of three layers: input layer, hidden layer and output layer. Each basis function Φ_k corresponds to a hidden unit and w_{kl} represents the weight from the *k*th basis function or hidden unit to the *l*th output units.

In the training phase of RBFNN, the basis function Φ_k for each hidden node can be determined by

$$\Phi_k^{RBF}(\mathbf{x}, \mathbf{v}_k) = \exp\left(\frac{-\|\mathbf{x} - \mathbf{v}_k\|^2}{2\sigma^2}\right), \tag{3}$$

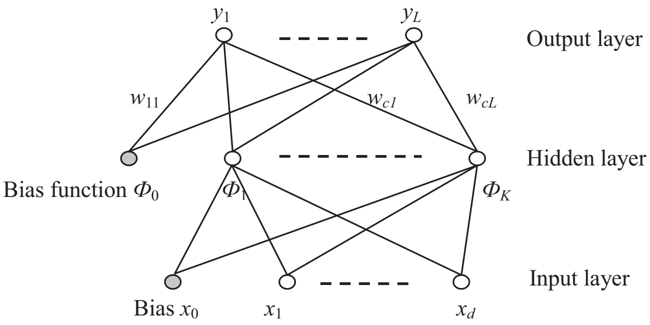


Fig. 2. Architecture of RBFNN.

where $\mathbf{v}_k$ and σ are the centroid and width for each hidden node, respectively. Here $\{\mathbf{v}_k\}$ determined by the clustering algorithm, such as K-Means [6] or FCM [7].Next, by minimizing the Mean Squared Error (MSE) between the target and actual outputs, the connection weights w_{kl} between the hidden and output layers can be calculated.

In RBFNN, the clustering method can ensure the good classification generalization. However, the connecting weights w_{kl} between the hidden and output layers conceal the learned knowledge, which leads to the poor transparency and interpretability for knowledge (representation).

2.3. Plane-Gaussian function networks

Plane-Gaussian Function Networks (PGFN) [21] utilizes K-Plane Clustering (*k*PC) [29] to generate corresponding hyperplane prototypes (similar to point prototypes in FCM). The objective function of *k*PC can be described as a nonconvex minimization problem as follows:

$$J_{kPC} = \sum_{j=1}^{c}\sum_{x_i\in c_j}\left\|w_j^T\mathbf{x}_i - \gamma_j\right\|^2, \tag{4}$$

where c_i is the ith cluster composed by those points closest to the ith plane L_i, w_j and γ_j are the unit normal vector and threshold of the L_i, respectively. The goal of *k*PC is to group n points into c clusters corresponding to their nearest hyperplanes $L = \{L_1, L_2, \ldots, Lc\}$. By using these plane prototypes to replace the cluster centers in RBFNN, a new activation function called Plane-Gaussian function (see Fig. 3) is defined and further developing a corresponding PGF network (PGFN).

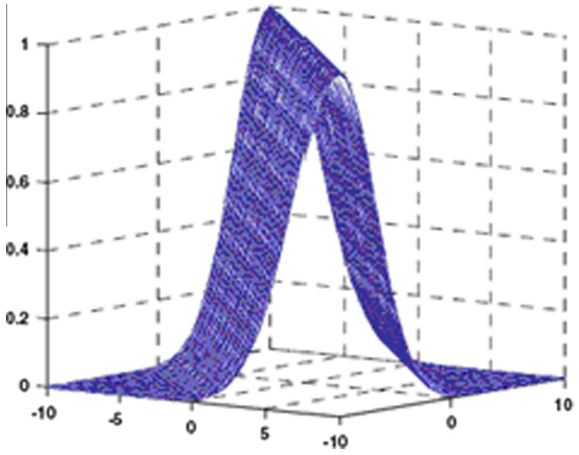


Fig. 3. Plane-Gaussian function.

$$\Phi_k^{PG} = \exp\left(\frac{-\left\|w_k^T\mathbf{x} - r_k\right\|^2}{2\sigma_k^2}\right), \tag{5}$$

where the parameters w_k and γ_i can be obtained by the *k*PC algorithm.

Similar to RBFNN, the connection weights w_{kl} in PGFN is obtained by optimizing the Mean Squared Error (MSE) classification criterion between the target and actual outputs, and thus fails to reflect the logical or statistical relationship between the formed subspaces and given classes.

2.4. Laplacian regularized least-squares classification

In 2006, Belkin et al. [22,23] proposed a framework for data-dependent regularization that exploits the geometry of the probability distribution. Within this general framework, the Laplacian Regularized Least Squares Classification (LapRLSC) was presented which is a natural extension of RLSC.

For a Mercer kernel $K : X \times X \to R$, there is an associated Reproducing Kernel Hilbert Space (RKHS) H_k of function $X \to R$ with the corresponding norm $\|\|_K$. Given a set of labeled examples (x_i, y_i), the objective function of LapRLSC for supervised examples can be formulated as follows

$$\min_{f\in H_K}\sum_{i=1}^{n}(y_i - f(x_i))^2 + r_A\|f\|_K^2 + r_I f^T Lf, \tag{6}$$

where L is the graph Laplacian, r_A controls the complexity of the function in the ambient space while r_I controls the complexity of the function in the intrinsic geometry. The Representer Theorem [30] can be used to show that the solution is an expansion of kernel functions over the labeled data:

$$f^*(x) = \sum_{i=1}^{n}\alpha_i^* K(x, x_i). \tag{7}$$

Substituting this form in the problem, a convex differentiable objective function of n-dimensional variable α is obtained:

$$\alpha^* = \min_{f\in H_K}(Y - K\alpha)^T(Y - K\alpha) + r_A\alpha^T K\alpha + r_I\alpha^T KLK\alpha, \tag{8}$$

where K is the $n \times n$ Gram matrix over labeled examples, Y is an n-dimensional label vector. The derivative of this objective function vanishes at the minimizer:

$$(Y - K\alpha)^T(-K) + (r_A K + r_I KLK)\alpha = 0, \tag{9}$$

which leads to the following solution:

$$\alpha^* = (K + r_A I + r_I LK)^{-1}Y \tag{10}$$

To sum up, by introducing an extra term of $f^T\ Lf$ into the objective function of RLSC, LapRLSC is naturally resulted, with the incorporation of the structure information of given samples. However, LapRLSC does not take into account the relationship between such manifold and the classes, thus making the classification result lack of explanation about the structure distribution.

3. Our algorithm

Our framework treats the manifold clustering learning and classification learning in a *sequential* manner. At first, the clustering through ranking on manifolds is executed to explore structures in data, then the relationship between the multi-manifold and classes is constructed by using the cluster partition and given class labels, and finally the class posterior probability is calculated by using the Bayesian rule.

3.1. Key point of our algorithm

3.1.1. How to find manifold structures

Within our framework, as long as the manifold clustering technique can result in a partition of samples, it can be adopted in the phase of clustering learning. Without loss of generality, in our framework, a clustering algorithm through ranking on manifolds [31] is employed. It can achieve better clustering results than the state of the art clustering technique, such as Spectral Clustering.

The main idea of this manifold clustering algorithm is to consider two points to be identical if they both have identical distance on the manifold to other points. Furthermore, based on the designed similarity metric, a subset of points is selected to be used as seeds for clusters. By means of these seeds, a partition of samples can be yielded and the clustering membership for new samples can also be calculated.

Specifically, in order to rank on manifold structure, the activation U of all points is used as similarity measure between the points [32]

$$U = \beta(I - \alpha S)^{-1} = [u_1^T, \quad \ldots, \quad u_n^T], \tag{11}$$

$$S = D^{-1/2} W D^{-1/2}, \tag{12}$$

$$W_{ij} = \begin{cases} \exp\left(-\|x_i - x_j\|^2/(2\sigma^2)\right) & if\ i \neq j \\ 0 & if\ i = j \end{cases} \text{ and } \quad D_{ij} = \begin{cases} \sum_{r=1}^{n} W_{ir} & if\ i = j \\ 0 & if\ i \neq j \end{cases}, \tag{13}$$

where $0 < \alpha < 1$ and $\beta = 1 - \alpha$. U_{ii} is the largest number in column i and the remaining values in u_i get smaller the further the points are away from the centroid point x_i according to the underlying intrinsic structure. Here the column i of matrix U is symbolized by the column vector u_i^T.

The rank based distance D_{Mij} between any points x_i and x_j are given as follows:

$$D_{Mij} = d_M(x_i, x_j) = 1 - v_i v_j^T, \tag{14}$$

where $V = [u_1^T \|u_1^T\|^{-1}, \quad \ldots, \quad u_n^T \|u_n^T\|^{-1}] = [v_1^T, \quad \ldots, \quad v_n^T]$ and by definition, $\|v_i\| = 1$. The meaning of this distance measure is that two points on a manifold are identical, iff the order of distances between all other points in the training set are identical. Under these conditions, $v_i v_j^T = 1$ and $d_M(x_i, x_j) = 0$. Conversely, if the point x_i has completely different distances along U to other points in the training data as does point x_j, then $d_M(x_i, x_j) = 1$, because $v_i v_j^T = 0$.

Using this rank-based distance, we want to pick the cluster seed points $\{x_{c1}, x_{c2}, \ldots, x_{ck}\}$ that are most similar to one another, while at the same time most different from points in other clusters. To determine x_{c1}, assume that all points to be a cluster and x_{c1} is the point that is closest to all other points. The average distance D_M between each point x_j and all the other points is defined and x_{c1} is the point that has the minimum value of D_{Mj}

$$D_M = \frac{1}{n}\left[\sum D_{M1}^T, \quad \ldots, \quad \sum D_{Mn}^T\right] \text{ and } D_{Mj} = [D_{Mj1}, \quad \ldots, \quad D_{Mjn}]. \tag{15}$$

To find x_{c2}, each element of $D_M^{(1)} = D_M$ is multiplied by the $1 - D_{Mc1}$ and a re-weighted vector $D_M^{(2)}$ is obtained. Let $D_M^{(n)}$ denote the nth re-weighting of the D_M vector. Re-weighting the vector gives all the points that were similar x_{ci} a large value and all the point are different a small value. Again we choose the point that is similar to all the other points, which is the point that has the minimum value of D_{Mj}. This procedure of re-weighting and finding the most similar point continues until c points have been found. For a more depth discussion about computing clustering membership $p(c_k|\mathbf{x})$ for new point x and the corresponding rigorous proof, please see [31,32].

Algorithm 1. Find centroid-points.

```
    Input: Matrix $V$, number of clusters $K$
    Output: Indices of the cluster $c1, \ldots, c_K$
1. $n \leftarrow 1$, compute $D_M^{(1)} = D_M$ from $V$.
2. $l_1 \leftarrow$ index of the example with minimum of $D_M^{(1)}$
3. $wt \leftarrow (1 - D_{Ml1})$
4. for $2 \leqslant n \leqslant K$ do
5.    $l_n \leftarrow$ index of the example with minimum of $D_M^{(n)}$
6.    $wt \leftarrow (1 - D_{Mln}).^{*}wt$
7. $D_M^{(n+1)} \leftarrow wt.^{*}D_M^{(n)}$
8. end for
```

To explain this algorithm, we take *Wine dataset* as an example to illustrate the procedure for finding the clustering seeds. *Wine* dataset contains 178 samples, from the 3 classes, with 13 attributes. On this dataset, the mean distance D_M for each sample is shown in Fig. 4(a) where the horizontal coordinates are the sample numbers, and the longitudinal coordinates are the values of D_M. To find the first cluster seed, the sample x_{24} with minimum value of D_M is selected as the first cluster. And then, the mean distance D_M after the first re-weighting is illustrated in Fig. 4(b). After reweighting, the mean distance D_M of the points that are similar to the sample $x_{24\psi}$ becomes large (with our similarity measure) and the points that are different from x_{24} become small. In the second cycle, the second cluster seed x_{166} is found by Step5 in Algorithm1. Similarly, Fig. 4(c) gives the mean distance D_M after the second re-weighting and the third cluster seed x_{93} is determined. Through the above process, the representative clustering centers can be found by the clustering method through ranking on manifold, thus the structure of data is effectively explored.

3.1.2. How to establish a bridge between the manifold and classes

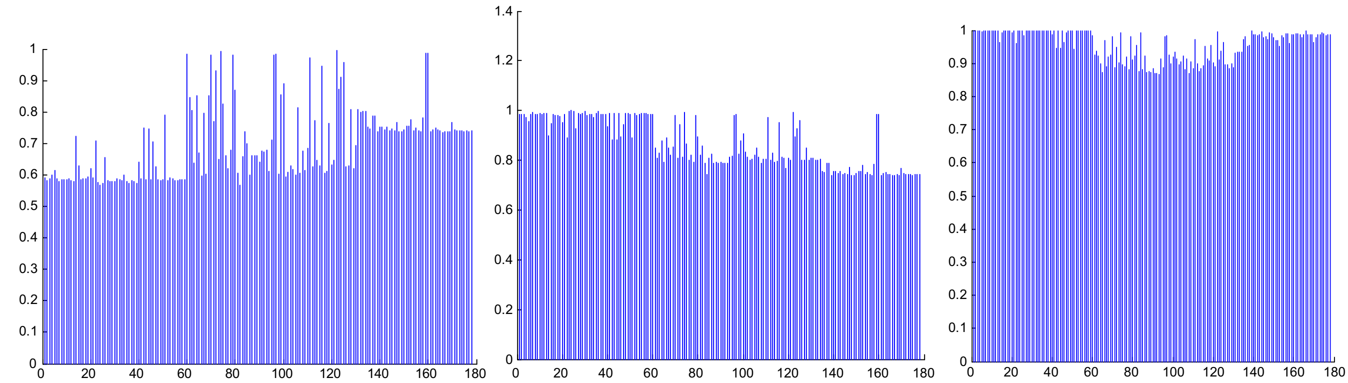
Assume that the data samples have already been assigned to different manifold groups, the next problem is how to connect these formed manifold clusters with the given classes. To describe such relationship between the obtained manifold clusters and given classes, we build a $K \times L$ matrix $\mathbf{P}$:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1L} \\ p_{21} & p_{22} & \cdots & p_{2L} \\ \cdots & \cdots & \cdots & \cdots \\ p_{K1} & p_{K2} & \cdots & p_{KL} \end{bmatrix}, \tag{16}$$

where K is the number of manifolds, and L is the number of classes. Based on Bayesian theory, the element p_{kj} of $\mathbf{P}$ can be represented as $p(\omega_l|c_k)$ which indicates the statistical relationship between the kth manifold structure and the lth classes. The value of $p(\omega_l|c_k)$ can be computed according to Bayesian rule [33]:

$$p(\omega_l|c_k) = \frac{p(\omega_l, c_k)}{p(c_k)}, \tag{17}$$

where $p(c_k)$ is the prior probability of the kth manifold and $p(\omega_l, c_k)$ is the joint distribution of the kth manifold and the lth class. Above them, $p(c_k)$ can be computed in terms of $Num(\mathbf{x} \in c_k)/n$, indicating



(a) n=1 (b) n=2 (c) n=3

Fig. 4. D_M in each step of finding centroid points: In each step the mean distances change as the D_M vector is re-weighted.

the proportion of the samples in the kth manifold. Similarly, $p(\omega_l, c_k)$ can be computed by $Num(\mathbf{x} \in \omega_l \text{ and } \mathbf{x} \in c_k)/n$, which means that the proportion of the samples lies in the kth manifold and the lth class. Furthermore, $p(\omega_l|c_k)$ can be rewritten as

$$p(\omega_l|c_k) = \frac{Num(\mathbf{x} \in \omega_l \text{ and } \mathbf{x} \in c_k)}{Num(\mathbf{x} \in c_k)}. \tag{18}$$

From (18), it can be observed that the value of $p(\omega_l|c_k)$ is proportional to the number of samples in manifold c_k from the class l. In conclusion, by using the Bayesian theory, the matrix $\mathbf{P}$ can indeed build a bridge from the formed manifolds to the given classes.

3.1.3. How to construct a classification mechanism

Next, we employ the Bayesian theory to design a classification mechanism relying on the relation matrix $\mathbf{P}$. In the classification learning, when the posterior probabilities $p(\omega_l|\mathbf{x}_i)$ can be modeled, the output class label $f(\mathbf{x}_i)$ can be determined as

$$f(\mathbf{x}_i) = \arg\max_{1\leqslant l\leqslant L} p(\omega_l|\mathbf{x}_i). \tag{19}$$

To introduce the manifold information into $p(\omega_l|\mathbf{x}_i)$, we resort to the formed manifold partition $\{c_k\}$ to reformulate $p(\omega_l|\mathbf{x}_i)$ through the total probability theorem as

$$\begin{aligned} p(\omega_l|\mathbf{x}_i) &= \sum_{k=1}^{K} p(\omega_l, c_k|\mathbf{x}_i) \\ &= \sum_{k=1}^{K} p(c_k|\mathbf{x}_i)p(\omega_l|c_k, \mathbf{x}_i) \\ &= \sum_{k=1}^{K} p(c_k|\mathbf{x}_i)p(\omega_l|c_k), \end{aligned} \tag{20}$$

where ω_l denotes the lth class, c_k represents the kth manifold, $p(c_k|\mathbf{x}_i)$ represents the posterior probabilities of the presence of corresponding samples and $p(\omega_l|c_k)$ denotes the manifold-cluster posterior probabilities of class membership. Notice that $p(\omega_l|c_k, \mathbf{x}_i)$ has no relationship with $\mathbf{x}_i$, and thus can be simplified as $p(\omega_l|c_k)$.

3.2. Analysis of our algorithm

3.2.1. Analysis about the manifold clustering

The manifold clustering used here considers two points to be identical if they both have identical distance, on the manifold, to all other points. Such similarity metric produces a number of properties [31,32] that differentiate the manifold clustering algorithm in this paper from Spectral Clustering. Specifically,

(1) The clustering algorithm identifies points that are most representative of each cluster.
(2) The clustering algorithm identifies points that are outliers from all other points.
(3) The clustering algorithm identifies points that are outliers within each cluster.

These properties can lead to significant improvements in the model quality and give additional insights into the data. In contrast, Spectral Clustering does not have these characteristics.

Moreover, the parameters σ, α, and K in the clustering algorithm can be further optimized. Let c_j be the set of points that belong to cluster j. With the matrix D_M, the intra-cluster mean distance of cluster j can be defined as

$$\bar{D}_M^{jj} = E\left[D_M(c_j,\ c_j)\right], \tag{21}$$

where $D_M(c_j, c_j)$ is all the elements of D_M which corresponding to columns and rows of points c_j, and $E[]$ is the mean value of these. Similarly, the inter-cluster mean distance between cluster j and k can be given as

$$\bar{D}_M^{jk} = E\left[D_M(c_j,\ c_k)\right] \tag{22}$$

Using the above definition, the framework for estimating σ, α and K is established

$$\Omega(K) = \max_{\sigma,\alpha,K}\left[E\left[\bar{D}_M^{jk}\right]_{\left\{\substack{j=1,\dots,K\\ k=1,\dots,K\\ j\neq k}\right\}} - E\left[\bar{D}_M^{jj}\right]_{\{j=1,\dots,K\}}\right] \tag{23}$$

The goal of (23) is to maximize the inter-cluster mean distance while minimize the intra-cluster mean distance. By solving this optimization problem, the clustering parameters can be resulted. The optimization problem in Eq. (23) is non-linear, and currently we use a brute force procedure for solving it. Namely, for each $K = 2, 3, 4, \dots, K_{max}$, we use the *Matlab* function *fminbnd* to perform a two dimensional optimization in α and σ to maximize $\Omega(K)$. We then choose the number of clusters c by finding the maximum of $\Omega(2), \dots, \Omega(K_{max})$, and use the α and σ associated with this number of clusters. Notice that this is not the optimal solution, but an approximation that works well in most of the practical cases.

3.2.2. Analysis about the relation matrix P

Our algorithm MCC aims at establishing a relation matrix $\mathbf{P}$ to naturally reflect the statistical relationship between the formed manifold structures and the given classes. Given a $K \times L$ matrix $\mathbf{P}$

where K and L are the number of the clusters and the classes, respectively.

$$\mathbf{P} = \begin{bmatrix} p(\omega_1|c_1) & p(\omega_2|c_1) & \dots & p(\omega_L|c_1) \\ p(\omega_1|c_2) & p(\omega_2|c_2) & \dots & p(\omega_L|c_2) \\ \dots & \dots & \dots & \dots \\ p(\omega_1|c_K) & p(\omega_2|c_K) & \dots & p(\omega_L|c_K) \end{bmatrix}. \quad (24)$$

The jth *row*-elements of $\mathbf{P}$, $[p(\omega_1|c_j), p(\omega_2|c_j), \dots, p(\omega_L|c_j)]$, uncover the relationship between the jth manifold structure and all the classes. Whereas, the lth *column*-elements, $[p(\omega_l|c_1), p(\omega_l|c_2), \dots, p(\omega_l|c_K)]$, reflect the relationship between the lth class and all the manifold structures. The *row*-elements of $\mathbf{P}$ satisfy the constraint that $\sum_{l=1}^{L} p(\omega_l|c_j) = 1$, while its *column*-elements do not.

By analyzing the row elements of $\mathbf{P}$, we can capture some distribution knowledge about the formed manifold. For the jth row corresponding to the jth manifold, if there is one and only one row element with the value of 1 and the others with 0, the manifold is pure and the samples falling into the manifold consistently belong to the same class; if the multiple non-zero elements exist in this row, the corresponding manifold is composed of the samples from multiple classes.

Similarly, by analyzing the column elements of $\mathbf{P}$, we can also know some structural knowledge from the column-elements of $\mathbf{P}$. Only one non-zero column-element implies that corresponding class contains only one manifold; while the multiple non-zero column elements implies that the samples in the corresponding class are scattered over multiple manifold structures.

Consequently, the $\mathbf{P}$ plays an important role in the discovery and formulation of the structural knowledge in given dataset and thus makes MCC prone to be transparent.

3.2.3. Analysis of the manifold classification

Based on the manifold clustering results and the relation matrix $\mathbf{P}$, the posterior probability of manifold classification can be calculated. Such classification mechanism has a number of perspectives which make it very different from the existing classifiers:

(1) the distribution information of manifold and sub-manifold are embedded into the classification, so that the generalization ability of classification learning can be boosted to some extent. The famous No Free Lunch (NFL) theorem and numerous works demonstrate that exploiting the structure knowledge can enhance the generalization ability of classification. Consequently, MCC follows this line and integrates the clustering information into the classifier design as much as possible.

(2) the relation matrix $\mathbf{P}$ used in classification can provide some explanation about the classification result from the viewpoint of the relationship between manifold clusters and given classes. It results that MCC prone to be transparent, and it further provides a chance to extract classification rule.

(3) the classification result is yielded in form of the posterior probability which can reflect the classification reliability of each unseen sample.

3.2.4. Analysis about time complexity

The time complexity of the proposed algorithm includes two parts: one part is for the manifold clustering (described in Algorithm1) to find centroid-points and form a partition of samples, and the other part is for the relation establishing to compute the relation matrix $\mathbf{P}$ between the clustering partition and given classes.

For the first part, according to Formula (11)–(13), the time complexity for computing the similarity measure U is $O(n^3)$ where n is the number of training samples, and hence based on the U, the rank based distance matrix D_M also needs the complexity of $O(n^3)$. In addition, in terms of Algorithm 1, the time complexity for finding centroid-points is $O(Kn)$ where K is the number of cluster seeds. Since the number n is generally much larger than K, the total time complexity of the first part is $O(n^3)$.

For the second part, the complexity for computing the $p(\omega_l|\mathbf{x}_i)$ is $O(n)$ in terms of Formula (18), and based on $p(\omega_l|\mathbf{x}_i)$, the computation for the relation matrix $\mathbf{P}$ needs $O(nKL)$ where L is the number of classes.

In conclusion, the total time of the proposed algorithm is $O(n^3 + NKL)$. Since the number n is respectively larger than K and L, the final time complexity is $O(n^3)$.

3.3. Summary of the MCC

In this subsection, the whole process of the proposed MCC algorithm is summarized and again it is worth mentioning that MCC holds the following characteristics:

(1) Our algorithm can perform manifold clustering learning which can auto-determine the clustering parameters without manual determining.

(2) Our algorithm can perform manifold classification learning which models the posterior probabilities $p(\omega_l|x_i)$ by using the Bayesian rule.

(3) Our algorithm can provide the statistical relationship between the manifold structure and the given classes.

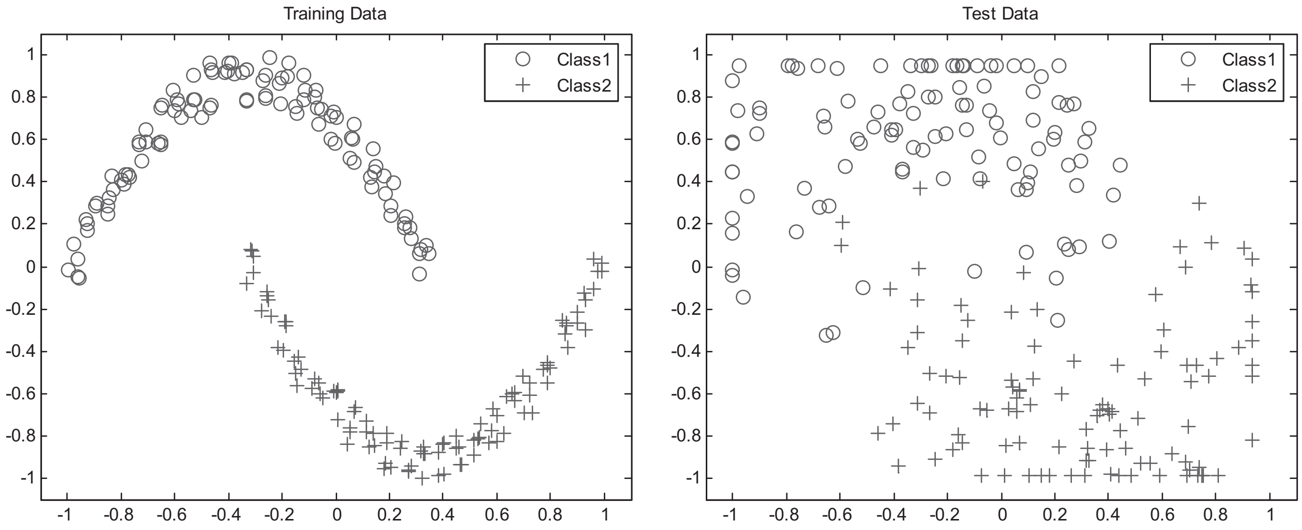
Algorithm 2. MCC.

```
Input: training dataset $\{\mathbf{x}_i, \mathbf{y}_i\}$, number of clusters $K$
Output: cluster and class memberships for new samples
Training Phase:
1. Use Algorithm 1 to find centroid-points and form a partition
   of samples.
2. Establish the relation matrix $\mathbf{P}$ between the clustering
   partition and given classes according to (16)–(18).
Test Phase:
1. For a test sample $\mathbf{x}$, compute the cluster membership
   degrees $p(c_k|\mathbf{x})$.
2. Compute the class memberships $p(\omega_l|\mathbf{x})$ using $\mathbf{P}$ and $p(c_k|\mathbf{x})$
   according to (20).
3. Determine the output class label $f(\mathbf{x})$ in items of (19).
```

4. Experiments

4.1. Synthetic data

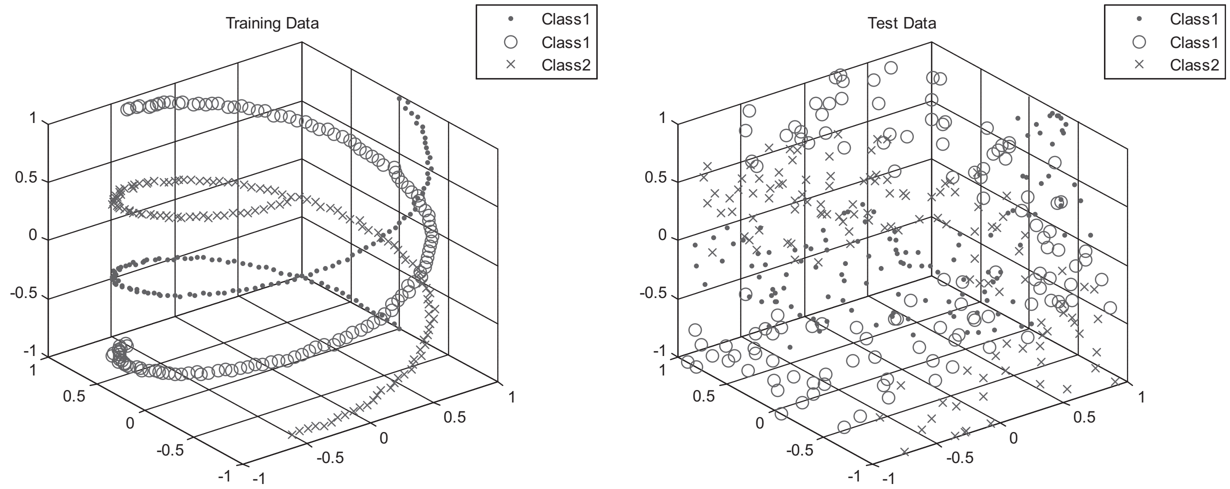
In this experiment, we consider the two-moon and spiral synthetic datasets. The training and test datasets of two-moon and spiral are given in Figs. 5 and 6, respectively. The two-Moon dataset has two manifolds denoted by 'O' and '+', respectively and each manifold belongs to single class. The spiral dataset includes three manifolds represented by '·', 'O' and '×', respectively. The data falling into the manifold '·' and 'O' belong to Class1 and the data scatter into the manifold '×' belong to Class2. That is to say, Spiral dataset consists of two classes, Class1 contains one manifold and Class2 contains two manifolds. Moreover, to evaluate the performance on out-of-sample, the test data of these two datasets are generated by distorting the datasets by adding a uniformly distributed random number within ±0.2 to each coordinate.



(a) Training data of Moon data

(b) Test data of Moon data

Fig. 5. Training data and test data of Moon data.



(a) Training data of Spirals data

(b) Test data of Spirals data

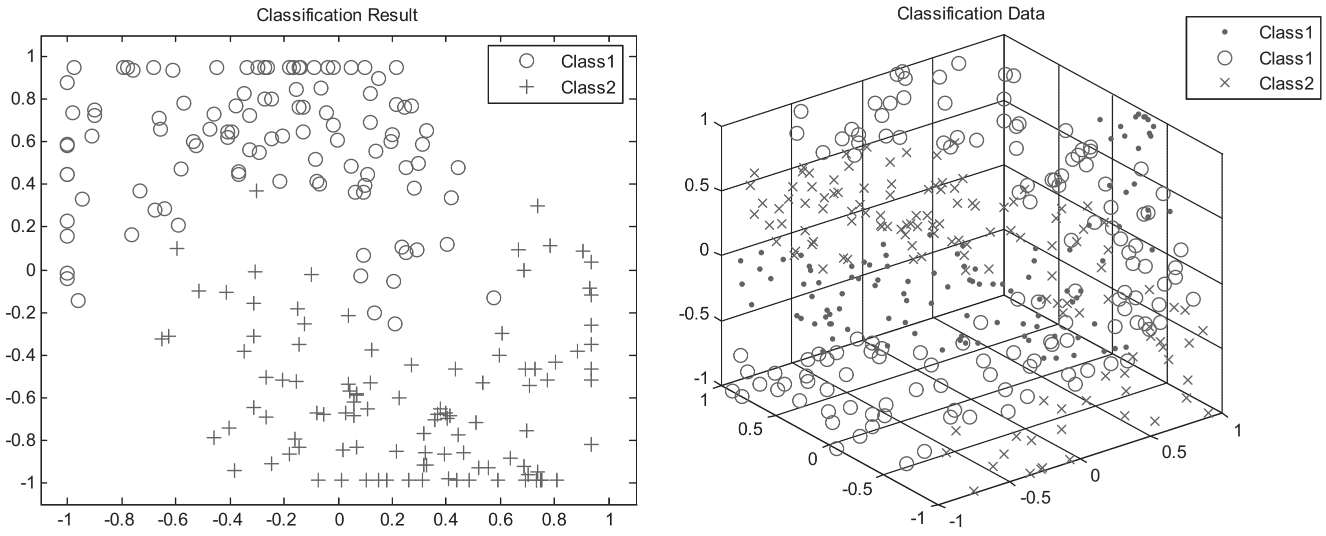
Fig. 6. Training data and test data of Spirals data.

To show the manifold clustering performance of MCC, Fig. 7(a) and (b) respectively give the clustering results obtained on Moon and Spirals datasets. Specifically, as shown in Fig. 7(a), two manifold clusters ‘O’ and ‘+’ are formed on Moon data; while, in Fig. 7 (b), three manifold clusters ‘·’, ‘O’ and ‘×’ are shaped. By comparing these figures with Figs. 5(b) and 6(b), we can see that the manifold structures hidden in data are identified correctly by MCC. It is worth emphasizing that although the test data is distorted to some extent and its distribution is different from the training data, the learned manifold cluster-models in MCC still successfully assign test data to manifold clusters without re-learning.

To further compare MCC with the related algorithms RBFNN, RFRC, PGNG, and LapRLSC, the relation matrixes **P**s and the classification accuracies on Moon and Spirals datasets are presented in Tables 1 and 2. At first, let us analyze the relationship between structures and classes revealed by RBFNN, RFRC, PGNG, and MCC. Notice that LapRLSC cannot represent such relationship explicitly, so this item is blank in Table 1.

(1) In RBFNN and PGNG, the weights connecting hidden layers with output layers constitute a $c \times L$ matrix, which has a similar formulation with the **P** in MCC. However, these weights are obtained by minimizing the Mean-squared Errors (MSE). As a result, the weights can be set to any values and fail to reveal any information between structure and classes. As shown in Table 1, the weights -0.24 and -0.25 in RBFNN are negative which are unsuitable to express the logical relationship between the distributions and classes.

(2) In RFRC, the relation matrix **R** between structures and class also has a similar formulation with the **P** in MCC. However, the values of the **R** are calculated by fuzzy composite operator, as a result, these values lack the statistical property and fail to exhibit the reliability of the obtained relationship. For example, in Table 1, it can be observed that the relation value $r11$ between Cluster1 and Class1 is 0.26, and while the value r_{22} between Cluster2 and Class2 is 0.32. Although the value 0.32 is larger than 0.26, we cannot conclude that the relation embodied by $r11$ is more reliable than that of $r22$.



(a) Manifold clustering result on Moon data

(b) Manifold clustering result on Spirals data

Fig. 7. Manifold clustering result.

Table 1
Relation matrices and classification accuracies obtained on Moon data.

Index	RFRC	RBFNN	PGFN	LapRLSC	MCC
Relation matrix P	$\begin{array}{rcc} & class1 & class2 \\ cluster1 & 0.26 & 0.02 \\ cluster2 & 0.02 & 0.32 \end{array}$	$\begin{array}{rcc} & output1 & output2 \\ hidden1 & 1.14 & -0.24 \\ hidden2 & -0.25 & 1.15 \end{array}$	$\begin{array}{rcc} & class1 & class2 \\ cluster1 & 0.25 & 0.28 \\ cluster2 & 0.30 & 0.27 \end{array}$	——	$\begin{array}{rcc} & class1 & class2 \\ cluster1 & 1 & 0 \\ cluster2 & 0 & 1 \end{array}$
Classification accuracy	84.76%	93.81%	55.24%	93.81%	95.71%

Table 2
Relation matrices and classification accuracies obtained on Spirals Three.

Index	RFRC	RBFNN	PGFN	LapRLSC	MCC
Relation matrix P	$\begin{array}{rcc} & class1 & class2 \\ cluster1 & 0.04 & 0.02 \\ cluster2 & 0.31 & 0.05 \\ cluster3 & 0.30 & 0.05 \end{array}$	$\begin{array}{rcc} & output1 & 2 \\ hidden1 & 0.30 & 0.32 \\ hidden2 & 0.47 & 0.15 \\ hidden3 & 0.46 & 0.15 \end{array}$	$\begin{array}{rcc} & output1 & 2 \\ hidden1 & 1.48 & -1.18 \\ hidden2 & -0.48 & 0.89 \\ hidden3 & -0.20 & 0.68 \end{array}$	——	$\begin{array}{rcc} & class1 & class2 \\ cluster1\,. & 1 & 0 \\ cluster2\ o & 1 & 0 \\ cluster3\,* & 0 & 1 \end{array}$
Classification Accuracy	24.60%	66.67%	67.20%	66.67%	85.45%

(3) Different from RFRC, RBFNN and PGNG, the matrix **P** in MCC is constructed by the Bayesian theory, and thus can naturally reflect the statistical relationship between the formed manifold structures and the given classes. For Spiral data, the first column elements of **P** in Table 2 indicates that the samples falling into Cluster1 and 2 belong to Class1 and while its second column elements imply that the samples in Cluster3 belong to Class2, such relationship between clusters and classes accords with the real relationship shown in Fig. 6 (b). Similar analysis can be made on Moon data.

Next, we compare the classification accuracies obtained by RFRC, RBFNN, PGNG, LapRLSC and MCC.

(1) The common point of RBFNN and RFRC is that the clustering methods FCM and KFCM used in their training phase all assume a compact shape for the data and thus fail to handle data that exposes a manifold structure, i.e. data is not shaped in the form of point clouds. As a result, the formed clusters in RBFNN and RFRC are almost incorrect, thus RBFNN and RFRC obtain the relatively low classification accuracies 93.81% and 84.76% on Moon data, and 66.67% and 24.60% on Spirals data, respectively. It is worth pointing out in RFRC, when the formed cluster is impure, the relation values between this cluster and all classes are close to 0 and thus do not contain any information. For example, as presented in Table 2, the first row elements [0.04 0.02] of relation matrix **R** approach 0, indicating that the relationship between the obtained Cluster1 and all the classes is very unreliable. Due to the incorrect clustering centers and unreliable relationship, RFRC is unable to make the decision for most test samples, and thus just achieves the accuracy of 24.60%.

(2) Different from RFRC and RBFNN, PGFN adopts the *k*PC rather than FCM to generate hyperplane prototypes and the corresponding activation function. However, the low classification accuracies of 55.24% and 67.20% on Moon and Spirals datasets demonstrate that PGFN cannot successfully handle these datasets. Especially, for Moon dataset, the values of weight elements obtained by PGFN are very close to each other. For example, 0.25 is close to 0.28 and 0.30 is close

Table 3
Relation matrices **P** and classification accuracies obtained on USPS dataset when the clustering numbers is set to 4, 6, and 8, respectively.

Index	Number of clusters = 4	Number of clusters = 6	Number of clusters = 8
Relation matrix P	$\begin{array}{r} \\ cluster1 \\ cluster2 \\ cluster3 \\ cluster4 \end{array}\begin{array}{c} \begin{array}{cccc} class1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0.93 & 0.02 & 0.02 & 0.03 \\ 0 & 0.96 & 0 & 0.04 \\ 0 & 0.42 & 0.57 & 0.01 \\ 0 & 0 & 0 & 1.00 \end{bmatrix} \end{array}$	$\begin{bmatrix} 0.96 & 0.01 & 0 & 0.03 \\ 0 & 0.98 & 0.02 & 0 \\ 0 & 0.86 & 0.14 & 0 \\ 0 & 0.75 & 0 & 0.25 \\ 0 & 0.03 & 0.97 & 0 \\ 0 & 0 & 0.01 & 0.99 \end{bmatrix}$	$\begin{bmatrix} 0.97 & 0 & 0 & 0.03 \\ 0 & 0.99 & 0.01 & 0 \\ 0 & 0.91 & 0.09 & 0 \\ 0 & 0.78 & 0 & 0.22 \\ 0 & 0.05 & 0.95 & 0 \\ 0 & 0 & 0.13 & 0.88 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0.01 & 0.99 \end{bmatrix}$
Classification Accuracy	93.25%	94.87%	96.00%

to 0.27, as a result, the discriminant ability of PGFN is significantly decreased and lead to a poor classification performance of 55.24%.

(3) LapRLSC obtains the relatively high classification accuracy of 93.81% on Moon dataset, this result indicates that the manifold structure introduced into LapRLSC play an important role in classification. However, when the dataset is very complex such as the dataset Spirals, its classification ability is decreased and achieves the accuracy only 66.67%.

(4) As illustrated in Fig. 7, MCC obtains the manifold clustering results that are relatively close to the original clustering distribution given in Fig. 5(a) and (b), implying that the our algorithm can effectively identify the underlying manifold structures hidden in data. The obtained matrix **P** on Moon data indicates the strong connection between Cluster1 and Class 1 and similarly, between Cluster2 and Class2; the same analysis can be made on Spiral data, it can be seen from the **P** that the strong connection exist between Cluster1 and Class 1 (similarly, between Cluster2 and Class1, between Cluster3 and Class2). Such relationship matches with the real relationship listed in Fig. 5. Based on the real manifold cluster and correct relationship matrix **P**, MCC achieves the good accuracies of 95.71% and 85.45% on Moon and Spiral data, respectively.

4.2. Handwritten digits recognition

USPS [34] is handwritten 16 × 16 digits dataset. In our experiment, the digits 1, 2, 3 and 4 are used and there are 200 examples for each class, for a total of 800. We set the clustering number c from 4 to 8, and analyze the obtained relation matrices **P** in our algorithm under the different settings.

From the relation matrix P and classification results listed in Table 3, we can make the following analyses to further understand our proposed algorithms:

(1) No matter what value of the cluster number c is, the *row-*elements of **P** in Table 3 satisfies the constraint $\sum_{l=1}^{L} p(\omega_l|c_j) = 1$, indicating the matrices **P**s can indeed reflect the statistical relationship between the formed clusters and classes.

(2) From the row elements of **P** in Table 3, we can discover some characteristics of the formed manifold. For example, when $c = 4$, for the first row [0.93 0.02 0.02 0.03] of **P** corresponding to the first manifold, we can see that this manifold is not pure and 93% samples in this manifold belongs to Digit 1 and other samples from Digit 2, 3 and 4; for the fourth row [0 0 0 1], we can infer that the fourth manifold is pure and the samples in this clusters all belong to the Digit 4.

(3) From the column elements of **P** in Table 3, some characteristics of the classes can also be captured. For example, when $c = 4$, for the first column [0.93 0 0 0] of **P**, it can be obtained that the samples belonging to Digit 1 are all located on the first manifold; the second column [0.02 0.96 0.42 0] implies that the samples from Digit 2 are scattered over the first, second and third manifold structures.

(4) In this experiment, with the increase of the number clusters, the formed manifold tends to be pure, and the classification performance is improved. Specifically, when $c = 4$, for the third cluster, 42% samples are from class 2 and 57% from class 3, so the purity of this cluster is very low; when $c = 6$ and 8, the purity of samples in each cluster is significantly improved. As the clustering purity increases, the logical relationship contained by P becomes more true and reliable, and thus the correct classification rate is effectively improved, increased from 93.25% to 96.00%.

Furthermore, by analyzing the matrices **P**, we can find out those samples which are hard to be classified. When $c = 8$, we find that for each manifold cluster, most samples are from the same class and few samples from the other classes. For example, in the first row of **P**, 97% samples in Cluster1 belong to Digit 1 and only 3% samples in Cluster1 belong to Digit 4. Intuitively, such 3% samples are hard to be classified correctly, the reason is that such 3% samples are similar to the other 97% samples in Cluster1, but in fact they fall into Digit4 rather than Digit1. We list some part of such samples in Fig. 8. From this figure, we can see that those digits are indeed difficult to be classified, because those digits are rotated, labeled with wrong classes, or otherwise illegibly written. Therefore, we can conclude that the values of P can guide us in looking for those samples difficult to be classified, thus making the classification results has some explanatory.

4.3. Object recognition

To test the algorithms performance in object recognition, we carry out experiments on Coil-20 (http://www.cs.columbia.edu/CAVE) and MPEG-7 dataset (http://www.dabi.temple.edu/~shape/MPEG7/dataset.html). The comparison is made among RFRC, RBFNN, VQ + LVQ3 [35,36] (state of art classifier based on learning

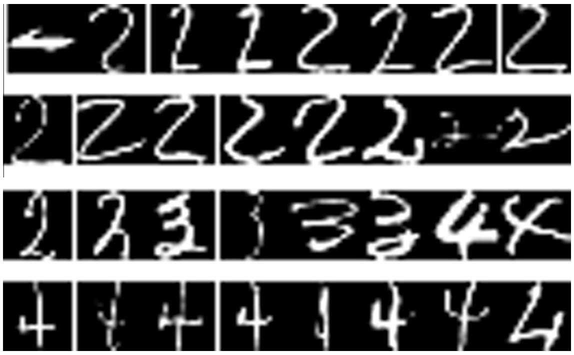


Fig. 8. Digits hard to be classified.

vector quantization), LapRLSC and MCC. Here we omit PGNG, since its classification performance is inferior to that of RBFNN [21].

Columbia Object Image Library (COIL-20) [37] is a database of gray-scale images of 20 objects which is given in Fig. 9. The images of objects were taken at pose intervals of 5°, i.e., 72 poses per object, so the total number of images is 1440. In this experiment, the training set consists of 36 images (one for every 10°) for each object, and the test set consists of the remaining 36 images for each object.

In this experiment, we not only record the recognition rates on 20 objects but also record the performance on the first 2, 4, 6, and 8 objects, respectively. From Table 4, we observe that no matter how many objects are included in the dataset, the recognition rates of MCC are the highest among five algorithms. When the number of objects is 2 or 4, the recognition accuracies of MCC are up to 100%; the accuracies of MCC slightly decrease when the number of objects increases to 6 or 8. When the number of objects reaches 20, the correct rate of MCC still remains at a high level and researches 94.31%. These results show that MCC has the good object recognition ability.

Above superior performance of MCC comes from the effective manifold clustering and subsequent manifold classification. Here we give the formed clusters obtained by MCC when the number of objects is 2. From Table 5, we can see that the cluster number optimized from the criterion function (23) is 4; the images with similar poses are assigned to a cluster and different objects are grouped to different clusters; the samples in Cluster1, 2, 3 are all from the first object and the samples in Cluster4 correspond to the second object. Such result of MCC reflects the underlying manifold distribution very correctly. Moreover, the relation matrix P computed by the formula (16)–(18) in MCC is [10; 10; 10; 01] (in Matlab format) which uncovers the real relationship between the manifolds and objects. As a result, the correct manifold clustering and effective classification mechanism make MCC better than other algorithms in object (or shape) recognition.

Similarly, we carry out the experiment on MPEG-7 dataset [38,39] to evaluate the performance of algorithms for shape recognition. This database is composed of 70 object classes with 20 shapes in each class. As shown in Fig. 10, we choose a set of shapes from MPEG-7 dataset and use leave-one-out strategy to test the recognition accuracy. From the results in Table 6, it can be seen that compared to RFRC, VQ + LVQ3, RBFNN and LapRLSC, our algorithm can achieve the highest recognition accuracy. Such result proved the superiority of MCC in object recognition again.

4.4. UCI real-life dataset

To further investigate the effectiveness of MCC on real-life datasets, we use 16 datasets cited from the University of California-Irvine (UCI) machine learning repository [40]. It is a repository of databases, domain theories and data generators collected by the machine learning community for the empirical analysis of machine learning algorithms.

We compare the experimental results among RFRC, RBFNN, VQ + LVQ3, LapRLSC and MCC. In MCC, the cluster number K is sought in the range from the number of classes up to c_{max}. Here the parameter c_{max} is set to $\sqrt{N}$ in terms of Bezdek's suggestion [41] where N is the number of the training samples. The other clustering parameters σ and α used here can be optimized from the criterion function (23) (see Section 3.2.1). In RFRC and RBFNN, the RBF kernel is adopted and its scale factor is determined by searching in $\{0.01, 0.05, 0.1, 0.5, 1, 5, 10, 15\}$. In LapRLSC, the regularization parameters are determined from $\{2^{-1}, 2^{0}, 2^{3}, 2^{5}, 2^{7}, 2^{9}\}$. Due to the multiple parameters existing in RFRC, VQ + LVQ3, RBFNN and LapRLSC, the discrete grid search [42] which is a minimization procedure based on exhaustive search in a limited range is adopted to acquire the optimal values with trial-and-error approach [43] on the training dataset. It is worth emphasizing that in our algorithm, there is only one parameter K which is needed to be selected in our algorithms, and while in other algorithms, there are two or three parameters need to be determined by the discrete grid search. Hence, our algorithm takes less time to estimate the parameter values.

Table 4
Comparison of object recognition rates on Coil dataset.

	RFRC (%)	VQ + LVQ3 (%)	RBFNN (%)	LapRLSC (%)	MSCC (%)
Coil-2 objects	79.17	100	100	100	100
Coil-4 objects	79.86	96.36	97.92	95.78	100
Coil-6 objects	34.26	93.11	94.04	93.67	98.30
Coil-8 objects	31.94	92.44	93.67	92.59	96.53
Coil-20 objects	25.417	91.94	91.81	91.42	94.31

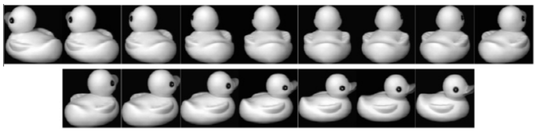
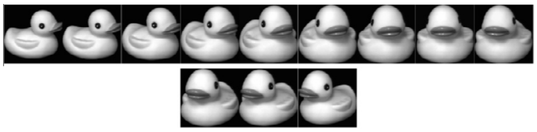
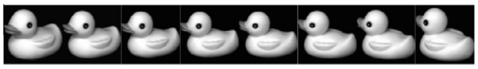
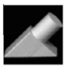
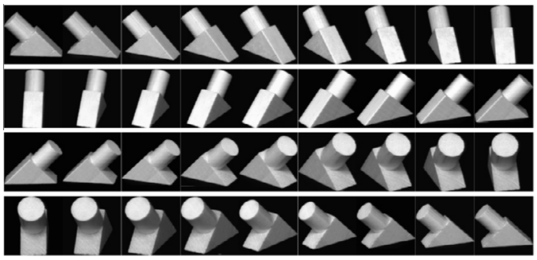
In all of our experiments, each dataset is randomly partitioned into two halves: one half is used for training and the other for testing. This process runs repeatedly and independently 10 times, and their averaged accuracy and the corresponding standard deviation are reported in Table 7.

First, we compare the classification accuracies resulted by RFRC, VQ + LVQ3 and MCC. It can be seen from Table 7 that the accuracies of MCC are respectively better than those of RFRC except for the dataset *Soybean_small*; meanwhile, MCC works better than VQ + LVQ3 except for the dataset *WDBC*, *Lenses*, *Heart_disease* and *Pima_Indian_diabetes*. Such a performance promotion of MCC attributes to the adoption of the manifold clustering and statistical relation matrices **P**, which makes MCC more appropriate for the datasets existing in real-life.

Second, we make the comparison between RBFNN and MCC. MCC obtains comparable or better performance on 7 datasets and worse performance on 9 datasets than RBFNN. It is worth pointing out that the comparison here is in fact not quite favorable for our algorithm MCC. It is because that the connection weights in RBFNN are the optimized results based on the MSE criterion, in contrast the relation matrix **P** in MCC are directly from the statistical-based construction rather than optimization. Therefore, the relatively inferior classification performance yielded by MCC is comprehensible. Our next work is to optimize the relation matrix **P** to further promote its performance.

Fig. 9. Objects in Coil-20 dataset.

Table 5
Clusters obtained by MCC when the number of objects is 2.

	Cluster centers	Formed clusters
Cluster1		
Cluster2		
Cluster3		
Cluster4		

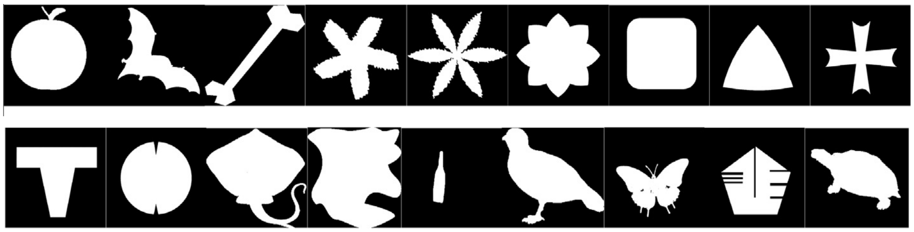


Fig. 10. A set of samples from MPEG-7 dataset.

Table 6
Object recognition rates on MPEG7 dataset.

	RFRC (%)	VQ + LVQ3 (%)	RBFNN (%)	LapRLSC (%)	MSCC (%)
MPEG-7	38.89	85.80	90.09	91.40	92.65

Finally, to give a reference, we make a comparison against the state-of-the-art classifier LapRLSC. MCC gains better accuracies than LapRLSC on 7 datasets, comparable accuracies on 5 datasets and worse accuracies only on 4 datasets, indicating that MCC is highly competitive in terms of classification accuracy. However, MCC possesses the following advantages compared to LapRLSC: (1) our algorithm can perform manifold clustering learning which can auto-determine the clustering parameters without manual determining, while LapRLSC cannot represent the manifold structures in a explicit way; (2) from the obtained relation matrix **P**, MCC can gain some insight into the structure of given data and the relation between the structure and their classes, while LapRLSC

Table 7
Comparison of classification accuracies on benchmark datasets.

Dataset (#samples × #dim × #class)	RFRC	VQ + LVQ3	RBFNN	LapRLSC	MCC
Water (116 × 38 × 2)	*97.9 ± 1.3	**98.4 ± 1.2**	**98.3 ± 1.0**	**98.4 ± 1.2**	**98.4 ± 1.5**
WBCD (683 × 9 × 2)	**97.0 ± 0.6**	*96.8 ± 0.6	*96.8 ± 0.5	*96.6 ± 0.9	**97.2 ± 0.4**
Lung_cancer (32 × 56 × 3)	40.6 ± 11.3	42.5 ± 10.8	*43.8 ± 15.8	40.1 ± 10.5	**54.4 ± 10.1**
Sonar (208 × 60 × 2)	*77.5 ± 3.9	73.9 ± 2.8	**80.2 ± 3.0**	66.2 ± 3.9	**80.1 ± 4.1**
Ionosphere (351 × 34 × 2)	**90.9 ± 1.7**	85.4 ± 1.8	88.4 ± 1.6	*89.2 ± 1.6	**90.6 ± 2.1**
Pid (768 × 8 × 2)	69.6 ± 2.8	72.1 ± 2.0	**74.6 ± 2.5**	*72.6 ± 1.6	**74.1 ± 0.3**
Iris (150 × 4 × 3)	*95.3 ± 1.1	94.7 ± 1.9	**96.4 ± 1.6**	*95.6 ± 2.1	*95.3 ± 2.0
Ecoli (336 × 7 × 8)	81.8 ± 3.3	78.8 ± 3.0	**85.2 ± 2.7**	81.9 ± 2.6	*83.5 ± 2.0
Thyroid (215 × 5 × 3)	91.8 ± 2.0	92.7 ± 2.2	**95.3 ± 1.0**	*93.8 ± 1.5	*93.7 ± 2.2
WDBC (569 × 30 × 2)	92.0 ± 1.6	**96.4 ± 0.9**	95.0 ± 1.2	94.8 ± 2.3	95.0 ± 1.2
Wine (178 × 13 × 3)	96.0 ± 1.7	96.5 ± 1.5	*97.3 ± 1.1	**98.3 ± 1.1**	96.0 ± 1.2
Lenses (24 × 4 × 3)	71.7 ± 7.6	74.2 ± 11.5	**75.8 ± 14.6**	**78.3 ± 10.0**	70.5 ± 10.7
Balance_scale (625 × 4 × 3)	84.7 ± 1.5	86.0 ± 1.8	*90.5 ± 1.0	**92.7 ± 1.3**	86.0 ± 1.5
Heart_disease (270 × 13 × 2)	80.9 ± 2.2	*81.4 ± 1.8	**82.5 ± 2.3**	79.1 ± 1.8	80.1 ± 1.5
Pima_Indian_diabetes(768 × 8 × 2)	70.7 ± 3.2	*72.6 ± 2.0	**74.2 ± 2.3**	71.6 ± 1.6	71.8 ± 1.7
Soybean_small (47 × 35 × 4)	**99.1 ± 1.7**	96.1 ± 10.4	98.1 ± 1.7	*98.7 ± 2.0	97.4 ± 4.4

The best result for each dataset is given in bold, and the second-best result for each dataset is marked with an asterisk.

does not care about the relationship between such manifold and the classes; (3) the class posterior probabilities computed in this framework can reflect the confidence of the classification decision, which is important for reliable and interpretable classification, while in LapRLSC, the classification result is lack of explanation.

Since the real-world data are complicated and may be from multifarious distributions, their underlying structure distribution is hard to know in advance. Under this situation, different algorithms make different assumptions about the data. For example, RFRC and RBFNN suppose that data arise from a mixture of Gaussian, while LapRLSC and MCC assume the data shaped in the form of manifold structure. When the assumption about the data distribution meet the data distribution, the algorithm is easy to obtain better results; otherwise the result may be poor. Our experiments demonstrate this conclusion again. It can be observed that no algorithm is absolutely dominant on 16 tested UCI datasets. This is well consistent with the no free lunch theorem which states that, there are no context-independent or usage-independent reasons to favor one classification method over another, unless appropriate prior information is incorporated in model selection.

5. Conclusion and further study

In this paper, a learning framework for both manifold clustering and classification (MCC) is presented. MCC is implemented in a two-step *sequential* manner. At first, the clustering through ranking on manifolds is first performed to discover the data distribution. Then, the class posterior probability is calculated based on the relationship between manifolds and classes. In MCC, this relationship plays an important role. Its function is to create a bridge between clustering learning and classification learning and its elements are calculated according to the Bayesian theory. The results reported in this paper show that our proposed algorithm is effectiveness in manifold clustering, manifold classification and relationship reveal. To conclude, this framework can achieve three goals at one time: (1) exploiting the effective manifold clustering hidden in data; (2) designing an effective and transparent classification mechanism; (3) reflecting the statistical relationship between clusters and classes.

For this learning framework MCC, there are still a number of open research issues. First, for different clusters, a different set of parameters can by optimized for different scales, which leaving much room for improvement for clustering performance. Second, the closeness between points in our framework is not measured by manifold distance, but by similarity ranking on manifold, and hence theoretically guaranteeing the effectiveness of the subsequent learning. Finally, the framework of MCC can easily be extended by using different clustering methods. If a clustering technique can form the partition for multi-manifold, it can also be adopted instead of clustering through Ranking on manifolds and thus a novel manifold learning algorithm can be derived.

Acknowledgements

We would like to thank the anonymous referees for their constructive comments and suggestions to greatly improve the presentation of this paper. This work was supported by the National Science Foundation of China under Grant Nos. 61432008, 61300165 and 61272222.

References

[1] D.T. Larose, Discovering Knowledge in Data: An Introduction to Data Mining, John Wiley & Sons, 2014.

[2] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, G. Zhang, Transfer learning using computational intelligence: a survey, Knowl.-Based Syst. 80 (2015) 14–23.

[3] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender system application developments: a survey, Decis. Support Syst. 74 (2015) 12–32.

[4] C.H. Lee, A gradient approach for value weighted classification learning in naive Bayes, Knowl.-Based Syst. 85 (2015) 71–79.

[5] D. Gómez, J. Yáñez, C. Guada, et al., Fuzzy image segmentation based upon hierarchical clustering, Knowl.-Based Syst. 87 (2015) 26–37.

[6] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recogn. Lett. 31 (8) (2010) 651–666.

[7] J.C. Bezdek, R. Ehrlich, W. Full, FCM: the fuzzy C-means clustering algorithm, Comput. Geosci. 10 (2) (1984) 191–203.

[8] Z. Zivkovic, Improved adaptive Gaussian mixture model for background subtraction, Proceedings of the 17th International Conference on Pattern Recognition, 2004, ICPR 2004, vol. 2, IEEE, 2004, pp. 28–31.

[9] A.J. Maren, C.T. Harston, R.M. Pap, Handbook of Neural Computing Applications, Academic Press, 2014.

[10] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1999.

[11] C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. (TIST) 2 (3) (2011) 27.

[12] M. Setnes, R. Babuška, Fuzzy relational classifier trained by fuzzy clustering, IEEE Trans. Syst. Man Cybern. Part B 29 (1999) 619–625.

[13] W. Cai, S. Chen, D. Zhang, Robust fuzzy relational classifier incorporating the soft class labels, Pattern Recogn. Lett. 28 (16) (2007) 2250–2263.

[14] D.Q. Zhang, S.C. Chen, A novel kernelized fuzzy C-means algorithm with application in medical image segmentation, Artif. Intell. Med. 32 (1) (2004) 37–50.

[15] S. Chen, D. Zhang, Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure, IEEE Trans. Syst. Man Cybernet. Part B: Cybernet. 34 (4) (2004) 1907–1916.

[16] H. Yu, T. Xie, S. Paszczynski, et al., Advantages of radial basis function networks for dynamic system design, IEEE Trans. Ind. Electron. 58 (12) (2011) 5438–5450.

[17] H.G. Han, J.F. Qiao, Adaptive computation algorithm for RBF neural network, IEEE Trans. Neural Networks Learn. Syst. 23 (2) (2012) 342–347.

[18] U. Von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.

[19] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, Adv. Neural Inform. Process. Syst. 2 (2002) 849–856.

[20] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, John Wiley & Sons, 2012.

[21] X. Yang, S. Chen, B. Chen, Plane-Gaussian artificial neural network, Neural Comput. Appl. 21 (2) (2012) 305–317.

[22] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.

[23] M. Belkin, P. Niyogi, V. Sindhwani, On manifold regularization, in: Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT 2005), 2005, pp. 17–24.

[24] I.W. Tsang, J.T. Kwok, Large-scale sparsified manifold regularization, Adv. Neural Inform. Process. Syst. (2006) 1401–1408.

[25] M.R. Daliri, V. Torre, Shape and texture clustering: best estimate for the clusters number, Image Vis. Comput. 27 (10) (2009) 1603–1614.

[26] L. Bai, J. Liang, C. Dang, An initialization method to simultaneously find initial cluster centers and the number of clusters for clustering categorical data, Knowl.-Based Syst. 24 (6) (2011) 785–795.

[27] A.D. Torshizi, M.H.F. Zarandi, A new cluster validity measure based on general type-2 fuzzy sets: application in gene expression data clustering, Knowl.-Based Syst. 64 (2014) 81–93.

[28] J. Sim, S.Y. Kim, J. Lee, Prediction of protein solvent accessibility using fuzzy *k*-nearest neighbor method, Bioinformatics 21 (12) (2005) 2844–2849.

[29] P.S. Bradley, O.L. Mangasarian, *k*-Plane clustering, J. Global Optim. 16 (1) (2000) 23–32.

[30] T. Poggio, C.R. Shelton, On the mathematical foundations of learning, Am. Math. Soc. 39 (1) (2002) 1–49.

[31] M. Breitenbach, G.Z. Grudic, Clustering through ranking on manifolds, in: Proceedings of the 22nd International Conference on Machine Learning, ACM, 2005, pp. 73–80.

[32] D. Zhou, O. Bousquet, T.N. Lal, et al., Learning with local and global consistency, Adv. Neural Inform. Process. Syst. 16 (16) (2004) 321–328.

[33] J.M. Bernardo, A.F.M. Smith, Bayesian Theory, John Wiley & Sons, 2009.

[34] F. Lauer, C.Y. Suen, G. Bloch, A trainable feature extractor for handwritten digit recognition, Pattern Recogn. 40 (6) (2007) 1816–1824.

[35] D. Nova, P.A. Estévez, A review of learning vector quantization classifiers, Neural Comput. Appl. 25 (3–4) (2014) 511–524.

[36] M. Biehl, A. Ghosh, B. Hammer, Dynamics and generalization ability of LVQ algorithms, J. Mach. Learn. Res. 8 (2007) 323–360.

[37] J. Gou, Y. Zhan, Y. Rao, et al., Improved pseudo nearest neighbor classification, Knowl.-Based Syst. 70 (2014) 361–375.

[38] R. Daliri M, V. Torre, Classification of silhouettes using contour fragments, Comput. Vis. Image Understand. 113 (9) (2009) 1017–1025.

[39] R. Daliri M, V. Torre, Shape recognition based on kernel-edit distance, Comput. Vis. Image Understand. 114 (10) (2010) 1097–1103.

[40] C. Blake, E. Keogh, C.J. Merz, UCI repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html], Department of Information and Computer Science, University of California, Irvine, CA, 1998.

[41] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, et al., An extensive comparative study of cluster validity indices, Pattern Recogn. 46 (1) (2013) 243–256.

[42] B. Jiménez Á, J.L. Lázaro, J.R. Dorronsoro, Finding optimal model parameters by discrete grid search, in: Innovations in Hybrid Intelligent Systems, Springer, Berlin, Heidelberg, 2007, pp. 120–127.

[43] S. Abe, Training of support vector machines with Mahalanobis kernels, in: Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005, Springer, Berlin, Heidelberg, 2005, pp. 571–576.