Manifold Contraction for Semi-Supervised Classification

HU EnLiang[1,2†], CHEN SongCan[1†] & YIN XueSong[1]

[1] School of Computer Science & Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China

[2] School of Mathematics, Yunnan Normal University, Kunming 650092, China

The generalization ability of classification is often closely related to both the intra-class compactness and the inter-class separability. Owing to the fact that many current dimensionality reduction methods, regarded as a pre-processor, often lead to the poor classification performance on real-life data, in this paper, a new data pre-processing technique called manifold contraction (MC) is proposed for the classification-oriented learning task. The main motivation behind MC lies in seeking a proper mapping of contracting the given multiple-manifold data such that the ratio of the intra-class to the inter-class scatters is minimized. Moreover, in order to properly control the contraction level in MC, an adaptive MC (AMC) criterion is developed in the semi-supervised setting. Due to its generality, MC can be not only applied in original space and reproducing kernel Hilbert space (RKHS), but also easily incorporated with dimensionality reduction method for further improvement of classification performance. The final experimental results show that MC, as a data preprocessor, is effective and promising in the subsequent classification learning, especially in small-size labeled sample case.

Manifold learning, dimensionality reduction, lower-dimensional embedding, semi-supervised learning, classification, manifold contraction, adaptive manifold contraction

1 Introduction

It is well-known that classification is a fundamental task in pattern recognition and machine learning communities, in which the control for classification complexity plays an important role especially in the small-size labeled sample case. According to the viewpoints in refs. [9][10], the classification complexity can be generally explored from two different aspects: classifier complexity and data complexity. Usually, the classifier complexity can be controlled by model regularization, parameter selection, etc., while the data complexity may be reduced through data pre-processing techniques such as dimensionality reduction, feature selection, data smoothing, etc.. Bishop demonstrated that the data pre-processing is one of the most significant factors in determining the performance of final system [6], and Duin also pointed out that the classification complexity mainly inherits data complexity [10]. These assertions largely indicate that both complexities of model and data are not independent of each other, and the final performance originates not only from the design of a good classifier, but also from the generation of a good pattern representation by reduction of data complexity.

Since manifold data is ubiquitous in the real world, how to yield good pattern representation for manifold data is a significant issue. Currently, a popular scheme of reducing manifold data is to embed them into a lower-dimensional space by dimension reduction technique (DR), also named lower-dimensional embedding under graph frame-

Received November " " ; accepted " "
doi:
†Corresponding author (email: helnuaa@.nuaa.edu.cn, s.chen@nuaa.edu.cn)
Supported by National Science Foundation of China Grant (Grant No. 60773061)

Citation: HU EnLiang, CHEN SongCan, YIN XueSong. Manifold Contraction for Semi-Supervised Classification. Sci China Ser F, " ", " ": , doi: " "

work [30]. As we have known, the representative DR methods include principal component analysis (PCA) [14], linear discriminant analysis (LDA) [11], semi-supervised discriminant analysis (SDA) [8], semi-supervised dimensionality reduction (SSDR) [31], kernel PCA (KPCA) [20], local linear embedding (LLE) [19], isometric mapping (ISOMAP) [22], diffusion map (DiffuMap) [16], Laplacian eigen-mapping (LE) [1], etc.. However, many DR methods, as a pattern generator, cannot necessarily lead to an improvement in classification performance. For instance, after analyzing many experiments on benchmarks, it has been found that many DRs like ISOMAP, LE and LLE often led to poor classification accuracy on real-world datasets [3]. Other poor classification accuracies of DRs have also been reported in ref. [25]. These reports partially demonstrate that many DRs do not always suit as a good pre-processor for classification learning.

By contrast, we find that the classification accuracies on some embedding data are even lower than those on original data [3][25]. This discovery inspires us to seek a new pattern representation in original space directly. Hence, in this paper, we propose a new data pre-processing technique named manifold contraction (MC), which tries to capture a good pattern representation directly in the original space. An intuition behind MC is that, after contracting manifold data in its distribution direction, the intra-class points become more compact and the inter-class points become further away. Thus, the so-generated pattern representation will be favorable for the subsequent classification learning.

The proposed MC aims to act as a classification-oriented pattern pre-processor. As is also well-known, the semi-supervised classification learning has become a popular topic recently [36]. It has empirically shown that many existing supervised classifiers cannot work well if their learning only depends on labeled samples. Therefore, many semi-supervised classification learning schemes are consecutively developed [36]. In particular, based on manifold hypothesis, some typical semi-supervised classification methods such as Laplacian SVM (LapSVM) [2], Markov random walk [21], manifold ranking [13][32][33] and label propagation (LP) [37][5] have been proposed. Usually, the scarcity of labeled training samples leads to a large solution space, implying an over-fitting risk or poor generalization ability. Meanwhile, controlling classifier's complexity is one important way to avoid the over-fitting and improve the generalization ability, e.g., in LapSVM [2], utilizing the manifold distribution of both labeled and unlabeled samples to control the classifier's complexity and developing a manifold regularization technique. Alternatively, following the Bishop's and Duin's viewpoints, another possible pathway of improving classification performance is to reduce the data complexity. That is, we can likewise improve the performance by managing to obtain a sufficiently good pattern representation closely associated to the given semi-supervised classification task. In this way, it is still possible to make an existing supervised-classifier work well as a semi-supervised classifier. Based on such a consideration, our motivation is that a semi-supervised classification learning can also be effectively fulfilled by a supervised-classifier as long as both of the labeled and the unlabeled points of the same class are contracted into a tighter space by MC.

Besides, our MC technique can be not only easily applied in original space and RKHS, but also conveniently incorporated with the technique of dimension reduction. To the best of our knowledge, there have not been any reports on MC technique as a classification-oriented pre-processor. The rest of paper is organized as follows: In Section 2, we introduce MC in original space and RKHS above all, then combine it with dimension reduction, and an adaptive MC (AMC) criterion is developed to control the level of MC in semi-supervised setting. In Section 3, the classification generalization ability of MC is discussed. In Section 4, many experimental results both on artificial and real-world datasets are provided to evaluate MC. Discussions are presented in Section 5 and conclusions and future works are offered in the last Section.

2 Manifold contraction

In many real-world problems, data often include multi-class manifold data. Focusing on the subsequent classification learning on such manifold data, in this paper, we propose MC as a pre-processor to achieve a better pattern representation in original space directly. The main idea of MC is to contract manifold data in its distribution direction so that both the intra-class compactness and the inter-class separability are enhanced. In order to help understand our MC, the following Figs. 1∼2 show a comparative overview between MC and some DR methods.

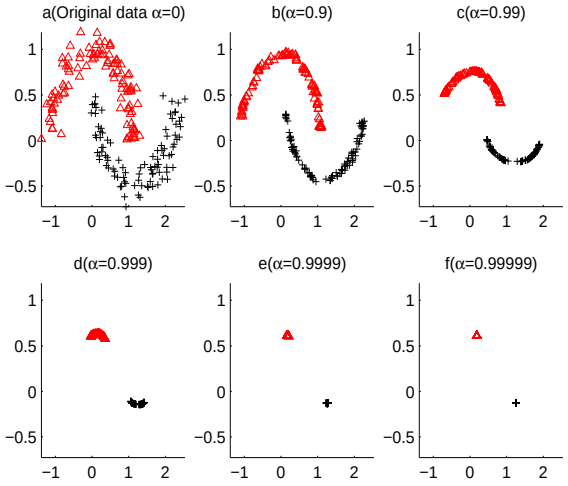

Fig. 1 The shapes of 2moons contracted by MC with different levels.

Concretely, Fig. 1 (a) consists of an original manifold data named 2moons and Figs. 1 (b, c, d, e, f) show the corresponding contracted shapes of 2moons by MC with different-levels (parameterized by α, interpreted later). Figs. 2 (a, b, c, d) consist of the one dimension (1D) embeddings of the original data 2moons respectively performed by PCA, SDA, DiffuMap and ISOMAP. Clearly, we observe that the 1D-embeddings (except ISOMAP) partially overlap each other despite all of them make the intra-manifold points more compact. Comparatively, our MC makes not only either moon more compact, but also the margin between the two moons larger. These intuitively demonstrate that MC is likely superior to some DR methods as a classification-oriented pre-processor for the multi-class manifold data here.

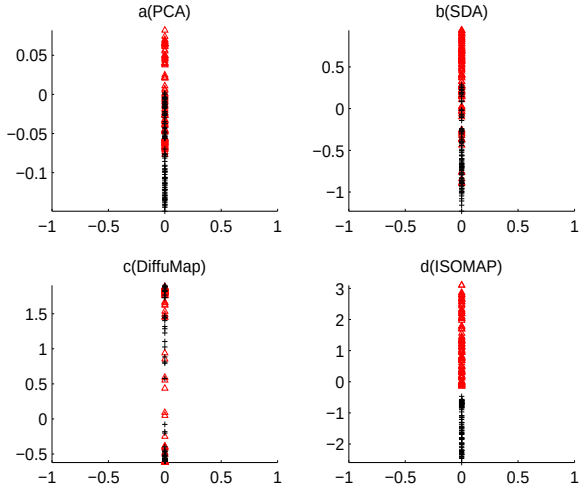

Fig. 2 The different 1D embeddings of the original data 2moons performed by PCA, SDA, DiffuMap and ISOMAP respectively.

Next, we will detail MC in original space firstly.

2.1 Manifold contraction in original space

Let $\mathcal{I} = \{1, 2, \cdots, n\}$ be the indices of all given samples and N_i be the index set of x_i's neighbors, defined as follows:

$$N_i = \{j \mid x_j \text{ is a neighbor of } x_i,\ j \in \mathcal{I} \text{ and } j \neq i\}$$

Let $p_{ij} \in [0, 1]$ be the weight reflecting how close x_j is to x_i. Then $m_i = \sum_{j \in N_i} p_{ij} x_j$ is a weighted mean in N_i with $\sum_{j \in N_i} p_{ij} = 1$.

In order to contract a given manifold, we force each x_i to move towards $m_i(t)$ in the direction of $m_i(t) - x_i$, where $m_i(t) = \sum_{j \in N_i} p_{ij} x_j(t)$ is m_i's state at time t. Now set the moving step at $\alpha (0 \leqslant \alpha \leqslant 1)$. For each x_i, such a moving leads its state at time t+1 to be in the form

$$x_i(t+1) = x_i + \alpha \left(m_i(t) - x_i \right).$$

The above iteration equation implies that each new point $x_i(t+1)$ originating from x_i will move in the direction of $m_i(t) - x_i$ in each iteration from t to t +1. Further, we have

$$x_i(t+1) = (1-\alpha) x_i + \alpha \sum\nolimits_{j \in N_i} p_{ij} x_j(t) \qquad (1)$$

Let $X = (x_1, x_2, \cdots, x_n)^T$ and $X(t) = (x_1(t), x_2(t), \cdots, x_n(t))^T$, i.e., the i-th row consisting of the i-th sample for each $i \in \mathcal{I}$. Then we can

reformulate eq. (1) as follows:

$$\begin{cases} X(t+1) = (1-\alpha)X + \alpha PX(t) \\ X(0) = X, \quad t = 0, 1, \cdots . \end{cases} \tag{2}$$

If we view P as a probability transition matrix, then eq. (2) is similar to the random walk in ref. [21] or the manifold ranking in ref. [33]. A key difference lies in the fact that both the random walk and the manifold ranking propagate class labels by P, but eq. (2) is to propagate samples by such a P. With some simple calculations to eq. (2), we have

$$\begin{cases} X(t+1) = [(\alpha P)^{t+1} + (1-\alpha)\sum_{i=0}^{t}(\alpha P)^i]X \\ X(0) = X, \quad t = 0, 1, \cdots . \end{cases} \tag{3}$$

The following Theorem 1 gives a convergence proof of the iterative eq. (3).

Theorem 1: If $0 \leqslant \alpha < 1$, then the iterative eq. (3) is convergent.

Proof: Since p_{ij} is subject to $0 \leqslant p_{ij} \leqslant 1$ and $\sum_{j\in N_i} p_{ij} = 1$, according to Gerschgorin disk Theorem, $\rho(P) \leqslant 1$ (where $\rho(P)$ is the spectral radius of P). Further from $0 \leqslant \alpha < 1$, we have $\rho(\alpha P) = \alpha(P) < 1$. This means that eq. (3) is convergent and the proof is completed.

From the convergence of Theorem 1, we have $\lim_{t\to\infty}(\alpha P)^{t+1} = 0$, $\lim_{t\to\infty}\sum_{i=0}^{t}(\alpha P)^i = (I-\alpha P)^{-1}$; thus

$$\lim_{t\to\infty} X(t+1) = (1-\alpha)(I-\alpha P)^{-1}X \tag{4}$$

Letting $S = (1-\alpha)(I-\alpha P)^{-1}$ and $X^* = \lim_{t\to\infty} X(t+1)$, we can rewrite eq. (4) as

$$X^* = SX \tag{5}$$

As a consequence, eq. (5) induces a contraction mapping $T : X \to X^*$, which is parameterized by α (hidden in S). We denote such a mapping by $X^* = T(X, \alpha)$ or $X^* = T(X)$ when α is just a constant. Furthermore, we name T MC and dub X^* contracted set or root shape of X. We also name S shrinkage matrix and α shrinkage parameter. Now, please review Figs. 1(b, c, d, e, f), which actually include the corresponding root shapes generated by different αs in parenthesis.

From eq. (1), we have $\lim_{t\to\infty} x_i(t+1) = (1-\alpha)x_i + \alpha\lim_{t\to\infty}\sum_{j\in N_i} p_{ij}x_j(t)$. Let $\lim_{t\to\infty} x_i(t+1) = x_i^*$. Then $x_i^* = x_i$ for $\alpha = 0$ and $x_i^* = \lim_{t\to\infty} m_i(t) = \sum_{j\in N_i} p_{ij}x_j^*$ for $\alpha = 1$. For an individual x_i, we can describe T as $T : x_i \to x_i^*$ or $x_i^* = T(x_i, \alpha)$, where x_i^* is the contractive image of x_i.

Let $cov(X)$ be the convex closure of X, i.e., $cov(X) = \{x \mid x = \sum_{i=1}^{n} a_i x_i,\ x_i \in X\}$ with $0 \leqslant a_i \leqslant 1$ and $\sum_{i=1}^{n} a_i = 1$. Then we have the following Theorem 2.

Theorem 2: $cov(X^*) \subseteq cov(X)$

Proof: According to $0 \leqslant p_{ij} \leqslant 1$, $\sum_{j\in N_i} p_{ij} = 1$ and $m_i(t) = \sum_{j\in N_i} p_{ij}x_j(t)$, we known $m_i(0) \in cov(X)$. If $m_i(t) \subseteq cov(X)$, from $x_i(t+1) = (1-\alpha)x_i + \alpha m_i(t)$ and $0 \leqslant \alpha \leqslant 1$, we have $x_i(t+1) \subseteq cov(X)$ and thus $m_i(t+1) \subseteq cov(X)$. By mathematical induction, we have $\lim_{t\to\infty} m_i(t+1) \subseteq cov(X)$. This implies $x_i^* \subseteq cov(X)$ for $\forall i \in \mathcal{I}$ by eq. (1). Hence, $X^* \subseteq cov(X)$ and so $cov(X^*) \subseteq cov(X)$. The proof is completed.

Theorem 2 confirms the contractive property of the mapping T induced by eq. (5). To sum up, our MC can map the original manifold data into its root shape by a shrinkage matrix S, and this process is directly performed in original space. The main steps of implementing MC is outlined below:

Algorithm of MC in original space

Input:

X — original manifold data;
N_i — the indices of k-nearest neighbors for x_i;
α — shrinkage parameter;
σ — band-width parameter.

Output:

X^* — root shape of X.

1. Form a weighted adjacent matrix $W = (w_{ij})$ where $w_{ij} = \exp\{-\|x_i - x_j\|^2 \big/ 2\delta^2\}$ for $j \in N_i$ and otherwise $w_{ij} = 0$;
2. Get $P = D^{-1}W$ as a normalization of W, where $D = diag(d_{ii})$ with $d_{ii} = \sum_j w_{ij}$;
3. Return $X^* = T(X, \alpha) = SX$ after constructing the shrinkage matrix $S = (1-\alpha)(I-\alpha P)^{-1}$.

According to eqs. (4) and (5), we have $T(X, 0) = X$ and $T(X, 1) = 0$ (zero matrix), meaning that X is un-contracted for $\alpha = 0$ and will collapse to 0 for $\alpha = 1$. Therefore, a naturally raised problem is how to adaptively determine α, by which the root

shape can be controlled properly. In order to resolve this problem, in subsection 2.4., we define an adaptive MC (AMC) criterion for semi-supervised classification setting, aiming to seek a locally optimal α.

In the next subsection 2.2, we will extend MC to RKHS.

2.2 Manifold contraction in RKHS

Recently, kernel trick has become a powerful technique for studying nonlinear data, and it implies that a mapping $\phi(\cdot)$ can be induced from a kernel function, i.e., $\phi : x \to \phi(x)$.

Let $\phi(X) = [\phi(x_1), \phi(x_2), \cdots, \phi(x_n)]^T$. Then we can describe $\phi(X)$'s root shape as $\phi^*(X) = S\phi(X)$ by eq. (5). Thus, two gram matrices for $\phi(X)$ and $\phi^*(X)$ can be respectively formulated into $K(X,X) = \phi(X)\phi(X)^T$ and $K^*(X,X) = \phi^*(X)\phi^*(X)^T$. Moreover, we have

$$K^*(X,X) = S\phi(X)\phi(X)^T S^T = SK(X,X)S^T$$

Here, we call $K^*(X,X)$ a shrinkage kernel of $K(X,X)$. Especially, the shrinkage kernel of the linear kernel XX^T is SXX^TS^T.

In the next subsection 2.3, we will combine MC with the dimension reduction method.

2.3 Dimension reduction+manifold contraction

Our MC can be naturally and conveniently incorporated into another preprocessing techniques such as dimensionality reduction (DR). In other words, when a DR process is needed too, we can combine MC with DR. Let each v_i be a projective vector. Then $V = (v_1, v_2, \cdots, v_d)$ becomes a projective matrix severed as DR. Based on both V and S (shrinkage matrix), we can obtain a preprocessed representation X^* as follows:

$$X^* = SXV$$

It is interesting that SXV can be regarded as two-side preprocessing for X, i.e., left MC and right DR. Further, "DR+MC" can be interpreted as either first MC then DR or first DR then MC. In the experimental part, we will examine such "DR+MC" scheme including ISOMAP+MC and SDA+MC.

2.4 Adaptive MC in semi-supervised setting

We will define an optimizing criterion to adaptively get a proper α. In order to show how MC depends on α, in Fig. 3, we give an illustration of 2moons' root shapes under some αs. In Fig. 3(a), the close points are firstly connected in a 5-nearest neighbor graph. To avoid "isolated" component as in ref. [13], we link all points in a global connected graph after adding the partial shortest edges generated by the minimum spanning tree algorithm.

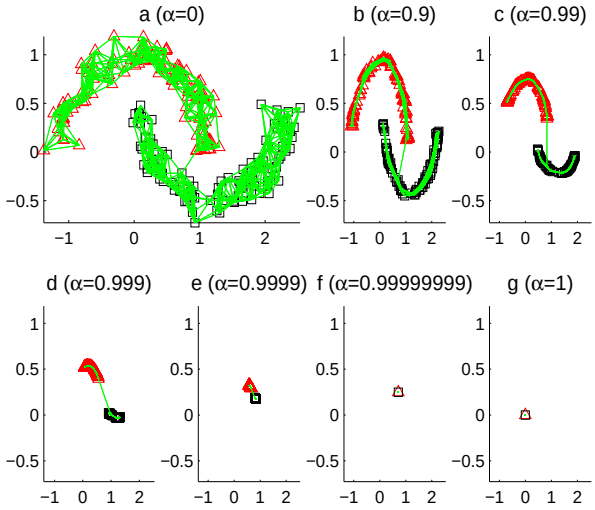

Fig. 3 The corresponding root shapes of 2moons contracted with different αs.

In Fig. 3, we observe that a proper α is desired. The reason is that a too small α means a very weak contraction, which will lead to a larger ratio of intra-class to inter-class scatters. At the same time, a too big α is more likely to make the inter-class points overlap each other as shown in Fig. 3(f). This is because the bad "bridge points" connect different classes, which will weaken the inter-class separability [28].

When c classes' manifold data $X_1, X_2, \cdots, X_c$ are given, we can connect all $X_i (i = 1, 2, \cdots, c)$ in a global connected graph. Thus after constructing a S, we can obtain $X^* = SX = T(X, \alpha)$, where $X = (X_1, X_2, \cdots, X_c)^T$ and $X^* = (X_1^*, X_2^*, \cdots, X_c^*)^T$. In order to obtain a desired α, we can define a criterion to minimize the ratio of intra-class to inter-class scatters in semi-supervised setting.

Concretely, let $x_i^* = T(x_i, \alpha)$ and $\|\cdot\|_H$ be the inner-product norm of the RKHS with respect to

K^*. Then we have

$$\left\|x_i^* - x_j^*\right\|_H^2 = K^*(x_i, x_i) + K^*(x_j, x_j) - 2K^*(x_i, x_j)$$

Let $e_i = (\underset{1}{0}, \cdots, \underset{i-1}{0}, \underset{i}{1}, \underset{i+1}{0}, \cdots, \underset{n}{0})$ a row selection vector and $e_{ij} = e_i - e_j$, we have

$$K^*(x_i, x_j) = e_i K^*(X, X) e_j^T = e_i S K(X, X) S^T e_j^T$$
$$\left\|x_i^* - x_j^*\right\|_H^2 = e_{ij} S K(X, X) S^T e_{ij}^T$$

Thus, we can define an intra-class scatter S_w and an inter-class scatter S_b as

$$\begin{aligned} S_w &= \tfrac{1}{l^2} \textstyle\sum_{i,j=1}^{l} L_{ij} e_{ij} S K(X, X) S^T e_{ij}^T \\ S_b &= \tfrac{1}{l^2} \textstyle\sum_{i,j=1}^{l} (1 - L_{ij}) e_{ij} S K(X, X) S^T e_{ij}^T \end{aligned}$$

where $\mathcal{I} = \{1, \cdots, l, l+1, \cdots, l+u\}$, $l + u = n$. l and u are respectively equal to the numbers of labeled and unlabeled samples. Meanwhile, L_{ij} is defined below:

$$L_{ij} = \begin{cases} 1 & \text{if label}(x_i) = \text{label}(x_j) \text{ and } 1 \leqslant \text{i, j} \leqslant l \\ 0 & otherwise \end{cases}$$

It is worthy noting that, as an alteration similar to marginal Fisher analysis in ref. [30], for each class, we can only choose k_1 longest labeled-pairs of in-class samples to define S_w, and choose k_2 nearest labeled-pairs of in-class and out-class samples to define S_b. Next, with both labeled and unlabeled samples, we can define a locally data-dependent regularizer S_r as follows:

$$S_r = \tfrac{1}{k(l+u)} \sum\nolimits_{i=1}^{l+u} \sum\nolimits_{j \in N_i} e_{ij} S K(X, X) S^T e_{ij}^T$$

Here, k is a parameter of the nearest neighbor number and N_i is the indices of k-nearest neighbors of x_i presented in subsection 2.1. Based on S_w, S_b and S_r, we define AMC as

$$\alpha^* = \arg \min_{\alpha \in [0,1)} \frac{S_w + \lambda S_r}{S_b} \tag{6}$$

In eq. (6), S_w and S_b respectively measure the intra-class and the inter-class scatters averaged on only labeled samples, while S_r reflects a local-neighborhood scatter averaged on both labeled and unlabeled samples. Meanwhile, $\lambda(> 0)$ balances the trade-off between S_w and S_r.

It is relatively easy to optimize eq. (6) in that α is just a scalar. We can get a locally-optimal α by one dimension search method and the search interval can be empirically restricted to (a, b) (where $0 \leqslant a \leqslant b < 1$).

3 Analysis of generalization ability for MC

If a proper α is obtained by optimizing AMC in eq. (6), we can obtain a good root shape X^* as a new representation of X. Nevertheless, thanks to our focus on classification performance more, a naturally arising problem is: whether we can achieve better generalization ability in X^* than in X? What it follows, we attempt to answer this problem in theory.

Giving a set of original manifold data $X = X_1 \cup X_2 \cup \cdots \cup X_c$ (where $X_i \cap X_j = \emptyset$ if $i \neq j$), we denote its corresponding root shape as $X^* = T(X) = X_1^* \cup X_2^* \cup \cdots \cup X_c^*$. Based on the definition of the convex closure in subsection 2.1, we can define the radius of X as:

$$r(X) = \tfrac{1}{2} \max_{x, x' \in \text{cov}(X)} \|x - x'\| \tag{7}$$

Meanwhile, a pair-wise margin between X_i and X_j $(i \neq j)$ can be defined as $\Delta(X_i, X_j) = \min\limits_{\substack{x \in \text{cov}(X_i) \\ x' \in \text{cov}(X_j)}} \|x - x'\|$. Thus, from such a pair-wise margin, a global margin of X can be defined as

$$\Delta(X) = \min_{X_i, X_j \in X (i \neq j)} \Delta(X_i, X_j) \tag{8}$$

Now, we have the following Theorem 3.

Theorem 3: If $X^* = T(X)$ and $X_i^* = T(X_i)$ for $i \in \{1, \cdots, c\}$, then $r(X^*) \leqslant r(X)$ and $\Delta(X^*) \geqslant \Delta(X)$.

Proof: Since $X_i^* = T(X_i)$ and $X_j^* = T(X_j)$ for $\forall X_i, X_j \in X (i \neq j)$, according to Theorem 2, we have $cov(X_i^*) \subseteq cov(X_i), cov(X_j^*) \subseteq cov(X_j)$ and $X^* \subseteq cov(X)$. Thus, we can easily get the following two inequalities:

$$\max_{x, x' \in cov(X^*)} \|x - x'\| \leqslant \max_{x, x' \in cov(X)} \|x - x'\|$$
$$\min_{\substack{x \in cov(X_i^*) \\ x' \in cov(X_j^*)}} \|x - x'\| \geqslant \min_{\substack{x \in cov(X_i) \\ x' \in cov(X_j)}} \|x - x'\|$$

Further from eqs. (7) and (8), we obtain $r(X^*) \leqslant r(X)$ and $\Delta(X^*) \geqslant \Delta(X)$. The proof is completed.

Since the hard-margin of classification hyperplane is equivalent to the nearest distance of between-class points as in eq. (8) [15], the following Theorem bridges the classification generalization ability to the ratio of r to Δ.

Theorem 4 [26]: If hyper-sphere with radius r encompasses the given data, then the VC dimensionality h of classification hyper-plane set satisfies $h \leqslant \min\left(\lceil (r/\Delta)^2 \rceil, d\right) + 1$, where d is the dimensionality of the given data.

Remark: According to Theorems 3 and 4, we can make a comparison between the VC dimensionalities of X and X^* with respect to the same hyper-plane hypothesis set. Firstly, by $r^* = r(X^*) \leqslant r(X) = r$ and $\Delta^* = \Delta(X^*) \geqslant \Delta(X) = \Delta$ from Theorem 3, we obtain $r^*/\Delta^* \leqslant r/\Delta$. Next, we have $h^* \leqslant \min\left(\lceil (r^*/\Delta^*)^2 \rceil, d\right) + 1$ and $h \leqslant \min\left(\lceil (r/\Delta)^2 \rceil, d\right) + 1$ according to Theorem 4. Although "$h^* \leqslant h$" is not strictly proved in general, it seems to follow "$h^* \leqslant h$" from "$r^*/\Delta^* \leqslant r/\Delta$" intuitively. In addition, from the conclusion that "the smaller VC dimension, the better generalization ability" in ref. [26], the classification generalization ability of X^* is likely to be better than that of X. Thus, we basically answer the previous problem positively. Below we will verify such a prediction by experiments.

4 Experiments

As a pre-processing technique, our MC can be extensively applied in classification, clustering, metric learning, etc.. In experiments, we mainly limit MC in the semi-supervised classification, in which many traditional supervised classifiers work unsatisfactorily due to the scarcity of labeled samples. Our motivation mainly lies in the fact that, when both the labeled and unlabeled points in the same class are contracted into a tighter space by MC, a semi-supervised classification can be effectively fulfilled by using only a supervised classifier trained on labeled samples. Hence, the classification procedure here consists of two steps: 1) capture the root shape as a new pattern representation by MC; 2) implement a supervised classification algorithm on such a new pattern representation, meaning that its training and predicting are performed on the labeled and unlabeled points of the obtained root shape, respectively.

4.1 Experiment setting

In MC algorithm, we use k-nearest neighbors to construct N_i for each $i \in \mathcal{I}$, and k is set at 5 as in ref. [3] throughout our experiments. The bandwidth parameter σ is specified as the average Euclid distance of all pair-wise samples as in ref. [23]. For AMC, we combine the golden section search with the parabolic interpolation method to optimize α and restrict $\alpha \in [0.9, 1)$ empirically. The trade-off factor λ in (6) is fixed at 0 for simplicity. In addition, in order to avoid "isolated" components, we always link all data points in a global connected graph whose partial shortest edges are generated by the minimum spanning tree algorithm.

Our experiment involves 5 artificial datasets and 5 real-world datasets, whose basic characteristics are described in Table 1, where, the datasets marked as "SSL" come from the benchmarks of the book "Semi-Supervised Learning" [3], while those of "COIL" and "UCI" come from Columbia image library and UCI machine learning repository respectively. We denote by "O" the original data and use the notations of "ISO" and "SDA" to respectively denote the pre-processed data by ISOMAP and SDA algorithms. The "O+MC", "ISO+MC" and "SDA+MC" correspond to the pre-processed data by MC, ISOMAP+MC and SDA+MC, which have been detailed in subsection 2.3. Additionally, the Gaussian kernel and its shrinkage kernel are always used to test the kernelized MC. In order to check how different the inter-class separabilities are between using and not using MC, and to measure a datum D, we define its ratio of intra-class to inter-class scatters (the smaller ratio, the better separability) as follows:

$$Ratio(D) = \frac{\underset{label(x)=label(x')}{average} \|x - x'\|}{\underset{label(x)\neq label(x')}{average} \|x - x'\|}$$

Owing to its independence on a specific classifier, MC is evaluated by the classification accuracies of 8 classifiers: nearest neighbor (NN), nonlinear-kernel nearest neighbor (KNN), linear-kernel SVM (LSVM), nonlinear-kernel SVM (KSVM), Laplacian SVM (LapSVM), J48 decision tree (J48), naïve Bayes (NB) and radial base function network (RBFNet).

Table 1 Characteristic descriptions of the 10 datasets

dataset	# of classes	# of dimensions	# of instances	comment	source
3-Lines	3	2	363	artificial	
3-Spirals(2D)	3	2	603	artificial	
3-Spirals(3D)	3	3	378	artificial	
24-Ducks	1	1024	24	artificial	COIL
24-Arrows	1	1024	24	artificial	COIL
Digit-1	2	241	1500(2×750)	real-world	SSL
COIL2	2	241	1500(734+766)	real-world	SSL
USPS	3	256	900(3×100)	real-world	UCI
Control	3	60	300(3×100)	real-world	UCI
Corel	3	89	300(3×100)	real-world	UCI

In the next subsection, we will firstly give some illustrations and tests on artificial data.

4.2 Illustrations and tests on artificial data

To help understand the behaviors of MC further, Fig. 4 shows three root shapes of artificial data 3-Spirals(3D), 3-Spirals(2D) and 3-Lines. Specifically, the top and bottom rows consist of the original data and their root shapes respectively, where, the corresponding shrinkage parameters α are given in parenthesis. From Fig. 4, we can clearly observe that the intra-manifold points become more compact and the inter-manifold margins become a bit larger.

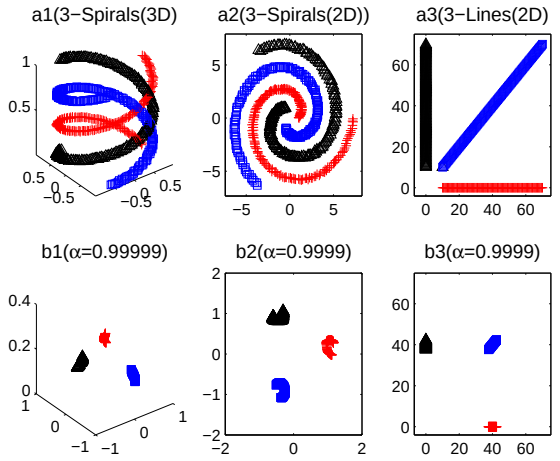

Fig. 4 Top row: 3-Spirals(3D), 3-Spirals(2D) and 3-Lines; Bottom row: the corresponding root shapes.

Another visual example is shown in Fig. 5, whose upside and underside subfigures respectively correspond to 24-Ducks and 24-Arrows after MC. In the respective first row of two subfigures, the original images of both Ducks and Arrows are sampled from the rotated manifold. Next from the top down, each row respectively shows the images of the corresponding root shapes with α orderly taken as $0.7 \rightarrow 0.8 \rightarrow 0.9 \rightarrow 0.95 \rightarrow 0.99$. We can observe that the images of Duck and Arrow are gradually congregated, and they collapse to two fixed-points in the respective bottommost row. This indicates that MC can squeeze the close patterns into a small space, and thus to generate a more compact pattern representation in original space.

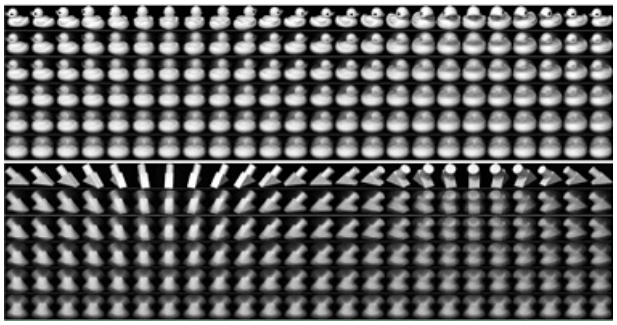

Fig. 5 The contracted images of 24-Ducks (upside subpart) and 24-Arrows (underside subpart) with $\alpha = 0 \rightarrow 0.7 \rightarrow 0.8 \rightarrow 0.9 \rightarrow 0.95 \rightarrow 0.99$ from the top down orderly

Now, we begin to compare the classification accuracies between without and with MC on 3-Spirals(3D). In Fig. 6, we draw three plots against the number of labeled samples for each class increasing from 2 to 20. Where, ISO(1D) and SDA(1D) are the 1-dimension embeddings of the original 3-Spirals(3D) yielded by

ISOMAP and SDA respectively. More specifically, the left plot shows 6 accuracies of the NN classifier on O, O+MC, ISO(1D), ISO(1D)+MC, SDA(1D) and SDA(1D)+MC respectively, and each of them is averaged over 10 trials; the middle plot shows 6 ratio measures for the 6 corresponding data; the right plot shows three α values optimized by AMC criterion in O+MC, ISO(1D)+MC and SDA(1D)+MC respectively. From Fig. 6, we can observe that: 1) the accuracies with MC outperform those without MC correspondingly; 2) the ratios on MC-ed data achieve a more significant degradation than those before MC; 3) three α values gradually approach to 1 (but unequal to 1) as the number of labeled samples increases, implying that, for an ideal manifold data here, the MC encourages a strong contraction if more labeled samples are provided.

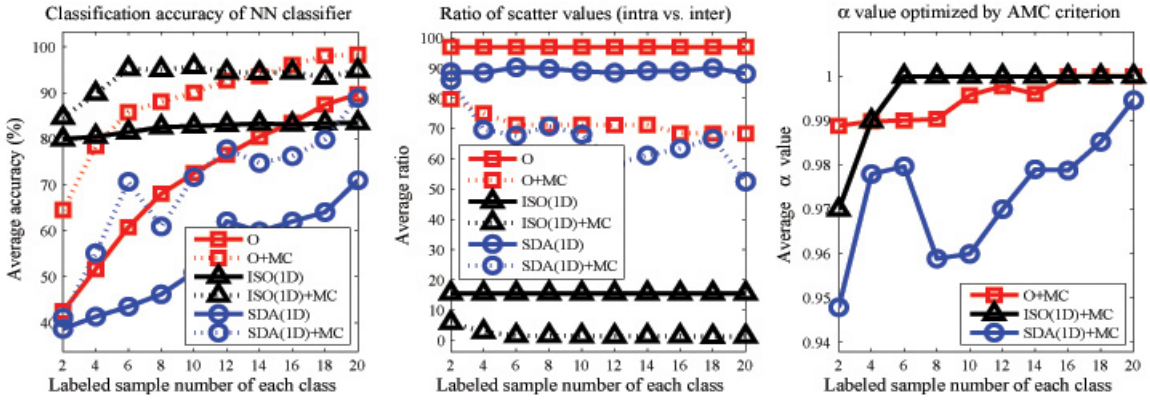

Fig. 6 Three plots against the number of labeled samples increasing from 2 to 20 for 3-Spirals(3D). (left) six accuracies(%) of NN classifier; (middle) six ratio-measures of intra-class vs. inter-class scatters, (right) three α values optimized by adaptive MC criterion.

Table 2 Comparative ratios and accuracies (%) of without vs. with MC on 3-Spirals(3D)

	Labeled # = 5%						Labeled # = 10%					
Data	O	O+MC*	ISO	ISO+MC*	SDA	SDA+MC*	O	O+MC*	ISO	ISO+MC*	SDA	SDA+MC*
Ratio	97.00	**71.23**	15.57	**1.60**	91.03	**62.18**	97.00	**71.21**	15.57	**1.43**	89.51	**57.12**
NN*	63.42	**86.69**	81.76	**95.85**	42.69	**68.43**	78.29	**92.95**	82.92	**94.25**	61.39	**76.76**
KNN*	63.42	**78.96**	81.76	**100.00**	42.69	**48.82**	78.29	**87.94**	82.92	**100.00**	61.39	**78.58**
LSVM*	34.12	**59.97**	80.92	**84.90**	32.97	**51.18**	37.73	**60.71**	82.39	**84.99**	36.11	**51.80**
KSVM*	35.88	**56.47**	80.39	**100.00**	33.61	**39.16**	37.82	**56.76**	81.50	**100.00**	37.35	**61.45**
J48*	40.14	**71.01**	79.75	**87.28**	39.27	**59.78**	48.97	**80.00**	83.16	**89.14**	47.91	**64.48**
NB*	32.80	**61.96**	81.68	**95.69**	33.59	**53.14**	31.92	**62.36**	83.54	**95.84**	37.46	**57.32**
RBFNet*	32.41	**59.97**	80.78	**86.86**	33.03	**50.53**	32.09	**58.44**	80.32	**88.05**	33.89	**50.24**
LapSVM*	92.77	**95.52**	82.91	**100.00**	48.80	**57.62**	97.05	**98.44**	84.54	**100.00**	62.15	**79.20**

For comparison of the classification accuracies of all 8 classifier on O, O+MC, ISO(1D), ISO(1D)+MC, SDA(1D) and SDA(1D)+MC of 3-Spirals(3D), we randomly label 5% and 10% samples respectively and thus get the optimized αs by AMC. As a specific sign, we mark a classifier "C" as "C*" (e.g. NN*) if all accuracies with MC consistently exceed those without MC in C's row, and we mark a datum "D" as "D+MC*" (e.g. ISO+MC*) if all accuracies in D+MC's column consistently exceed those in D's column.

Having been averaged over 10 trials, the classification accuracies are tabulated in Table 2. Comparing "with MC" with "without MC" in Table 2, we observe that: 1) although different classifiers get different gains, the accuracies of all 8 classifiers are overall improved on the MC-ed data no matter whether labeled # = 5% or 10%; 2) all the ratios significantly decrease after MC, and an evident correspondence is "the smaller the ratio, the higher the accuracy".

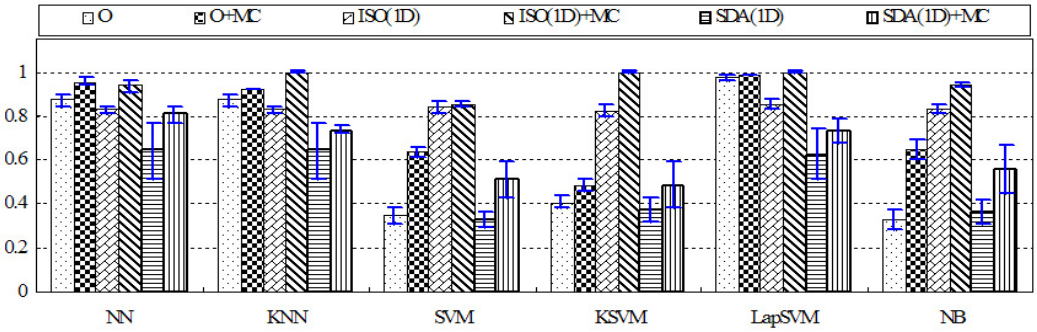

Fig. 7 A comparative overview (without vs. with MC) of the average mean and variance of accuracies, both of which are averaged over the labeled sample number increasing from 2 to 50 with increment 1.

In order to investigate the stability and consistency of MC, we compare the classifiers' means and variances of accuracies between with and without MC. Both of them are averaged over the labeled-sample number increasing from 2 to 50 with increment 1. In Fig. 7, we show the means and variances of six classifiers by 6-group histograms. We observe that all means after MC outperform those before MC and the MC-ed variances are less than those before MC except KSVM's and NB's groups.

As a summary, MC works well on the artificial data here, suggesting that this datum fits our assumption: its intra-class points nearly reside on a manifold while its inter-class points distribute on different manifolds. Below, we will test MC in noisy settings.

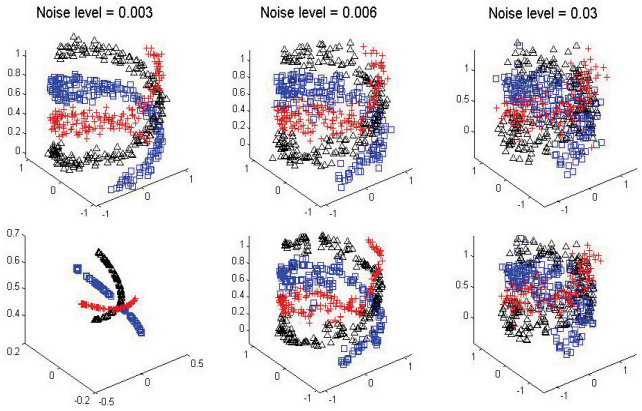

Fig. 8 Top row: the noisy 3-Spirals(3D)s with noise bandwidth 0.003, 0.006 and 0.03 in order. Bottom row: the corresponding root shapes of the noisy 3-Spirals(3D)s.

4.3 Examination of robust performance of MC

In order to carry out a robust analysis of our adaptive MC (AMC), we add different-level Gaussian noises with different bandwidths to artificial data 3-Spirals(3D). We randomly label 5% samples and leave the rest as unlabeled samples in our robust examination. The experimental results are respectively depicted in Figs. 8∼10. In Fig. 8, we add different-level noises with bandwidth 0.003, 0.006 and 0.03 orderly to 3-Spirals(3D) in the top row. After AMC, the corresponding root shapes are displayed in the bottom row, in which it shows a locally and globally consistent contraction, a local accumulation or contraction and almost no contraction respectively for the noise levels in 0.003, 0.006 and 0.03. Moreover, In Fig. 9, we can observe that the shrinkage parameter α drops as noise grows.

As a summary, Figs. 8∼9 partially demonstrate that AMC can adapt to different noise settings in some degree and hence shows a robust behavior. For testing classification accuracy in different noise levels, we run NN classifier respectively on O and O+MC for 3-Spirals(3D). After 20 trials, we display the comparative classification accuracy in Fig. 10. It can be observed that the accuracies with MC outperform those without MC in the noise-free and low-level noisy cases. But as noises are added more and more, two accuracies gradually drop and become closer (even overlapping) to each other in the high-level noises. This result indicates that 1) our adaptive MC is relatively insensitive to noise; 2) MC can well perform its function under the noise-free or low-level noisy circumstances.

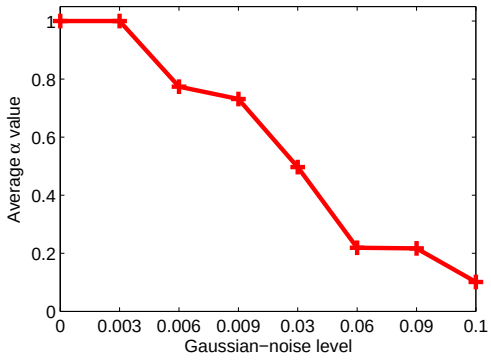

Fig. 9 Different αs adaptively determined by AMC on the noisy 3-Spirals(3D).

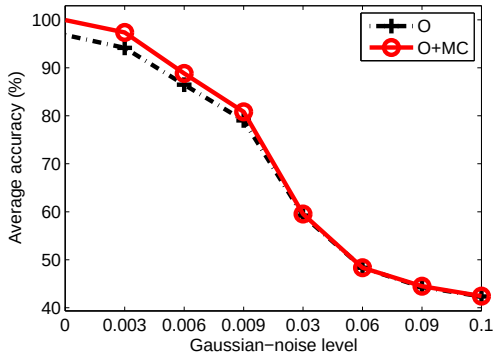

Fig. 10 The comparative classification accuracies of NN classifier respectively on O and O+MC for the noisy 3-Spirals(3D).

As a practical application, we will further verify MC's effectiveness on real-world datasets in next subsection.

4.4 Experiments on the real-world datasets

We take 5 real-world datasets Digit-1, COIL2, USPS, Control and Corel as our testing benchmark, whose citations can be found in refs. [3][7][18][34]. Concretely, **Digit-1** is started from handwritings of digit '1', but they are perturbed through five freedoms: two for translation, one for rotation, one for line thickness, and one for the length of a small line added at bottom. The class labels of Digit-1 are set according to the tilt angle with the boundary to an upright digit; **COIL2** is originally from the Columbia object image library, and it is also perturbed including 1) down-sampling the red channel of each image to 16×16 pixels by averaging over blocks of 88 pixels, and randomly selecting and permuting 241 columns; 2) adding each column a random bias drawing from $N(0,1)$; 3) multiplying each column by a value from uniform $[-1,-0.5] \cup [0.5,1]$; 4) adding an independent noise from $N(0,2I)$ to each row. Additionally, the 24 objects of COIL2 are partitioned into six groups, i.e., four objects each. As a binary classification problem, each new class contains three groups; **USPS** includes 10-class handwriting digits $0 \sim 9$, we just choose digits "2", "3", "5" classes. After randomly sampling 300 points from each class, 900 samples in total as a dataset are used here; **Control** contains 6 classes and each class consists of 100 samples, from which we select three classes (the second, third and sixth) in our experiment; **Corel** is often applied in image retrieval or image classification. Here, we choose 3 classes from it after randomly sampling 100 instances from each class.

When testing MC on ISO and SDA, we have to firstly estimate the intrinsic dimensionality for ISOMAP and SDA algorithm. We select three intrinsic dimensionality estimators *maximum likelihood estimator*, *eigenvalue-based estimator* and *geodesic minimum spanning tree estimator* [24][17][12] here, and the final intrinsic dimensionality is evaluated as the rounding of the average value of these three estimators. In this way, the calculated intrinsic-dimensionalities are respectively 20 for Digit-1, 24 for COIL2, 11 for USPS, 10 for Control and 8 for Corel.

In order to assess the classification accuracy, we partition a datum into the labeled and unlabeled samples for training and predicting respectively. For Digit-1 and COIL2, we follow those in ref. [3], where two their partitions correspond to the labeled samples number = 10 and 100 respectively and each partition contains 12 random realizations. For USPS, Control and Corel, we also give two partitions of the labeled sample number = 5% and 10%, and 12 realizations are randomly generated for each partition. Thus, each following classification accuracy is averaged over such 12 realizations correspondingly. For these five data, we tabulate their experimental results in the following Tables 3∼7 orderly.

Table 3 Comparative ratios and accuracies (%) of without vs. with MC on Digit-1

	Labeled # = 10						Labeled # = 100					
Data	O	O+MC	ISO	ISO+MC	SDA	SDA+MC	O	O+MC	ISO	ISO+MC	SDA	SDA+MC*
Ratio	93.29	**57.96**	82.23	**50.43**	59.48	**46.47**	93.29	**57.96**	81.80	**50.39**	45.54	**27.35**
NN*	76.53	**91.72**	81.49	**93.37**	73.15	**80.84**	93.88	**97.30**	94.51	**97.32**	85.57	**94.52**
KNN*	87.87	**93.17**	81.49	**92.97**	73.15	**81.23**	96.07	**97.37**	94.51	**97.33**	85.57	**94.33**
LSVM	**77.02**	60.67	**82.79**	60.15	**73.14**	69.57	**92.04**	50.11	**95.29**	50.11	85.58	**94.54**
KSVM	**82.22**	60.53	**69.08**	60.23	73.15	**82.46**	**97.60**	50.11	**94.27**	50.11	85.57	**94.38**
J48*	55.83	**84.85**	77.48	**83.97**	50.70	**58.80**	78.40	**95.89**	90.95	**94.73**	64.83	**92.89**
NB*	68.55	**89.83**	65.40	**80.42**	50.87	**79.69**	94.18	**97.20**	88.70	**96.22**	55.40	**94.71**
RBFNet*	61.26	**86.01**	67.51	**87.49**	73.14	**81.83**	75.13	**94.95**	79.02	**94.29**	85.57	**94.74**
LapSVM	**94.28**	59.26	90.56	**95.60**	76.22	**84.17**	**97.71**	58.40	**97.23**	96.58	85.22	**94.45**

Table 4 Comparative ratios and accuracies (%) of without vs. with MC on COIL2

	Labeled # = 10						Labeled # = 100					
Data	O	O+MC	ISO	ISO+MC	SDA	SDA+MC*	O	O+MC	ISO	ISO+MC	SDA	SDA+MC*
Ratio	95.57	**90.55**	95.96	**86.28**	80.47	**77.76**	95.57	**86.94**	96.51	**81.36**	55.91	**42.42**
NN*	56.97	**60.37**	58.54	**60.89**	56.80	**58.32**	88.10	**94.34**	87.16	**94.79**	79.97	**84.73**
KNN	**64.16**	63.40	58.54	**61.63**	56.80	**58.05**	95.08	**95.73**	87.16	**94.91**	79.97	**85.44**
LSVM*	56.29	**60.55**	58.01	**62.81**	56.81	**58.32**	80.78	**92.35**	67.81	**80.81**	79.97	**85.12**
KSVM	62.44	**62.53**	**51.39**	51.18	56.81	**57.67**	**95.97**	90.66	**61.23**	54.18	79.97	**85.16**
J48*	52.34	**60.16**	56.30	**58.03**	50.85	**55.86**	70.17	**87.96**	76.87	**86.75**	62.84	**84.84**
NB	54.63	**58.04**	57.13	**60.15**	50.41	**56.81**	**62.65**	62.62	**64.85**	60.66	55.25	**85.22**
RBFNet*	50.55	**51.09**	50.03	**50.41**	56.79	**57.32**	54.28	**55.90**	50.99	**53.93**	79.99	**85.08**
LapSVM	63.20	**64.34**	59.78	**61.40**	57.13	**58.35**	**95.97**	95.04	**81.84**	79.83	79.24	**84.99**

Table 5 Comparative ratios and accuracies (%) of without vs. with MC on USPS

	Labeled # = 5%						Labeled # = 10%					
Data	O	O+MC	ISO	ISO+MC	SDA	SDA+MC*	O	O+MC	ISO	ISO+MC	SDA	SDA+MC*
Ratio	87.71	**50.71**	64.39	**43.84**	46.23	**26.41**	87.71	**50.71**	64.39	**43.84**	43.09	**23.86**
NN*	83.93	**92.64**	85.81	**92.20**	87.37	**93.13**	88.21	**93.32**	89.48	**93.14**	88.47	**93.31**
KNN*	83.93	**92.58**	85.81	**92.33**	87.37	**93.11**	88.21	**93.38**	89.48	**93.06**	88.47	**93.73**
LSVM*	86.91	**92.77**	84.90	**93.60**	87.37	**93.30**	89.79	**93.67**	87.59	**94.32**	88.47	**93.40**
KSVM	**84.96**	83.65	**88.01**	83.56	87.39	**93.15**	**89.32**	82.86	**90.65**	83.83	88.47	**94.10**
J48*	74.26	**89.81**	75.12	**83.04**	50.08	**86.51**	76.58	**91.78**	83.04	**88.60**	52.05	**90.89**
NB*	74.77	**91.38**	79.93	**87.75**	38.22	**92.60**	77.74	**92.36**	85.26	**91.44**	43.40	**93.46**
RBFNet*	63.96	**84.67**	57.15	**83.74**	87.43	**92.95**	68.70	**85.74**	56.96	**84.83**	88.41	**93.77**
LapSVM	75.92	**86.02**	**89.82**	84.61	86.42	**93.06**	80.19	**86.85**	92.95	**85.38**	87.70	**94.33**

Table 6 Comparative ratios and accuracies (%) of without vs. with MC on Control

	Labeled # = 5%						Labeled # = 10%					
Data	O	O+MC*	ISO	ISO+MC	SDA	SDA+MC*	O	O+MC*	ISO	ISO+MC	SDA	SDA+MC*
Ratio	35.87	**2.42**	13.77	**1.92**	20.96	**5.89**	35.87	**2.42**	13.77	**1.92**	23.07	**5.84**
NN	96.81	**100.00**	100.00	100.00	96.35	**99.12**	98.44	**100.00**	100.00	100.00	97.78	**99.63**
KNN	96.81	**100.00**	100.00	100.00	96.35	**99.72**	98.44	**100.00**	100.00	100.00	97.78	**99.63**
LSVM	98.84	**100.00**	100.00	100.00	96.35	**99.12**	99.74	**100.00**	100.00	100.00	97.78	**99.63**
KSVM	99.47	**100.00**	**98.32**	95.26	96.35	**99.09**	99.96	**100.00**	**99.96**	95.07	97.78	**99.59**
J48	71.82	**92.32**	**91.72**	84.98	58.77	**80.60**	83.78	**94.74**	**94.59**	92.89	62.22	**88.41**
NB*	94.63	**99.54**	95.09	**99.37**	38.25	**100.00**	99.41	**99.93**	97.93	**99.89**	41.85	**100.00**
RBFNet*	97.09	**100.00**	93.44	**98.53**	96.32	**98.32**	99.74	**100.00**	93.96	**98.22**	97.81	**99.59**
LapSVM*	99.51	**100.00**	99.93	**100.00**	98.60	**100.00**	99.56	**100.00**	99.93	**100.00**	98.37	**99.63**

Table 7 Comparative ratios and accuracies (%) of without vs. with MC on Corel

	Labeled # = 5%						Labeled # = 10%					
Data	O	O+MC*	ISO	ISO+MC*	SDA	SDA+MC*	O	O+MC	ISO	ISO+MC	SDA	SDA+MC
Ratio	71.82	**18.01**	57.34	**16.86**	29.28	**18.16**	71.82	**18.01**	57.34	**16.86**	31.10	**18.95**
NN	86.00	**88.56**	86.53	**88.35**	90.25	**90.60**	**89.56**	88.44	**88.44**	88.00	92.37	92.37
KNN	86.00	**88.70**	86.53	**88.77**	90.25	**90.70**	**89.56**	88.74	88.44	88.44	92.37	**92.67**
LSVM*	88.46	**91.37**	85.93	**90.70**	90.21	**93.26**	92.07	**92.59**	88.07	**90.44**	92.37	**94.52**
KSVM*	89.47	**92.21**	90.04	**91.23**	90.21	**93.44**	91.74	**92.70**	91.11	**92.30**	92.37	**94.85**
J48	69.75	**78.67**	79.68	**81.09**	57.51	**80.28**	79.33	**86.33**	**85.63**	84.00	60.70	**88.89**
NB*	75.72	**87.51**	76.98	**85.96**	39.33	**88.88**	85.30	**91.33**	85.19	**90.30**	45.48	**93.63**
RBFNet*	83.93	**89.44**	83.93	**90.70**	90.14	**93.54**	88.19	**93.26**	89.70	**93.00**	92.30	**94.81**
LapSVM*	79.54	**91.23**	88.25	**91.44**	88.53	**93.12**	79.81	**91.96**	90.15	**92.00**	92.15	**94.30**

From the results in Tables 3∼7, we can make several observations and conclusions as follows:

1) In most cases, the accuracies of "with MC" outperform those of "without MC" in terms of two views. One is that most classifiers get higher accuracies after MC. E.g., those classifiers marked by "*" are consistently improved on the MC-ed data whichever settings for the labeled sample number; the other is that the accuracies of O, ISO and SDA are mostly improved after MC no matter whichever classifier is used. E.g., the MC-ed data marked by "*" result in higher accuracies than those without MC.

2) The ratios of O, ISO and SDA get a significant drop after optimizing the AMC, i.e., ratio(O+MC) $<$ ratio(O), ratio(ISO+MC) $<$ ratio(ISO) and ratio(SDA+MC) $<$ ratio(SDA). It indeed accords with the saying that "the lower ratio, the higher accuracy" for most classifiers used here.

3) Many accuracies of ISO and SDA are smaller than those of O. By contrast, most accuracies of O+MC are consistently higher than those of ISO and SDA for most classifiers, manifesting that MC is often superior to ISOMAP and SDA. Furthermore, after being combined with MC, ISOMAP and SDA are actually improved further.

4) After MC, the most accuracies of the supervised classifiers are upgraded clearly and become comparable to the accuracies of the semi-supervised classifier LapSVM. This partially confirms our conclusion that "as long as all the labeled and unlabeled points in the same class are contracted to a tighter space by MC, then a semi-supervised classification task can be effectively fulfilled by a supervised-classifier".

However, some of the classifiers here also occasionally undergo degradation after MC, as shown in Tables 3∼7, specifically in: 1) LSVM, KSVM and LapSVM on

Digit-1 for both labeled # = 10 and 100; 2) KNN and KSVM on COIL2 for labeled # = 10, and KSVM, NB and LapSVM on COIL2 for labeled # = 100; 3) KSVM on USPS for labeled # = 5% and 10%, LapSVM on USPS for labeled # = 5%; 4) KSVM and J48 on Control for both labeled # = 5% and 10%; 5) NN, KNN and J48 on Corel for labeled # = 10%. Such degradations may be mainly due to four aspects: first, compared with artificial data, the real-world datasets often suffer from the under-sampling [4][27] such that the contracting path will be misguided since MC contracts a manifold data in its distribution direction; secondly, if outliers are also contracted into a smaller space together with the clean samples, then it will be more difficult to separate them again than before MC; thirdly, SVM seems more sensitive to outlier than RBFNet and NB, etc. in that it is more likely to tend to fail if outliers act on its support vectors; finally, a good root shape also involves a proper nearest neighbor parameter—k, and an improper k will naturally lead to a bad root shape unfavorable for the sequent classification.

We also have to point out that almost all graph-based learning methods are often confronted with some common problems besides how to determine the neighbor number k, e.g., how to repair under-sampling density and how to eliminate outliers, etc.. MC inherits such problems due to its dependence on an adjacent graph. It can be foreseen that MC and many other graph-based algorithms such as label propagation can hardly work effectively if the between-class points are seriously overlapping each other. If these common problems can be solved with the invention of some new techniques in future, then our MC can also benefit from such advances.

5 Discussions

In recent years, semi-supervised learning including semi-supervised classification, semi-supervised clustering, semi-supervised dimensionality reduction, etc. has received a great amount of attention. For a dataset bearing a manifold structure, some specific semi-supervised classification methods have been developed, typically including manifold-regularization [2], Markov random walk [21], manifold ranking [33][7], label propagation [37], etc.. Usually, Markov random walk is used to model the process of spreading the label information through a stochastic matrix [21]. The key point of manifold ranking is suggesting "vector-ranking" in Euclidean space in analogy to "page-ranking" [33] and the final ranking-list of all labeled and unlabeled samples can be evolved from an initial incomplete ranking. Label propagation [37] originates from the boundary-value theory of harmonic function, i.e., a harmonic function can be uniquely determined by its boundary values. Based on the assumption that all class labels are generated by a harmonic function, the labels of unlabeled samples can be derived from a few known labels (as boundary value).

The proposed MC is partly inspired by Markov random walk, manifold ranking and label propagation, but it differs from these three methods in the following several aspects: 1) MC aims to spread the samples by a transition matrix, but these three methods all focus on spreading class label by such a matrix; 2) MC can be conveniently kernelized as demonstrated in section 2.2, but these three methods do not associate with kernel trick naturally enough; 3) MC as a pre-processor mainly concerns how to better the data's distribution to benefit the subsequent learning, so it can incorporate with many off-the-shelf classifiers more generally like those in our experiments, but any of these three methods only corresponds to a specific classifier; 4) MC can be implemented even in an unsupervised way (only need a fixed α), but these three methods will surely fail when no initial class labels (or rank values) are given. The above analyses indicate that MC is quite different from Markov random walk, label propagation and manifold ranking.

Another enlightened point of MC originates from the "general function" defined in ref. [34], and the intention in general function is to improve the generalization ability of semi-supervised learning. The operation of general function is to replace each original sample-point with the mean point within its k-nearest neighbors. Thereby, such a general function can be specifically viewed as the first iteration of eq. (1) in subsection 2.1.

6 Conclusions and future woks

In this paper, we propose a new data pre-processor named MC. When just a few labeled samples available, we define an adaptive criterion to optimize the shrinkage parameter for controlling the level of MC. Our staring point lies in the fact that a supervised classifier can effectively fulfill a semi-supervised classification task after reducing the data complexity by MC. The superiorities of MC can be roughly summarized as follows:

First, MC can directly work in the original space, so we can avoid the crux of intrinsic-dimensionality estimation as in DR methods; Second, after MC, the ratio of intra-manifold to inter-manifold scatters can be reduced, thus benefiting the subsequent classification learning. Third, MC can contract the labeled and unlabeled points in the same class to a tighter space, which makes some existing supervised classifiers still work well in semi-supervised classification setting. Finally, MC can also be conveniently combined with DR methods such as ISOMAP and SDA, and the classification performance on the preprocessed data by DR+MC can be further improved.

Likewise, MC can also be applied in clustering and metric learning, etc., which will be the topic of our next research. In addition, we will also further develop a more powerful MC technique with out-of-sample predicting ability and good scalability to large-scale data.

Acknowledgement We thank anonymous reviewers for their valuable comments and suggestions. This work is supported by the National Natural Science Foundation of China (Grant No. 60773061).

1 Belkin M., Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation, 2003, 15(6), 1373-1396.
2 Belkin M., Niyogi P., Sindhwani V. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research, 2006(November), vol(7), 2399-2434.
3 Chapelle O., Schölkopf B., Zien A. Analysis of benchmarks. Semi-Supervised Learning, Eds. Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, MIT Press, 2006, 377-390.
4 Bengio Y., Larochelle H., Vincent V. Non-local manifold parzen windows. In: Advances in Neural Information Processing Systems (NIPS'05), 2005, 18, 115-122.
5 Bengio Y.,Delalleanu O.,Roux N.L. Label propagation and quadratic criterion. Semi-Supervised Learning, Eds. Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, MIT Press, 2006, 193-215.
6 Bishop M. Neural networks for pattern recognition. Oxford University Press, 1995.
7 Breitenbach M., Grudic G. Clustering through ranking on manifolds. In: Proceedings of the 22th International Conferences on Machine Learning (ICML'05), 2005, 73-80.
8 Cai, D., He, X., Han, J. Semi-supervised discriminant analysis. In: Proceedings of 11th International Conference Computer Vision (ICCV'07), 2007, 1-7.
9 Cherkassky V. Model complexity control and statistical learning theory. Natural Computing, 2006, Vol(1), Number 1, 109-133.
10 Duin R.P.W., Pekalska E. Object representation, sample size and data complexity. In M. Basu and T.K. Ho (eds.), Data Complexity in Pattern Recognition, Springer, London, 2006, 25-47.
11 Fisher R.A. The use of multiple measurements in taxonomic problem. Annals of Eugenics, 1936, 7, 179-188.
12 Fukunaga K. Olsen D.R. An algorithm for finding intrinsic dimensionality of data. IEEE Transactions on Computers, 1971, C-20, 176-183.
13 He J.R., Li M.J., Zhang H.J., Tong H.J., Tong H.H., Zhang C.S. Manifold-ranking based image retrieval. In ACM Multimedia, 2004, 9-16.
14 Hotelling H. Analysis of a complex of statistical variables into principle components. Journal of Educational Psychology, 1933, 24, 417-441.
15 Keerthi S.S., Shevade S.K., Bhattacharyya C., Murthy K.R.K. A fast iterative nearest point algorithm for support vector machine classifier design. IEEE Transactions on Neural Networks, 2000, 11(1), 124-136.
16 Lafon S., Lee A.B. Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI'06), 2006, 28(9), 1393-1403.
17 Levina E., Bickel P.J. Maximum likelihood estimation of intrinsic dimension. In: Advances in Neural Information Processing Systems (NIPS'04), 2004, 17, 777-784.
18 Porkaew K., Chakrabarti K., Mehrotra S. Query refinement for multimedia retrieval and its evaluation techniques in MARS. ACM International Multimedia Conference, Orlando, Florida, 1999, Oct 30, vol(4), 235-238.
19 Roweis S. T., Saul L. K. Nonlinear dimensionality reduction by locally linear embedding. Science, 2000, 290, 2323-2326.
20 Schölkopf, B., Smola A.J., Mller K.R. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, 1998, 10(5), 1299-1319.
21 Szummer M., Jaakkola T. Partially labeled classification with markov random walks. In: Advances in Neural Information Processing Systems (NIPS'02), 2002, 15, 945-952.
22 Tenenbaum J.B., de Silva V., Langford J.C. A global geometric framework for nonlinear dimensionality reduction. Science, 2000, vol(290), 2319-2323.
23 Tsang W.I., Kocsor A., Kwok J.T. Efficient kernel feature extraction for massive data sets. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06), 2006, 724-729.
24 Van der Maaten L.J.P. An Introduction to Dimensionality Reduction Using Matlab. Technical Report MICC 07-07. Maastricht University, Maastricht, The Netherlands, 2007.

25 Van der Maaten L.J.P., Postma E.O., van den Herik H.J. Dimensionality Reduction: A Comparative Review. Submitted to Neurocomputing, 2008.
26 Vapnik V. N. The nature of statistical learning theory. Second Edition, 1995.
27 Vincent P., Bengio Y. Manifold parzen windows. In: Advances in Neural Information Processing Systems (NIPS'06), 2002, 15, 825-832.
28 Wang F., Zhang C. S. Label Propagation Through Linear Neighborhoods. IEEE Transactions on Knowledge and Data Engineering (TKDE'08), 2008, Vol(20), 55-67.
29 Weng C. G., Poon J. A Data complexity analysis on imbalanced datasets and an alternative imbalance recovering strategy. Web Intelligence, 2006, 270-276.
30 Yan S., Xu D., Zhang B., Zhang H.J., Yang Q., Lin S. Graph embedding and extension: a general framework for dimensionality reduction. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI'07), vol(29), 2007, 40-51.
31 Zhang D., Zhou Z.H., Chen S.C. Semi-supervised dimensionality reduction. In: Proceedings of the 7th SIAM International Conference on Data Mining (ICDM'07), 2007, 629-634
32 Zhou D., Bousquet O., Lal TN., Weston J., Schölkopf B. Learning with local and global consistency In: Advances in Neural Information Processing Systems (NIPS'04), 2004, 16, 321-328.
33 Zhou D., Weston J., Gretton A., Bousquet O. Schölkopf B. Ranking on data manifold. In: Advances in Neural Information Processing Systems (NIPS'04), 2004, 17, 169-176.
34 Zhou Z.H., Chen K.J., Dai H. B. Enhancing relevance feedback in image retrieval using unlabeled data. ACM Transactions on Information Systems, 2006, 24(2), 219-244.
35 Zhou Z.H., Zhan D., Yang Q. Semi-supervised learning with very few labeled training examples. In: Proceeding of the 22nd AAAI Conference on Artificial Intelligence (AAAI'07), Vancouver, Canada, 2007, 675-680.
36 Zhu X.J., Semi-supervised learning literature survey. Computer Sciences TR 1530, University of Wisconsin-Madison, Last modified on July 19, 2008.
37 Zhu X. J., Ghahramani Z., Lafferty J. Semi-supervised learning using gaussian fields and harmonic functions. In: Proceedings of the 20th International Conferences on Machine Learning (ICML'03), 2003, 912-919.