Neurocomputing 330 (2019) 138–150

Contents lists available at ScienceDirect

Neurocomputing

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

Re-KISSME: A robust resampling scheme for distance metric learning in the presence of label noise

Fanxia Zeng[a,c], Wensheng Zhang[b,c,*], Siheng Zhang[b,c], Nan Zheng[a]

[a] State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[b] Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[c] University of Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Article history:
Received 9 February 2018
Revised 14 September 2018
Accepted 11 November 2018
Available online 16 November 2018

Communicated by Dr Yiming Ying

Keywords:
Resampling scheme
KISSME
Distance metric learning
Label noise

ABSTRACT

Distance metric learning aims to learn a metric with the similarity of samples. However, the increasing scalability and complexity of dataset or complex application brings about inevitable label noise, which frustrates the distance metric learning. In this paper, we propose a resampling scheme robust to label noise, Re-KISSME, based on *Keep It Simple and Straightforward Metric* (KISSME) learning method. Specifically, we consider the data structure and the priors of labels as two resampling factors to correct the observed distribution. By introducing the true similarity as latent variable, these two factors are integrated into a maximum likelihood estimation model. As a result, Re-KISSME can reason the underlying similarity of each pair and reduce the influence of label noise to estimate the metric matrix. Our model is solved by iterative algorithm with low computational cost. With synthetic label noise, the experiments on UCI datasets and two application datasets of person re-identification confirm the effectiveness of our proposal.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Distance metric learning (DML) depicts the intrinsic structure and the correlation between different dimensions of data. As a preprocess step of data analysis, the learned metric can improve the performance of metric-based algorithm, such as clustering, ranking [1] and classification. Many typical DML methods are proposed and successfully used in many applications, especially in image retrieval [2], face recognition [3], image annotation [4], visual tracking [5], and person re-identification [6,7]. For a comprehensive review of distance metric learning, please refer to [8,9].

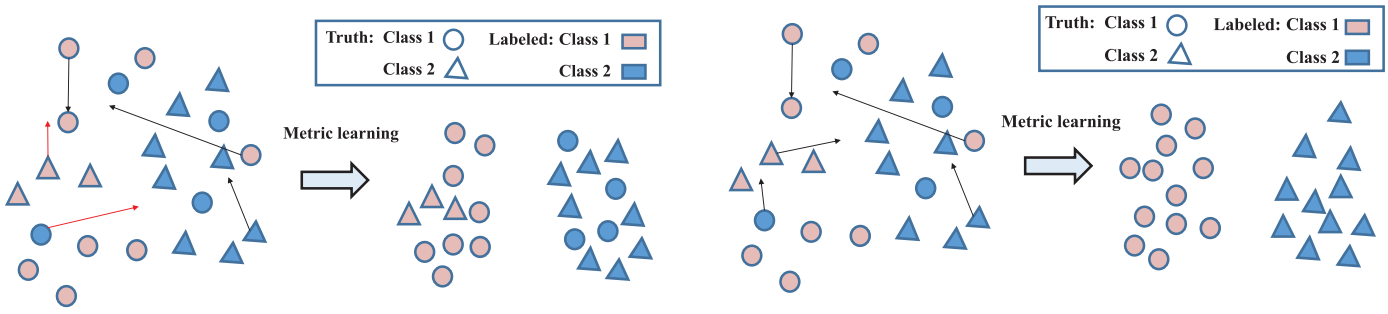
Usually guided by the constraints from the labels or similarity of samples, a good metric aims to keep each group compact and different groups apart. Integrating the constraints as loss terms and the priors as regularization terms into the objective function, most of algorithms transform the task into a complex optimization problem of finding a positive semi-definite matrix. Common DML methods assume that all labels or similarity of samples are correct, which is impossible with the increasing scalability and complexity of dataset or in complex application. As a result, the inevitable existence of label noise affects the performance of DML methods. As shown in Fig. 1, the wrong constraints can mislead the learning process into pushing dissimilar pairs close together and pulling similar pairs far away, which may betray the goal of distance metric learning and lead to poor generalization on testing data.

Learning with noisy labels has been studied for several years. One kind of methods directly remove or relabel the suspected samples according to some predefined criterion. Unfortunately, several work has shown that relabeling the mislabeled samples possibly harms more than removing them [10]. Besides, methods of removing suspicious samples have risk of cleaning right samples or conserving wrong samples, which would mislead the learning as well [11]. The other kind of methods design noise-tolerant classifiers, which usually consider a noise model in addition with the classification model or design specific surrogate loss. However, there is little attention on addressing the problem of label noise in DML besides Robust Neighbourhood Components Analysis (RNCA) [12] and Generalized Maximum Entropy model for learning from noisy side information (GMEns) [13].

One of the most effective and widely used methods in DML is Keep It Simple and Straightforward Metric (KISSME) [14] learning method, which is a statistical proposal assuming that pair differences are sampled from two different Gaussian distributions. This method does not rely on the complex optimization with respect to the *Mahalanobis* matrix and has low computational cost. This

* Corresponding author at: Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.
E-mail addresses: zengfanxia2014@ia.ac.cn (F. Zeng), wensheng.zhang@ia.ac.cn (W. Zhang), zhangsiheng2015@ia.ac.cn (S. Zhang), nan.zheng@ia.ac.cn (N. Zheng).

https://doi.org/10.1016/j.neucom.2018.11.009
0925-2312/© 2018 Elsevier B.V. All rights reserved.



(a) Without considering label noise

(b) With considering label noise

Fig. 1. Without vs. With considering label noise. The red arrows indicate the wrong direction of learning, and the black arrows indicate the right direction of learning.

efficient method depends on the estimation of two covariance matrices, which would be biased by the label noise. The major problem of bias is that pairs of two Gaussian distributions are wrong sampled due to label noise, which makes the final *Mahalanobis* matrix less discriminative. Instead of removal of potential mislabelled samples, which results in loss of information, making use of samples as much as possible is a better choice. Motivated by the work of [15], we introduce a resampling scheme for the sample pairs based on KISSME to optimize the estimation of metric matrix.

In this paper, we propose a robust resampling method based on KISSME, to address the problem of label noise for DML methods with pairwise constraints. We assume that the *Mahalanobis* matrix learned during the iteration can reflect the similarity of sample pairs to some extent, and that label noise is random. Considering the structure of samples and the priors of labels as two resampling factors, the pair differences are resampled to estimate the covariance matrices. By introducing the true similarity as a latent variable, we model these two factors through a maximum likelihood estimation model, and estimate the *Mahalanobis* matrix in each iteration till convergence. The learned metric and the label flipping parameter can reflect the potential true similarity of each pair from feature space and label space respectively, so as to reason the latent true similarity of each pair and reduce the effect of label noise. Also our algorithm has low computational cost. The experiments are conducted on UCI datasets as well as two application datasets of person re-identification with different level of synthetic label noise, and the results validate the effectiveness of our proposal. The main contributions of our work are as follows:

(1) We propose a robust resampling scheme to correct the observed distribution for DML in the presence of label noise, given the observed features and labels of sample pairs;
(2) By modeling the data structure and the prior of label through a maximum likelihood estimation model, an iterative algorithm is developed. Our algorithm Re-KISSME improves the performance of KISSME with and without existence of label noise;
(3) We conduct extensive experiments on general classification task and application of person re-identification with different levels of synthetic label noise. The results show the effectiveness and robustness of our proposal.

The rest of this paper is organized as follows. We discuss related work about DML and learning with noisy labels in Section 2. Reviewing KISSME algorithm is in Section 3. Through introducing the true equivalence of each pair as latent variable in a likelihood estimation model, we propose our method Re-KISSME in Section 4. The experiments on performance are shown in Section 5, and conclusion is in Section 6.

2. Related work

This section reviews some of the previous work on topics closely related to this paper. We first briefly review DML in this section, and then the studies of learning with noisy labels are followed.

2.1. Distance metric leaning

From the way of solver, one type of related work involves convex or nonconvex optimization, which is usually solved by iterative gradient-based methods. Mahalanobis metric for clustering (MMC) [16] formulates the problem as a convex optimization learning from side-information, which arouses the later work of DML. Neighborhood Component Analysis (NCA) [17] maximizes the probability that each data sample selects the points of same label as neighbours. Large Margin Nearest Neighbor (LMNN) [18] aims to penalize large distances between anchor point to its target neighbors and small distances between anchor point to its impostors for k-NN classification. Information Theoretic Metric Learning (ITML) [19] minimizes the relative entropy between two Gaussians under pairwise constraints, which is solved by an iterative Bregman projection algorithm. Parametric Local Metric Learning (PLML) [20] learns local metrics consisting of basis metrics, based on anchor points from different regions of the instance space. Based on the similar triplet constraints of LMNN, much work extends LMNN from different views, such as GB-LMNN [21], χ^2-LMNN [21]. Similarly, Unified Multi-Metric Learning (UM^{2}L) [22] is a framework combining multiple types of metrics from different perspectives under triplet constraints. These methods all optimize the objection in positive direction for similar pairs while in negative direction for dissimilar pairs [14]. They assume that the labels or constraints are correct, whose objective function would be violated by the phenomenon of label noise.

The other type of related work needs no complex optimization procedure. Relevant Components Analysis (RCA) [23] learns a global linear transformation matrix from the positive pairs, which of solution is the inverse of the average chunklet covariance matrix. Discriminative Component Analysis (DCA) [2] improves RCA by considering negative pairs in addition, whose objective function is the ratio of determinants of two covariance matrices. Xiang et al. [24] extends DCA by exploring the trace of two covariance matrix based on both similar and dissimilar pairs, which is solved by eigenvalue decomposition. Assuming the pair differences are sampled from two Gaussian distributions, KISSME is an efficient way to learn a metric matrix from pairwise constraints. The solver of these methods relied on the estimation of covariance matrices, which would still be biased in the presence of label noise. There also has been study showing that the large noise in the pairwise constraints could seriously deceive the DML and lead to poor

generalization on testing samples [25], as well as in our empirical study.

2.2. Learning with noisy labels

The methods designed for the existence of label noise can be roughly categorized into two categories. One kind of them are data-oriented methods, which add data preprocessing steps before classification. The other kind are model-based methods, which construct the model without data preprocessing. Survey refers to [26].

The first kind of methods delete or relabel the mislabeled samples, and the classifier of each method is learned on the remaining training samples. The suspicious samples are selected according to some criterion, such as gain criteria [27], complexity measure [28], and geometrical structure [29]. There are researches showing that removing suspicious samples is more efficient than relabeling them. But this category of methods may lead to removing correct samples or remaining wrong samples, which would harm the performance of classification more. Regression based Distance Metric Learning (RDML) [25], as a kind of DML method for crowdsource, solves the problem via filtering noisy pairwise constraints and recovering the similarity matrix through matrix completion algorithm. Note that RDML differs from our work in that the crowdsource offers multi-annotator for each sample, which could reason the confidence of label directly. Our algorithm explores the problem of DML given one label for each sample, which gives no confidence of each label directly.

The model-based methods focus on designing specific surrogate loss function robust to label noise, or considering a noise model besides the classification model. Natarajan et al. [30] propose two methods, an unbiased estimators of any loss and a weighted loss function, to modify any given surrogate loss function for the class-conditional situation. Liu and Tao [31] prove that using importance reweighting in any surrogate loss function is effective for classification with noisy labels , which is like transfer learning. The Labeled Instance Centroid Smoothing (LICS) [32] approach reduces the influence of noisy labels through incorporating labeled instance centroid and considering the influence of variance. For multiclass classification problem of deep Neural Networks, Ghosh et al. [33] derive some sufficient conditions on a loss function, which would be inherently tolerant to label noise. Some model-based methods are solved by iterative algorithm, which consider a noise model and a classification model simultaneously in the training. The probabilistic methods are widely used in modeling the noise process, which reason the label flipping probability. Based on EM algorithm, Lawrence and Schölkopf [34] assign a label flipping rate of each sample and provide a data-generative process probabilistically for kernel Fisher Discriminant in the presence of label noise. This method has inspired much related work, such as probabilistic Kernel Fisher (PKF) [35]. The robust logistic regression (rLR) [36] model learns the label flipping probabilities and a logistic regressor for classifier simultaneously. A new robust boosting (rBoost) [37] algorithm is designed by employing a label-noise robust base learner and modifying the exponential loss. Recently Raykar et al. [15] estimate the true labels of subjects through an EM approach for learning from crowds, which has inspired much later work. However, these methods are not proposed for the problem of DML and most of them are designed for binary classification.

Although not yet a popular research direction in DML literatures, there is some model-based work attempting to consider learning a metric in the presence of label noise. GMEns [13] proposes a framework for learning from noisy side information to reason the similarity of each pair. RNCA [12] solves the problem of label noise for NCA algorithm, and needs the given information of labels, which is harder to get than side-information. Because the gradient with respect to the metric matrix is involved in iteration, these methods usually require high computational cost.

3. Premilinary

Given the training dataset of n samples in original space: $X = [x_1, \ldots, x_n] \in R^{d\times n}$, the side-information is reformulated as pair relations: $(x_i, x_j) \in \mathscr{S}$ or $(x_i, x_j) \in \mathscr{D}$, where $\mathscr{S}$ denotes the similar pairs and $\mathscr{D}$ denotes the dissimilar pairs. The *Mahalanobis* distance is defined as: $d_M(i,j) = \sqrt{(x_i - x_j)^T M (x_i - x_j)}$, where the metric matrix M is required to be positive semi-definite. We introduce difference vector $x_{ij} = x_i - x_j$ and label y_{ij}: $y_{ij} = 1$ for $(x_i, x_j) \in \mathscr{S}$ or $y_{ij} = 0$ for $(x_i, x_j) \in \mathscr{D}$.

Assuming the similar and dissimilar pairs belong to two different Gaussian distribution with zero mean, KISSME tests the hypothesis H_0 that a pair is dissimilar versus the alternative H_1:

$$f(x_{ij}) = \log \frac{p(x_{ij}|H_0)}{p(x_{ij}|H_1)} = \log \frac{p(x_{ij}|\theta_0)}{p(x_{ij}|\theta_1)} \tag{1}$$

where $p(x_{ij}|\theta_m)$ is a probability of a Gaussian distribution for hypothesis H_m parameterized by θ_m. The large value indicates that x_i and x_j are dissimilar pair, vice-versa small value for a similar pair. So a small value of $f(x_{ij})$ implies that pair (i, j) is similar and H_0 is rejected, a large value implies that H_0 is validated. From the probability of Gaussian distribution:

$$p(x_{ij}|H_m) = \frac{1}{\sqrt{(2\pi)^d \mid \Sigma_m \mid}} exp\Big(-\frac{1}{2} x_{ij}^T \Sigma_m^{-1} x_{ij}\Big), \tag{2}$$

the ratio becomes:

$$f(x_{ij}) = \log \frac{\frac{1}{\sqrt{(2\pi)^d|\Sigma_0|}} exp\big(-\frac{1}{2} x_{ij}^T \Sigma_0^{-1} x_{ij}\big)}{\frac{1}{\sqrt{(2\pi)^d|\Sigma_1|}} exp\big(-\frac{1}{2} x_{ij}^T \Sigma_1^{-1} x_{ij}\big)}. \tag{3}$$

Stripping the constant terms without relation to x_{ij}, the simple formulation is as follows:

$$f(x_{ij}) = x_{ij}^T \big(\Sigma_1^{-1} - \Sigma_0^{-1}\big) x_{ij} \tag{4}$$

where the covariance matrices are computed as

$$\Sigma_0 = \frac{1}{N_0} \sum_{y_{ij}=0} x_{ij} x_{ij}^T \tag{5}$$

$$\Sigma_1 = \frac{1}{N_1} \sum_{y_{ij}=1} x_{ij} x_{ij}^T. \tag{6}$$

N_0 and N_1 are the number of dissimilar and similar pairs respectively. Through projecting $\Sigma_1^{-1} - \Sigma_0^{-1}$ onto the cone of positive semidefinite matrices, we get the *Mahalanobis* matrix M, and the distance between two samples is

$$f(x_{ij}) = x_{ij}^T M x_{ij}, \tag{7}$$

f_{ij} is in short for convenience.

4. The proposed work

The KISSME algorithm and its variants [38,39] all rely on the estimation of two covariance matrices of pair differences. These methods assume that the labels of pairs are correct. However, making all labels accurate is time-consuming and resource-consuming, which is difficult to meet in practice. In the presence of label noise, the estimation of covariance matrices is biased seriously, which leads to a deviating distribution learned from the training samples and poor performance on the testing samples. Under the circumstances, the goal of DML is to output a *Mahalanobis* matrix M from the observed noisy distribution $\tilde{\mathcal{D}}$:

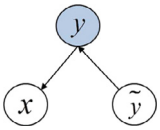


Fig. 2. The latent model representing the relationship among the true label, the observed label and the sample feature.

$\{(x_1, \tilde{y}_1), \ldots, (x_{ij}, \tilde{y}_{ij}), \ldots, (x_N, \tilde{y}_N)\}$, where $\tilde{y}_{ij}$ is the observed similarity of the ijth sample pair.

As mentioned in the related work, the removal of certain samples will cause loss of information in application, it could result in a potential change of the data structure [40]. If there is information telling the label noise from the true label, it will reduce the effect of noise more or less. From the cognitive process of human, people is able to tell the class of noisy samples from the clear samples in vision, this mainly results from the structure information and some prior information. Similarly, given the pair differences and observed labels, we can reason the probability of sample pairs being potential mislabeled and resample each pair without discarding samples. Actually this could correct the observed noisy distribution to learn the underlying true distribution of dataset. We assume that the structure learned during the iterations inflects some part of true distribution to some degree, and that one class is flipped to other classes at random. The structure information corresponds to a classification model, as one factor influencing the resampling scheme. The noise probability corresponds to a label noise model, as another factor.

Label noise model: We introduce a true label y_{ij} for each pair x_{ij} as a latent variable. We only discuss the random noise, which means that the observed label $\tilde{y}_{ij}$ only depends on the true label y_{ij} and is independent of the observed sample pairs x_{ij}. The relationship is shown as Fig. 2. The label noise model is expressed through the probability of conditional distribution. The link between the true label y_{ij} and the given observed label $\tilde{y}_{ij}$ is presented by parameter r_{lm}, namely the flipping probability that is defined as

$$r_{lm} = p(\tilde{y}_{ij} = m | y_{ij} = l). \tag{8}$$

where, $m \in \{0, 1\}$, $l \in \{0, 1\}$ are the value of $\tilde{y}_{ij}$ and y_{ij} respectively. Accordingly, we denote $\tilde{r}_{ml} = p(y_{ij} = l | \tilde{y}_{ij} = m)$.

Classifier model: Even during iteration, the metric matrix can reflect the true similarity of some pairs to some degree. As for another factor influencing resampling scheme, we consider the *Mahalanobis* matrix as the structure information. We use a probability depicting the similarity of each pair difference. Through the *Mahalanobis* distance, it can verify that whether two test samples belong to a group or not. The problem could be characterized as a binary classification. For a threshold b and a scale coefficient k, the classification model is of the form: $y_{ij} = 1$ if $k(f_{ij} - b) \le 0$ and $y_{ij} = 0$ otherwise. f_{ij} is the *Mahalanobis* distance of the ij-th sample pair defined in (7). We hope that the probability of a pair being similar is high when the *Mahalanobis* distance is small, and that the probability of a pair being dissimilar is low when the *Mahalanobis* distance is large. Thus, a logistic sigmoid function is a good representation of the probability describing the similarity of pairs:

$$p(y_{ij} = 1 | x_{ij}, M, k, b) = 1/(1 + e^{k(f_{ij} - b)}). \tag{9}$$

So, we can express the probability of a pair being dissimilar:

$$p(y_{ij} = 0 | x_{ij}, M, k, b) = 1 - p(y_{ij} = 1 | x_{ij}, M, k, b). \tag{10}$$

Here it remarks $\delta_1(x_{ij}) = p(y_{ij} = 1 | x_{ij}, M, k, b)$ and $\delta_0(x_{ij}) = p(y_{ij} = 0 | x_{ij}, M, k, b) = 1 - \delta_1(x_{ij})$.

Parameter learning: Remarks parameter set $\theta = \{M, k, b, r_{lm}\}$. For convenience, we denote $p(y_{ij} | x_{ij}, \theta) = p(y_{ij} = l | x_{ij}, M, k, b)$.

Combining these above two factors and assuming that the distribution of $p(y_{ij} = l)$ is uniform, the pair differences are resampled according to the probability given the difference x_{ij} and observed label $\tilde{y}_{ij}$ for each pair:

$$\begin{aligned} p(y_{ij} | x_{ij}, \tilde{y}_{ij}, \theta) &\propto p(y_{ij} | x_{ij}, \theta) p(y_{ij} = l | \tilde{y}_{ij} = m) \\ &= p(y_{ij} = l | x_{ij}, \theta) \tilde{r}_{ml}. \end{aligned} \tag{11}$$

Thus based on the above uniform assumption of $p(y_{ij} = l)$ and Bayesian theorem

$$p(y_{ij} | x_{ij}, \tilde{y}_{ij}, \theta) = \frac{p(y_{ij} = l | x_{ij}, \theta) p(y_{ij} = l | \tilde{y}_{ij} = m)}{\sum_{l'} p(y_{ij} = l' | x_{ij}, \theta) p(y_{ij} = l' | \tilde{y}_{ij} = m)}. \tag{12}$$

Given the noisy distribution $\tilde{\mathcal{D}}$, our problem is simplified to make inference about the parameter set θ. From the perspective of maximum likelihood, the objective function of our model can be defined as follows:

$$L(\theta) = \sum_{ij=1}^{N} \log p(\tilde{y}_{ij} | x_{ij}, \theta). \tag{13}$$

By using the unknown true label y_{ij} as a latent variable to calculate a posterior $q(y_{ij} | x_{ij}, \tilde{y}_{ij})$, it can derive a lower bound of the log-likelihood

$$L(\theta) \ge \sum_{ij=1}^{N} \sum_{y_{ij}=0}^{1} q(y_{ij} | x_{ij}, \tilde{y}_{ij}) \log \frac{p(\tilde{y}_{ij}, y_{ij} | x_{ij})}{q(y_{ij} | x_{ij}, \tilde{y}_{ij})} \equiv L_q(\theta). \tag{14}$$

The bound of (14) becomes an equality, when $q(y_{ij} | x_{ij}, \tilde{y}_{ij}) = p(y_{ij} | x_{ij}, \tilde{y}_{ij}, \theta)$. We address the above problem through EM-like algorithm, where the algorithm mainly estimates the conditional expectation of true label y_{ij} (E-step) and then optimizes the parameter set θ (M-step).

For E-step, based on $\tilde{r}_{ml} = p(y_{ij} = l | \tilde{y}_{ij} = m)$ and (12), the posterior $q(y_{ij} | x_{ij}, \tilde{y}_{ij})$, which also represents the conditional expectation of y_{ij}, can be computed in the tth iteration as

$$\begin{aligned} q(y_{ij} | x_{ij}, \tilde{y}_{ij}) = p(y_{ij} | x_{ij}, \tilde{y}_{ij}, \theta^t) &= \frac{p(y_{ij} = l | x_{ij}, \theta^t) \tilde{r}^t_{ml}}{\sum_{l'} p(y_{ij} = l' | x_{ij}, \theta^t) \tilde{r}^t_{ml'}} \\ &= \frac{\delta^t_l(x_{ij}) \tilde{r}^t_{ml}}{\sum_{l'} \delta^t_{l'}(x_{ij}) \tilde{r}^t_{ml'}}. \end{aligned} \tag{15}$$

For M-step, after discarding the constant and irrelative terms with respect to parameter set θ, the objective function $L_q(\theta)$ in the t-th iteration becomes as follows:

$$L_{obj}(\theta; \theta^t) = \sum_{ij=1}^{N} \sum_{l=0}^{1} \frac{\delta^t_l(x_{ij}) \tilde{r}^t_{ml}}{\sum_{l'} \delta^t_{l'}(x_{ij}) \tilde{r}^t_{ml'}} \times \log p(\tilde{y}_{ij} = m, y_{ij} = l | x_{ij}, \theta). \tag{16}$$

Because

$$\begin{aligned} p(\tilde{y}_{ij} = m, y_{ij} = l | x_{ij}, \theta) &= p(y_{ij} = l | x_{ij}, \theta) p(\tilde{y}_{ij} = m | y_{ij} = l) \\ &= \delta_l(x_{ij}) r_{lm}, \end{aligned} \tag{17}$$

substitute (17) into (16), the expected log-likelihood of t-th iteration becomes as follows:

$$L_{obj}(\theta; \theta^t) = \sum_{ij=1}^{N} \sum_{l=0}^{1} \frac{\delta^t_l(x_{ij}) \tilde{r}^t_{ml}}{\sum_{l'} \delta^t_{l'}(x_{ij}) \tilde{r}^t_{ml'}} \log \delta_l(x_{ij}) r_{lm}. \tag{18}$$

We discuss the computation of noise rate r_{lm} firstly, and consider the computation of k, b and *Mahalanobis* matrix M later. About r_{lm}, we add constraints $r_{00} + r_{01} = 1$ and $r_{10} + r_{11} = 1$ as Lagrange multipliers. For r_{00}, adding $\lambda_1(1 - r_{00} - r_{01})$, the partial derivation in the tth step is

$$\frac{\partial L_{obj}}{\partial r_{00}} = \sum_{ij=1}^{N} \frac{\delta^t_0(x_{ij}) \tilde{r}^t_{00}}{\sum_{l'=0}^{1} \delta^t_{l'}(x_{ij}) \tilde{r}^t_{0l'}} \frac{\mathbf{1}(\tilde{y}_{ij} = 0)}{r_{00}} - \lambda_1, \tag{19}$$

where, $\mathbf{1}(x)$ is indicative function. Thus:

$$r_{00} = \frac{1}{\lambda_1}\sum_{ij=1}^{N}\frac{\delta_0^t(x_{ij})\tilde{r}_{00}^t\mathbf{1}(\tilde{y}_{ij}=0)}{\sum_{l'=0}^{1}\delta_{l'}^t(x_{ij})\tilde{r}_{0l'}^t}. \tag{20}$$

Similarly for r_{01}, it gets:

$$r_{01} = \frac{1}{\lambda_1}\sum_{ij=1}^{N}\frac{\delta_0^t(x_{ij})\tilde{r}_{10}^t\mathbf{1}(\tilde{y}_{ij}=1)}{\sum_{l'=0}^{1}\delta_{l'}^t(x_{ij})\tilde{r}_{1l'}^t}. \tag{21}$$

Adding Eq. (20) to Eq. (21), and combing $\lambda_1(r_{00}+r_{01})=\lambda_1$, r_{00} is updated in the tth step as

$$r_{00} = \frac{\sum_{ij=1}^{N}\frac{\delta_0^t(x_{ij})\tilde{r}_{00}^t\mathbf{1}(\tilde{y}_{ij}=0)}{\sum_{l'=0}^{1}\delta_{l'}^t(x_{ij})\tilde{r}_{0l'}^t}}{\sum_{ij=1}^{N}\frac{\delta_0^t(x_{ij})\tilde{r}_{00}^t\mathbf{1}(\tilde{y}_{ij}=0)}{\sum_{l'=0}^{1}\delta_{l'}^t(x_{ij})\tilde{r}_{0l'}^t}+\sum_{ij=1}^{N}\frac{\delta_0^t(x_{ij})\tilde{r}_{10}^t\mathbf{1}(\tilde{y}_{ij}=1)}{\sum_{l'=0}^{1}\delta_{l'}^t(x_{ij})\tilde{r}_{1l'}^t}}. \tag{22}$$

Similarly for r_{11} in the tth step, we have

$$r_{11} = \frac{\sum_{ij=1}^{N}\frac{\delta_1^t(x_{ij})\tilde{r}_{11}^t\mathbf{1}(\tilde{y}_{ij}=1)}{\sum_{l'=0}^{1}\delta_{l'}^t(x_{ij})\tilde{r}_{1l'}^t}}{\sum_{ij=1}^{N}\frac{\delta_1^t(x_{ij})\tilde{r}_{11}^t\mathbf{1}(\tilde{y}_{ij}=1)}{\sum_{l'=0}^{1}\delta_{l'}^t(x_{ij})\tilde{r}_{1l'}^t}+\sum_{ij=1}^{N}\frac{\delta_1^t(x_{ij})\tilde{r}_{01}^t\mathbf{1}(\tilde{y}_{ij}=0)}{\sum_{l'=0}^{1}\delta_{l'}^t(x_{ij})\tilde{r}_{0l'}^t}}. \tag{23}$$

and compute $r_{10}=1-r_{11}$, $r_{01}=1-r_{00}$.

Through similar derivation to [34], given the rate $p(\tilde{y}=1)=\tilde{\pi}$, $\tilde{r}_{ml}$ in tth iteration can be got through:

$$\tilde{r}_{10} = \frac{(1-r_{10}-\tilde{\pi})r_{01}}{\tilde{\pi}(1-r_{01}-r_{10})}, \tag{24}$$

$$\tilde{r}_{01} = \frac{(\tilde{\pi}-r_{01})r_{10}}{(1-\tilde{\pi})(1-r_{01}-r_{10})}. \tag{25}$$

Then we consider the optimization of k and b. These two parameters could be optimized individually or meanwhile. However, each of them acts different role in the classification model, which is briefly explained in the definition of classifier model. b is a threshold for the *Mahalanobis* distance, it focuses more on the discrimination. k is a scalar, it focuses more on the scale of distance and influences the rate of convergence more. The probability $\delta(x_{ij})$ of more pairs approaches 0 or 1 when k is large, the algorithm may converge fast and get into local optimization. The probability $\delta(x_{ij})$ of more pairs approaches 0.5 when k is small, the algorithm may converge slowly or not converge within the default maximum iteration. So we optimize b firstly and k later, instead of optimizing them simultaneously. For computation of b, we use probability $\delta_{0(1)}(x_{ij})$, as the soft label instead of hard label in ROC curve, and use the Mahalanobis distance of threshold point as b when the true positive rate is equal to the true negative rate.

k is learned through the conjugate gradient method [41]: $k_{t'+1}=k_{t'}-\alpha d_{t'}$. The step size α is obtained by a line search, and the search direction $d_{t'}$ is updated by the rule: $d_{t'}=-g_{t'}+\beta_{t'}d_{t'-1}$, the scalar $\beta_{t'}$ is computed by: $\beta_{t'}=\frac{g_{t'}^T(g_{t'}-g_{t'-1})}{d_{t'-1}(g_{t'}-g_{t'-1})}$. $g_{t'}$ is the t'th gradient with respect to k, and the gradient is computed through (for convenience, subscript t' omitted):

$$g = \frac{\partial L_{obj}}{\partial k} = \sum_{ij=1}^{N}\left[\frac{\delta_0^t(x_{ij})\tilde{r}_{m0}^t}{\sum_{l'}\delta_{l'}^t(x_{ij})\tilde{r}_{ml'}^t}\delta_1(x_{ij})-\frac{\delta_1^t(x_{ij})\tilde{r}_{m1}^t}{\sum_{l'}\delta_{l'}^t(x_{ij})\tilde{r}_{ml'}^t}\delta_0(x_{ij})\right] \times(f_{ij}-b) \tag{26}$$

After calculating these parameters, we can infer the true similarity of x_{ij} from the given label in the presence of noise according to the probability in (15). Therefore, we resample each ij-th pair according to the rescaled weights $w_{0_{ij}}^t, w_{0_{ij}}^t$ in the two distribution during the tth iteration as

$$w_{0_{ij}}^t = \frac{\delta_0^t(x_{ij})\tilde{r}_{m0}^t}{\sum_{l'=0}^{1}\delta_{l'}^t(x_{ij})\tilde{r}_{ml'}^t} \tag{27}$$

Algorithm 1 Resampling-KISSME.

Input:
$\{(x_{ij})|ij=1,2,\ldots,N\}$: training samples;
$\{\tilde{y}_{ij}\}$: the observed similarity of pairs;
T: the maximal iteration step;
Initialize: $\theta=\{M,k,b,r_{lm}\}$.
while problem(12) does not converge or $t\leqslant T$ **do**
  M-step: optimize the parameter set θ according to Eq. (22) $\sim$ (31), and project (31) onto the cone of positive semidefinite.
  E-step: re-estimate $q(y_{ij}=l|x_{ij},\tilde{y}_{ij}=m,\theta_t)$ according to Eq. (15).
end while
Output: the learned M

and

$$w_{1_{ij}}^t = \frac{\delta_1^t(x_{ij})\tilde{r}_{m1}^t}{\sum_{l'=0}^{1}\delta_{l'}^t(x_{ij})\tilde{r}_{ml'}^t}. \tag{28}$$

Under the resampling scheme, we get

$$\Sigma_0^t = \sum_{ij} w_{0_{ij}}^t x_{ij}x_{ij}^T \tag{29}$$

$$\Sigma_1^t = \sum_{ij} w_{1_{ij}}^t x_{ij}x_{ij}^T \tag{30}$$

and

$$M_t = (\Sigma_1^t)^{-1}-(\Sigma_0^t)^{-1}. \tag{31}$$

Through projecting $(\Sigma_1^t)^{-1}-(\Sigma_0^t)^{-1}$ onto the cone of positive semidefinite matrices to get the *Mahalanobis* matrix of the iteration step t and to finish the t-th iteration. The algorithm iterates till convergence. Due to the resampling scheme, we name the method as Resampling-KISSME (Re-KISSME).

In fact, the Eqs. (29) and (30) imply that the Re-KISSME is a relaxation of KISSME. The resampling weight guidelines the learning with a soft label instead of directly making use of the labels containing noise. Taking the $w_{1_{ij}}^t$ as an example, if the resampling weight $w_{1_{ij}}^t$ approximates to 0.5, the ij-th pair is near the decision boundary and is prone to be misclassified. Otherwise when the probability $w_{1_{ij}}^t$ approximates 1, the pair is far from the decision boundary and is prone to be classified as similar pair. At the same time, this means that we also conserve the weight $w_{0_{ij}}^t$ even approaching 0. We note that the resampling scheme aligns weight on each pair in the estimation both of similar and dissimilar covariance matrix at the same time. Although a pair is rare to be similar or dissimilar, a very low probability of this event is a more reasonable estimation than the zero weight. This resampling scheme also makes that the covariance matrices need no regularization. Besides, there is no need to compute the gradient with respect to *Mahalanobis* matrix in Re-KISSME, resulting from the efficiency of KISSME. This makes the algorithm fast, which is validated in our experiment.

5. Experiment

To validate the effectiveness of our proposed algorithm, we conduct three series of experiments on nine benchmark data sets. The first one is conducted on seven datasets downloaded from the UCI machine learning data repository [42], including Breast Tissue, Statlog (Heart), Iris, Parkinsons [43], Protein [16], Seeds, Wine and two person re-identification datasets, iLIDS and one camera pair of RAiD. The goals of our experiments are three folds. First, we want

Table 1
The UCI-datasets used in experiment.

Dataset	Sample number (n)	Feature number (d)	Class number (c)
Breast Tissue	106	9	6
Statlog (Heart)	270	13	2
Iris	150	4	3
Parkinsons	195	22	2
Protein	116	20	6
Seeds	210	7	3
Wine	178	13	3

to confirm the comparative behavior with other distance metric. Second, we want to known the performance and degeneration of our algorithm in application. Third, we want to verity the computation complexity of the algorithm.

The original labels of the 9 benchmark data sets are clear. After splitting the training/testing set, we make the testing set clear and generate the label noise of the training set through: (1) select the samples of each class at random according the given noise rate, (2) flip their labels into one of other classes at random. Because some methods make use of the label directly, while other methods including the proposed method make use of the side-information. To make the comparison fair, we flip the similarity of pairs after generating similar or dissimilar pairs from the clean label. Using the observed labels, we use the *Mahalanobis* matrix M of regular KISSME as the initial. We initialize b using the maximum of all *Mahalanobis* distances of all pairs, while initializing k and r by setting $1/k = b,\ r = [0.9, 0.1; 0.1, 0.9]$.

5.1. Experiments on general classification

In experiments on UCI datasets, the details of the datasets are listed in Table 1. Each dataset is divided into 4/1 for training/testing split at random, and the percentage of label noise in training set varies from 0% to 30%. This process is repeated 20 times. To make quantitative analysis, we calculate the average accuracy, standard deviation and mean rank of each method, then use the Friedman test to evaluate the significance of comparison.

We compare our method with several methods, including the 5-NN without DML as the baseline (Eucli), and five state-of-the-art metric learning methods, LMNN, ITML, DML-eig [44], RNCA, KISSME. The RNCA is designed for the label noise. The parameter tuning of compared methods is the same as the original literatures. In each dataset, we pick up 30c(c-1) constraints for each dataset. The results of accuracy are shown from Tables 2–8, as well as mean rank in Table 9.

From these seven tables about accuracy, it is shown that the performance of DML methods degenerates with the increasing noise rate on the several datasets: (1) Table 2 verifies that Re-KISSME is not worse than other methods when the data set is clear. What's more, the accuracy of Re-KISSME is better than other methods at least on six data sets with different noise levels, which accounts to more than half of whole data sets. (2) When there is label noise, our proposal outperforms other method. Especially when the percentage of noise is 5%, 10%, 15%, the accuracy of Re-KISSME dominates at least on six datasets. Over the second best method, our method has an improvement of about 3% on Breast Tissue. (3) As designed for label noise, RNCA performs poor because that it makes use of local structure information and suffers noise more. Re-KISSME outperform regular KISSME in the presence of label noise, which is validated at least on five datasets in these six tables. For example, the gaps between these two method is 3.2% at noise level of 15% on breast tissue.

To analyze the results further, we use the non-parametric statistical analysis, Friedman test to validate whether the comparisons are of significance. The Friedman test is conducted on the rank values of each algorithm at each noise level. The null hypothesis states that all algorithms not perform differently significantly, and post hoc test proceeds if the null hypothesis is rejected. We use the corrected Friedman statistic:

$$F_F = \frac{(N'-1)\chi_F^2}{N'(k'-1)-\chi_F^2} \tag{32}$$

where,

$$\chi_F^2 = \frac{12N'}{k'(k'+1)}\left[\sum_{j'} R_{j'}^2 - \frac{k'(k'+1)^2}{4}\right] \tag{33}$$

The $R_{j'}$ is the rank of j'-th algorithm, k' is the number of all methods, and N' represents the number of data sets. The F_F is distributed according to the F-distribution with $k'-1$ and $(k'-1)(N'-1)$ degrees of freedom.

Calculating the responding statistic of each noise level, the null hypothesis is rejected at risk of $\alpha = 0.05$. So we use the Bonferroni Dunn test to compare our method, which is the control method, with other seven methods by the critical difference (CD):

$$CD = q_\alpha \sqrt{\frac{k'(k'+1)}{6N'}} \tag{34}$$

The corresponding CD is $2.638\sqrt{\frac{7\cdot 8}{6\cdot 7}} = 3.0461$. The Bonferroni-Dunn test and rank differences between Re-KISSME and other seven algorithms is listed in Table 9. It is shown that Re-KISSME significantly performs better than the compared methods in most cases, and that it beats other methods at least with a rank difference of 1 in all cases. The statistic test validates the effectiveness of our method.

5.2. Experiments on iLIDS for person re-identification

It's well known that person re-identification is a challenging problem resulting from its large intra-class variation in view angle, pose, illumination, and occlusion. Feature representation and metric learning are two fundamental problems for person re-identification, and it is suitable to conduct experiments to validate

Table 2
Comparison of average accuracy (%, mean ± std) on UCI-datasets without label noise, the result of rank-1 is in bold face.

	Eucli	LMNN	ITML	DML-eig	RNCA	KISSME	Re-KISSME
Breast tissue	51.60 ± 2.38	57.46 ± 3.70	55.28 ± 2.68	54.64 ± 2.79	51.60 ± 2.38	62.37 ± 2.99	**62.98 ± 3.98**
Statlog (Heart)	66.52 ± 1.45	65.20 ± 1.90	76.72 ± 2.89	67.59 ± 1.27	66.24 ± 1.74	77.07 ± 1.96	**77.72 ± 2.00**
Iris	96.56 ± 0.51	96.69 ± 0.75	96.79 ± 0.61	96.42 ± 1.02	96.35 ± 0.73	96.99 ± 0.69	**97.02 ± 0.78**
Parkinsons	85.05 ± 1.44	82.82 ± 1.64	84.46 ± 1.63	83.59 ± 1.56	85.05 ± 1.44	84.92 ± 2.24	**87.54 ± 1.57**
Protein	69.42 ± 2.00	71.36 ± 2.18	71.41 ± 2.93	71.43 ± 2.11	68.90 ± 3.27	**71.79 ± 2.11**	71.66 ± 2.95
Seeds	88.71 ± 0.88	90.21 ± 0.96	95.33 ± 0.97	90.00 ± 0.93	91.76 ± 1.26	95.60 ± 1.03	**95.74 ± 0.96**
Wine	69.40 ± 2.05	90.75 ± 2.05	92.46 ± 1.49	76.14 ± 2.81	69.60 ± 1.96	96.77 ± 1.07	**97.06 ± 0.89**
Average	75.32	79.21	81.78	77.12	75.64	83.64	**84.25**
Mean rank	5.5714	5	3.5714	5	5.5714	2.1429	**1.1429**

Table 3
Comparison of average accuracy (%, mean ± std) on UCI-datasets with label noise of 5%, the result of rank-1 is in bold face.

	Eucli	LMNN	ITML	DML-eig	RNCA	KISSME	Re-KISSME
Breast tissue	50.67 ± 3.37	56.13 ± 3.57	53.78 ± 2.74	52.92 ± 4.80	50.67 ± 3.37	58.73 ± 4.13	**61.95 ± 3.65**
Statlog (Heart)	65.57 ± 1.78	63.31 ± 2.67	73.50 ± 3.69	66.59 ± 1.65	65.72 ± 1.71	74.74 ± 2.17	**75.98 ± 2.20**
Iris	95.92 ± 0.87	89.50 ± 1.64	95.39 ± 1.64	95.36 ± 1.03	96.28 ± 1.15	96.56 ± 1.44	**96.85 ± 1.03**
Parkinsons	83.28 ± 1.84	82.10 ± 2.04	83.41 ± 2.52	81.87 ± 1.82	83.28 ± 1.94	84.08 ± 2.14	**86.05 ± 1.80**
Protein	67.13 ± 2.95	66.89 ± 2.63	68.14 ± 3.11	66.63 ± 2.95	66.52 ± 3.08	70.13 ± 2.72	**71.98 ± 2.16**
Seeds	88.88 ± 1.15	90.71 ± 1.31	93.50 ± 1.90	89.38 ± 1.07	90.95 ± 1.46	95.12 ± 1.24	**95.17 ± 1.50**
Wine	69.22 ± 2.80	81.13 ± 2.93	91.25 ± 2.27	70.20 ± 2.99	69.38 ± 2.63	95.88 ± 0.96	**97.24 ± 0.97**
Average	74.38	75.68	79.85	74.71	74.69	82.12	**83.60**
Mean rank	5.5714	5.2857	3.4286	5.5714	5.1429	2	**1**

Table 4
Comparison of average accuracy (%, mean ± std) on UCI-datasets with label noise of 10%, the result of rank-1 is in bold face.

	Eucli	LMNN	ITML	DML-eig	RNCA	KISSME	Re-KISSME
Breast tissue	49.89 ± 4.39	54.97 ± 3.91	52.01 ± 4.61	52.51 ± 3.76	49.89 ± 4.39	58.69 ± 4.60	**61.15 ± 3.96**
Statlog (Heart)	64.43 ± 2.50	62.74 ± 2.30	69.43 ± 3.71	63.69 ± 2.60	64.39 ± 2.41	73.20 ± 2.39	**74.65 ± 2.16**
Iris	95.09 ± 1.00	90.80 ± 1.80	94.96 ± 1.35	94.27 ± 1.53	94.96 ± 1.72	94.96 ± 1.49	**96.02 ± 1.08**
Parkinsons	82.33 ± 2.25	80.85 ± 2.19	82.05 ± 1.87	80.26 ± 1.85	82.28 ± 2.17	81.59 ± 2.73	**83.69 ± 2.11**
Protein	66.42 ± 2.47	62.65 ± 4.59	65.51 ± 3.94	63.80 ± 2.82	65.30 ± 2.70	68.76 ± 2.40	**69.64 ± 2.80**
Seeds	88.57 ± 1.34	89.52 ± 1.34	91.43 ± 2.05	88.50 ± 1.45	89.76 ± 1.66	94.02 ± 1.22	**94.24 ± 0.89**
Wine	67.88 ± 1.94	76.12 ± 2.69	88.66 ± 2.58	69.41 ± 1.83	67.88 ± 1.94	95.18 ± 1.50	**96.38 ± 0.81**
Average	73.51	73.95	77.72	73.20	73.49	80.92	**82.25**
Mean rank	4.2857	5.5714	3.7143	5.8571	5	2.5714	**1**

Table 5
Comparison of average accuracy (%, mean ± std) on UCI-datasets with label noise of 15%, the result of rank-1 is in bold face.

	Eucli	LMNN	ITML	DML-eig	RNCA	KISSME	Re-KISSME
Breast tissue	49.03 ± 4.69	53.06 ± 3.65	52.81 ± 5.03	51.01 ± 5.05	49.03 ± 4.69	56.94 ± 4.24	**59.34 ± 5.03**
Statlog (Heart)	62.81 ± 1.95	60.54 ± 3.00	67.30 ± 3.65	63.19 ± 2.64	62.96 ± 2.02	70.20 ± 3.30	**72.04 ± 3.84**
Iris	92.69 ± 1.91	89.74 ± 2.80	91.55 ± 2.11	92.31 ± 1.78	91.48 ± 2.18	92.38 ± 2.73	**93.38 ± 2.54**
Parkinsons	80.79 ± 2.20	79.36 ± 1.87	80.33 ± 2.83	80.87 ± 2.38	80.79 ± 2.20	81.03 ± 1.60	**82.74 ± 2.66**
Protein	65.49 ± 2.82	60.63 ± 5.74	65.46 ± 2.66	61.93 ± 2.91	63.93 ± 3.14	68.21 ± 3.42	**70.47 ± 2.39**
Seeds	87.69 ± 1.19	88.62 ± 1.34	89.71 ± 1.88	87.81 ± 1.41	88.55 ± 1.40	91.74 ± 1.77	**92.43 ± 2.01**
Wine	66.95 ± 3.01	71.90 ± 3.53	85.56 ± 3.93	68.13 ± 2.65	66.95 ± 3.01	93.15 ± 1.04	**94.21 ± 1.80**
Average	72.20	71.98	76.10	72.18	71.96	79.10	**80.66**
Mean rank	5.0714	5.5714	4	4.7143	5.5	2.1429	**1**

Table 6
Comparison of average accuracy (%, mean ± std) on UCI-datasets with label noise of 20%, the result of rank-1 is in bold face.

	Eucli	LMNN	ITML	DML-eig	RNCA	KISSME	Re-KISSME
Breast tissue	47.52 ± 2.74	51.86 ± 5.11	49.60 ± 4.65	48.63 ± 3.04	47.52 ± 2.74	55.86 ± 4.01	**59.07 ± 5.72**
Statlog (Heart)	62.28 ± 2.44	59.57 ± 2.52	65.33 ± 3.35	61.85 ± 3.15	62.04 ± 2.67	67.72 ± 3.97	**68.76 ± 3.38**
Iris	91.32 ± 1.53	88.42 ± 2.17	90.76 ± 1.60	90.76 ± 1.34	90.62 ± 2.43	90.74 ± 1.82	**92.29 ± 1.58**
Parkinsons	78.15 ± 1.94	77.77 ± 1.89	77.18 ± 1.99	78.08 ± 1.87	78.15 ± 1.94	76.54 ± 2.65	**79.00 ± 3.67**
Protein	62.11 ± 3.08	56.50 ± 4.61	61.92 ± 3.48	57.61 ± 3.23	60.15 ± 3.38	**66.52 ± 3.58**	66.13 ± 3.33
Seeds	84.69 ± 1.94	85.26 ± 1.65	86.88 ± 2.30	85.21 ± 1.89	84.71 ± 2.35	90.02 ± 2.26	**90.64 ± 1.58**
Wine	66.31 ± 2.66	70.04 ± 3.36	80.81 ± 4.92	67.53 ± 2.79	66.33 ± 2.68	90.68 ± 1.41	**93.27 ± 1.96**
Average	70.34	69.92	73.21	69.95	69.93	76.87	**78.45**
Mean rank	4.5714	5.2857	3.8571	4.8571	5.2857	3	**1.1429**

Table 7
Comparison of average accuracy (%, mean ± std) on UCI-datasets with label noise of 25%, the result of rank-1 is in bold face.

	Eucli	LMNN	ITML	DML-eig	RNCA	KISSME	Re-KISSME
Breast tissue	44.97 ± 2.77	50.20 ± 3.21	49.79 ± 4.02	47.15 ± 3.80	44.97 ± 2.77	50.80 ± 4.16	**54.44 ± 5.32**
Statlog (Heart)	59.00 ± 2.80	57.39 ± 2.27	61.19 ± 3.22	58.52 ± 2.94	59.06 ± 2.81	65.20 ± 2.44	**66.65 ± 3.24**
Iris	88.96 ± 2.59	87.15 ± 3.50	87.70 ± 2.97	87.61 ± 2.77	87.25 ± 3.40	88.08 ± 2.90	**89.23 ± 2.21**
Parkinsons	73.26 ± 3.60	72.64 ± 3.16	72.69 ± 3.14	72.92 ± 3.06	73.15 ± 3.75	71.97 ± 3.92	**74.69 ± 2.94**
Protein	60.22 ± 3.65	55.89 ± 4.74	59.10 ± 4.89	55.30 ± 3.80	59.11 ± 4.35	**64.27 ± 3.62**	64.15 ± 3.66
Seeds	83.02 ± 2.24	83.64 ± 2.48	83.79 ± 2.48	82.95 ± 2.27	82.52 ± 2.51	86.24 ± 3.04	**86.81 ± 2.43**
Wine	64.62 ± 3.31	67.35 ± 3.06	77.52 ± 3.42	65.34 ± 3.04	64.76 ± 3.41	87.18 ± 2.46	**88.75 ± 2.14**
Average	67.72	67.75	70.25	67.11	67.26	73.40	**74.96**
Mean rank	4.3571	5.2857	3.8571	5.4286	5.2143	2.7143	**1.1429**

Table 8
Comparison of average accuracy (%, mean ± std) on UCI-datasets with label noise of 30%, the result of rank-1 is in bold face.

	Eucli	LMNN	ITML	DML-eig	RNCA	KISSME	Re-KISSME
Breast tissue	43.35 ± 4.76	46.36 ± 4.31	47.53 ± 4.63	44.85 ± 5.83	43.35 ± 4.76	47.49 ± 5.25	**50.93 ± 4.88**
Statlog (Heart)	57.46 ± 2.69	56.54 ± 2.81	59.48 ± 3.27	58.06 ± 2.24	57.33 ± 2.76	61.56 ± 3.61	**62.31 ± 3.15**
Iris	81.68 ± 2.37	82.02 ± 3.35	80.84 ± 2.98	81.77 ± 2.35	82.12 ± 3.73	80.40 ± 2.81	**83.28 ± 2.32**
Parkinsons	70.64 ± 3.42	69.44 ± 3.65	70.08 ± 4.39	70.64 ± 4.06	70.49 ± 3.30	67.90 ± 3.35	**70.74 ± 2.92**
Protein	57.56 ± 2.65	52.95 ± 4.70	56.82 ± 3.28	51.55 ± 4.02	56.48 ± 3.81	**60.25 ± 5.38**	62.25 ± 3.46
Seeds	80.02 ± 2.36	80.21 ± 2.19	80.57 ± 2.43	79.90 ± 2.01	79.05 ± 2.84	82.07 ± 2.80	**82.83 ± 2.43**
Wine	61.18 ± 3.1	62.71 ± 3.34	67.58 ± 4.13	62.31 ± 2.90	61.18 ± 3.17	80.77 ± 3.66	**81.81 ± 2.34**
Average	64.56	64.32	66.13	64.16	64.29	68.63	**70.60**
Mean rank	4.7857	4.8571	3.7143	4.7857	5.2857	3.5714	**1**

Table 9
Mean Rank Values Differences Between Re-KISSME and Other Methods.

Noise level	Friedman test (p-value)	Bonferroni-Dunn test (rank difference)					
		Eucli	LMNN	ITML	DML-eig	RNCA	KISSME
0%	0.008	**4.4285**	**3.8571**	2.4285	**3.8571**	**4.4285**	1
5%	0.001	**4.5714**	**4.2857**	2.4286	**4.5714**	**4.1429**	1
10%	0.001	**3.2857**	**4.5714**	2.7143	**4.8571**	**4**	1.5714
15%	0.001	**4.0714**	**4.5714**	3	**3.7143**	**4.5**	1.1429
20%	0.002	**3.4285**	**4.1428**	2.7142	**3.7142**	**4.1428**	1.8571
25%	0.001	**3.2142**	**4.1428**	2.7142	**4.2857**	**4.0714**	1.5714
30%	0.004	**3.7857**	**3.8571**	2.7143	**3.7857**	**4.2857**	2.5714

the performance of the DML method. Here we focus on the distance metric, so we extract the LOMO [6] descriptors from each image as the original feature representation.

The iLIDS dataset [45] a widely-used benchmark, which has 476 images of 119 pedestrians. All images are resized to 128×48. The number of images for each individual varies from 2 to 8. Since this dataset was collected at an airport, the images often have severe occlusions caused by people and luggage. These images were captured by multiple cameras and there exist great challenges in same class. After adding the different percentages of label noise, the recognition of objective function will be more complex.

In our experiments, the images of ps persons are randomly selected to compose testing set, while the rest images compose training set. The ps are set as 59 in the iLIDS, which means that one half of persons are used for training, and the rest for testing. The percentage of label noise in training sets varies from 0% to 40%, where the percentage 0% means that the label is clear. This partition is repeated 10 times. The single-shot evaluation approach is adopted, one image is randomly selected as the gallery image and the rest are used as probe images for each person in the test set. This process is repeated 10 times. We evaluate the performance by calculating the average Cumulative Matching Characteristic (CMC) curves and reporting the proportion of uncertainty removed (PUR) scores. For CMC curves, $CMC(r) = \sum_{i=1}^{r} p(i)$ is the top r matches, $p(i)$ represents the probability of correct match at rank-i. For the PUR scores is computed as

$$PUR = \frac{log(N) + \sum_{i=1}^{N} p(i) log(p(i))}{log(N)} \tag{35}$$

The proposal is compared with ten popular metric learning methods, including DML-eig, RNCA, XQDA [6], SVMML [46], KISSME, PCCA [47], rPCCA [48], LFDA [49], kLFDA [48] and MFA [48,50]. The parameter tuning of respective methods refers to the original literature. To make the computation tractable, the feature dimension is reduced to 45 through first conducting PCA suggested as in [48]. The results on this dataset are reported as shown in Fig. 3: (1) These methods degenerate more or less with the increasing percentage of label noise, this validates that the label noise would harm the performance of DML in application. Especially for SVMML and KISSME, these two methods suffer from the serious degeneration of performance in the presence of label noise. (2) The CMC curve of our algorithm is over other methods when the label is clear. This validates that our proposal also improves the performance over KISSME in absence of label noise. (3) Benefited from the resampling scheme, our proposal Re-KISSME outperforms other methods at different level of label noise. This superiority of Re-KISSME is obvious in comparison with XQDA and KISSME. The performances of these three methods are the top 3 and similar in the Fig. 3 a, because that they shares a common assumption that the similar and dissimilar pairs come from two different Gaussian distributions. (4) Although as a method designed for label noise, RNCA is inferior to Re-KISSME. This may because the RNCA make uss of local structue, which is influenced by the label noise more than Re-KISSME.

To detail the results more clearly, the scores of the first 20 ranks and the PUR scores are shown from Tables 10–12. Our method achieves the best performance, with 51.8%, 50.3%, 48.8%, 47.7%, 46% rank-1 identification rates, 48.1%, 44.9%, 43.1%, 42.1%, 40.3% PUR score with label rate of 0%, 10%, 20%, 30%, 40% respectively. When the label is clear, Re-KISSME achieves an improvement of 8.3% at rank-1, and 7% at PUR score on XQDA. When the noise rate increases to 40%, the proposed method outperform KISSME and RNCA with an improvement of 44.8% and 15.5% at top-1, respectively. The results show that our performance is competitive and robust in the presence of label noise.

5.3. Experiments on camera pair 2–4 of RAiD for person re-identification

RAiD [51] dataset is collected recently consisting of two indoor (camera 1 and 2) and two outdoor (camera 3 and 4). The size of all images is 128×48. This new dataset consists of 43 subjects and 6920 images, which has large illumination variation. To make it tractable for compared methods, we choose the camera pairs 2–4 (indoor-outdoor) with 2655 images. PCA is first applied to reduce the dimension of LOMO descriptors to 100, because the images is larger than iLIDS. The recognition of subjects also becomes more challenging after adding label noise. In this experiment, we focus on analysis of degeneration with increasing label noise. So the percentage of label noise ranges from 5% to 40%.

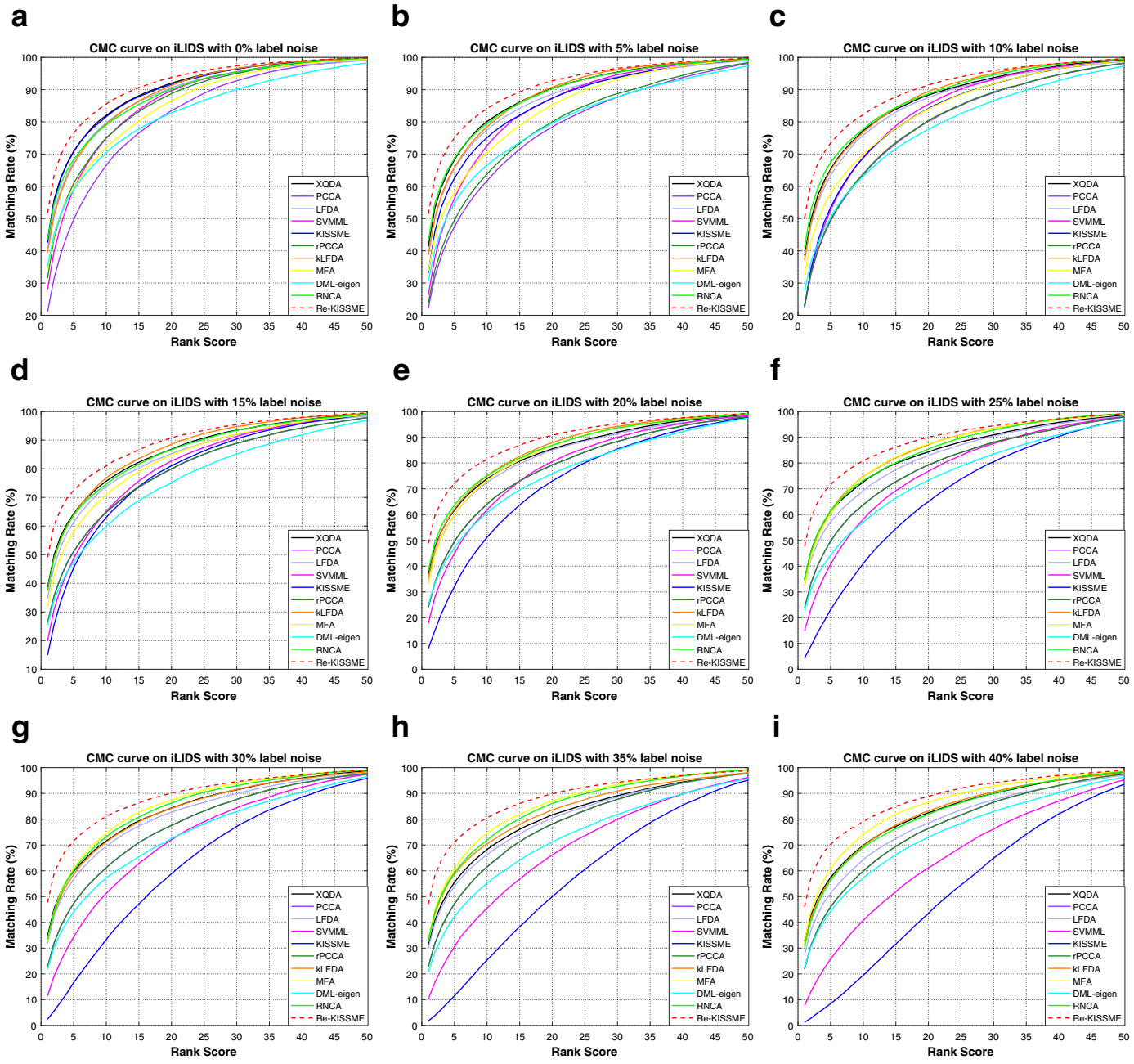


Fig. 3. CMC curves of each method on iLIDS.

Table 10
CMC at r=1, 5, 10, 20 and PUR scores (%) on iLIDS with 0%-10% label noise.

r	*Noise rate=0%*					*Noise rate=5%*					*Noise rate=10%*				
	1	5	10	20	PUR	1	5	10	20	PUR	1	5	10	20	PUR
XQDA	43.5	70.9	82.0	92.0	41.1	41.3	68.3	80.1	90.4	38.3	38.5	65.3	77.2	88.3	35.0
PCCA	21.1	49.8	66.5	83.5	22.4	22.3	47.5	61.7	78.4	19.5	22.9	49.3	63.6	80.1	20.7
LFDA	40.7	67.7	79.7	90.5	37.6	39.2	65.5	77.5	88.7	35.3	37.2	63.3	75.9	87.6	33.2
SVMML	28.1	59.2	74.9	89.7	29.8	26.3	56.3	72.6	87.3	27.6	22.9	52.7	69.0	85.5	24.3
MFA	34.3	59.4	72.0	86.5	30.8	33.7	58.0	70.4	85.3	29.6	32.6	57.8	69.9	84.0	28.3
rPCCA	31.6	61.0	75.1	88.8	31.3	23.7	49.5	63.8	80.0	21.2	23.1	49.7	63.9	80.5	21.0
kLFDA	39.6	66.9	79.4	91.3	37.6	38.7	65.9	78.4	90.8	36.5	37.1	64.9	76.8	89.4	34.8
DML-eig	35.2	58.9	70.5	82.8	29.1	30.4	54.8	66.6	79.6	24.6	27.6	50.3	63.2	77.7	21.6
RNCA	43.5	68.3	79.1	90.2	39.2	43.4	68.6	79.4	90.3	39.2	41.1	67.4	77.7	88.7	37.2
KISSME	42.5	70.8	81.6	91.7	40.6	33.1	62.5	75.1	87.4	31.7	22.5	53.4	68.8	84.2	23.9
Re-KISSME	**51.8**	**76.6**	**85.6**	**93.8**	**48.1**	**51.3**	**75.1**	**84.1**	**92.8**	**46.6**	**50.3**	**73.4**	**82.2**	**91.4**	**44.9**

Table 11
CMC at r=1, 5, 10, 20 and PUR scores (%) on iLIDS with 15%-25% label noise.

r	*Noise rate=15%*					*Noise rate=20%*					*Noise rate=25%*				
	1	5	10	20	PUR	1	5	10	20	PUR	1	5	10	20	PUR
XQDA	38.6	64.3	75.7	86.8	34.2	36.8	61.8	73.9	85.5	31.8	34.9	61.0	72.6	84.3	30.3
PCCA	26.3	51.0	64.8	79.9	22.1	24.0	49.5	64.0	79.2	20.9	23.6	49.6	63.7	79.2	20.6
LFDA	34.7	61.4	73.8	85.3	30.7	34.3	59.8	72.6	85.0	29.8	32.4	57.2	69.2	82.9	27.4
SVMML	20.0	48.7	65.2	82.8	21.2	17.8	45.0	61.9	80.5	18.7	14.9	40.8	58.1	76.8	15.6
MFA	32.1	57.9	71.0	84.7	28.5	33.0	59.6	73.3	86.7	30.2	32.7	60.5	74.5	87.4	30.6
rPCCA	26.4	51.2	64.9	80.0	22.3	24.1	49.7	64.1	79.2	21.0	23.7	49.7	63.8	79.3	20.7
kLFDA	37.4	63.8	76.8	88.5	34.3	35.4	62.2	75.0	87.9	32.4	34.3	61.3	74.8	87.1	31.4
DML-eig	25.4	47.4	60.0	75.1	19.3	25.1	47.2	60.7	75.8	19.3	22.8	44.2	57.1	73.3	16.9
RNCA	37.7	63.9	74.7	86.7	33.4	38.4	63.7	75.0	86.9	33.7	34.8	61.5	73.3	85.5	31.0
KISSME	15.0	45.5	63.1	81.1	18.6	8.0	32.1	51.2	73.0	11.4	4.3	23.1	41.0	65.0	7.1
Re-KISSME	**49.1**	**72.5**	**81.0**	**90.8**	**43.6**	**48.8**	**72.1**	**81.4**	**90.8**	**43.1**	**47.7**	**71.6**	**80.8**	**90.0**	**42.1**

Table 12
CMC at r=1, 5, 10, 20 and PUR scores (%) on iLIDS with 30%-40% label noise.

r	*Noise rate=30%*					*Noise rate=35%*					*Noise rate=40%*				
	1	5	10	20	PUR	1	5	10	20	PUR	1	5	10	20	PUR
XQDA	34.7	59.8	71.6	84.4	29.9	31.1	55.6	68.2	81.7	25.7	32.7	57.4	69.8	82.5	27.5
PCCA	22.8	47.3	61.2	77.6	19.3	22.9	47.1	61.5	78.1	19.4	21.8	45.9	59.9	76.4	18.2
LFDA	31.9	56.8	69.5	82.7	27.0	30.1	54.0	66.3	80.4	24.5	27.4	51.2	64.4	78.5	22.1
SVMML	11.5	34.3	51.3	72.0	11.7	10.2	30.5	45.5	66.1	8.8	7.7	25.8	40.8	61.1	6.2
MFA	32.8	60.9	74.5	87.4	31.0	33.1	60.8	74.6	87.0	30.9	32.9	60.9	74.1	86.6	30.4
rPCCA	22.8	47.4	61.2	77.6	19.4	22.8	47.2	61.5	78.1	19.5	21.9	46.0	59.9	76.4	18.2
kLFDA	33.6	58.6	71.2	84.2	28.8	33.4	58.7	70.5	83.6	28.3	31.4	56.6	69.6	83.2	26.9
DML-eig	21.9	44.1	57.2	72.6	16.4	20.8	42.2	55.2	71.2	15.0	22.3	44.3	57.2	73.0	16.6
RNCA	33.5	60.2	73.2	86.3	30.3	32.9	59.3	71.9	86.1	29.7	30.5	56.4	69.0	81.9	26.2
KISSME	2.4	16.7	33.4	58.9	5.1	1.8	11.5	25.4	50.0	3.0	1.2	8.5	19.4	43.5	2.0
Re-KISSME	**47.7**	**71.6**	**81.1**	**90.1**	**42.1**	**47.1**	**71.2**	**80.5**	**89.9**	**41.4**	**46.0**	**70.3**	**79.1**	**88.9**	**40.3**

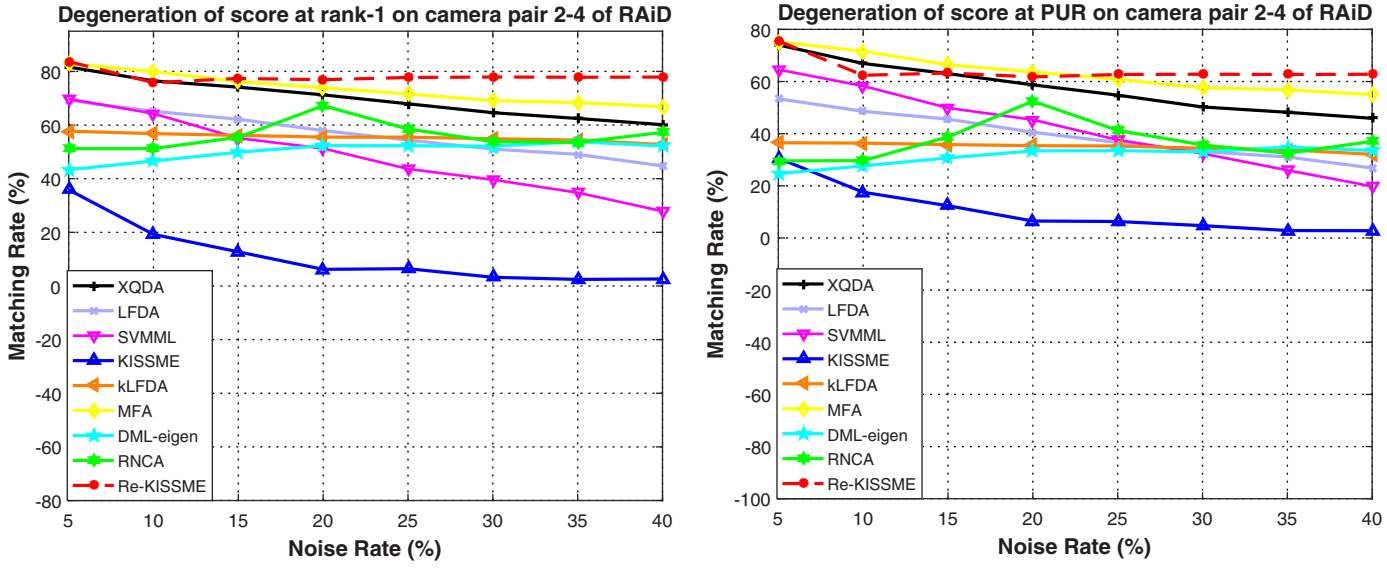


Fig. 4. The degeneration of each method on camera pair 2–4 of RAiD with 5%-40% label noise.

For this dataset, the ps is set as 21 in the RAiD, which also means that one half of persons are used for training, and the other half for testing. This partition is repeated 10 times. The single-shot evaluation approach is adopted, one image of each person is randomly selected to construct the gallery set and other images construct the probe set in the test set. This process is repeated 10 times. Figure 4 plots the average accuracies of rank-1 and PUR scores of different noise levels for each methods, and details are shown in Table 13 and Table 14.

Considering the high time cost and discarding PCCA and rPCCA, Re-KISSME is evaluated by comparison with nine popular metric learning methods, XQDA, SVMML, KISSME, DML-eig, RNCA, LFDA, kLFDA, MFA, ITML. The parameter setting of comparable methods is as mentioned before. The observations on this dataset are follows: (1) It is shown that the Re-KISSME outperforms other methods at rank-1 and PUR score in most cases from Tables 13 and 14. The performance of Re-KISSME defeats other methods at rank-1 when label noise rate is higher than 15%, although the accuracy of Re-KISSME at rank-1 score is little smaller than MFA and XQDA with 10% noise. Similar results are also found for the PUR. (2) The scores of Re-KISSME are about 77% at rank-1 and 62% at PUR score when noise rate is higher than 15%, there is little difference.

Table 13
CMC at r=1 and PUR scores (%) on camera pair 2–4 of RAiD with 5%-20% label noise.

r	*Noise rate=5%*		*Noise rate=10%*		*Noise rate=15%*		*Noise rate=20%*	
	1	PUR	1	PUR	1	PUR	1	PUR
XQDA	81.7	74.0	76.5	66.9	74.1	63.0	71.3	58.7
LFDA	69.2	53.4	65.1	48.6	62.2	45.6	57.9	40.5
SVMML	69.7	64.6	64.3	58.3	55.1	49.9	51.3	45.3
MFA	82.9	75.2	**80.0**	**71.6**	76.2	**66.4**	73.9	**63.7**
kLFDA	57.7	36.5	56.8	36.4	56.2	35.8	55.5	35.4
ITML	70.5	55.9	68.7	53.8	62.1	46.2	62.3	45.3
DML-eig	43.3	24.7	46.6	27.6	49.9	30.7	52.3	33.4
RNCA	51.2	29.6	51.2	29.7	55.4	38.8	67.2	52.5
KISSME	35.9	30.6	19.2	17.5	12.7	12.4	6.2	6.5
Re-KISSME	**83.7**	**75.6**	75.8	62.4	**77.4**	63.3	**76.9**	61.8

Table 14
CMC at r=1 and PUR scores (%) on cameras pair 2–4 of RAiD with 25%-40% label noise.

r	*Noise rate=25%*		*Noise rate=30%*		*Noise rate=35%*		*Noise rate=40%*	
	1	PUR	1	PUR	1	PUR	1	PUR
XQDA	67.9	54.8	64.6	50.2	62.5	48.2	60.1	45.9
LFDA	54.4	36.6	51.0	33.2	49.1	31.0	44.8	26.8
SVMML	43.7	37.6	39.6	32.4	34.8	26.0	27.9	19.8
MFA	71.6	60.9	69.1	57.6	68.4	56.8	66.9	55.0
kLFDA	55.5	35.3	54.9	34.4	54.4	33.8	52.8	32.1
ITML	61.2	45.0	63.3	46.8	63.1	48.9	60.3	43.8
DML-eig	52.3	33.5	52.1	33.0	53.8	34.9	52.2	33.6
RNCA	58.5	41.3	53.9	35.7	53.5	32.5	57.3	37.0
KISSME	6.4	6.3	3.3	4.7	2.4	2.8	2.6	2.8
Re-KISSME	**77.7**	**62.7**	**77.9**	**62.8**	**77.8**	**62.7**	**77.9**	**62.8**

Table 15
Training time (seconds) of each method on UCI-dataset iris with 5% label noise.

	Eucli	LMNN	ITML	DML-eig	RNCA	KISSME	Re-KISSME
Time	3.40E-06	0.9014258	1.9182508	0.0942964	0.9727386	0.000559	0.0261398
Solving method	Non-iterative	Iterative	Iterative	Iterative	Iterative	Non-iterative	Iterative

Table 16
Training time (seconds) of each method on cameras pair 2–4 of RAiD with 5% label noise.

	XQDA	LFDA	SVMML	KISSME	kLFDA	ITML	DML-eig	RNCA	MFA	Re-KISSME
Time	0.0158	229.2761	227.7476	0.0030	23.1695	564.3897	5.5920	911.9961	87.0308	13.0250
Solving method	Non-iterative	Iterative	Iterative	Non-iterative	Iterative	Iterative	Iterative	Iterative	Iterative	Iterative

However, the score of KISSME degenerates to 2.6% at rank-1 at the same time. When the noise rate increases to 40%, our proposal achieves an improvement of 20.6% on RNCA.

5.4. Computational cost

Finally we compare the computational cost of the proposed method, so the training time of each method is recorded. We choose the experiments on UCI data set iris and person re-identification dataset RAiD2-4 at noise level of 5% as analysis examples. The averaging time of one trial in each case is recorded, and the results are shown in Tables 15 and 16 for two datasets, respectively.

From Table 15, the computational cost of KISSME and Re-KISSME are less than LMNN, ITML, DML-eig and RNCA. As these methods on general classification are iterative methods except KISSME, it is natural that Re-KISSME is of a little higher time cost than KISSME. From Table 16, XQDA, KISSME, DML-eig and Re-KISSME are faster than LFDA, SVMML, kLFDA, MFA, ITML and RNCA. From these two tables, the proposal is of low computational cost compared to most of comparable DML methods.

6. Conclusion

To address the problem of label noise for the DML with pairwise constraints, this paper proposes a method Re-KISSME. This method reasons the true similarity of each pair and resample each pair to optimize the metric matrix. First, the resampling scheme is based on two factors: (1) the structure of sample pairs; (2) the priors of label. Second, the covariance matrices are iteratively computed according to the resampling scheme. Introducing the true constraint as a latent variable, a maximum likelihood estimation model is constructed to solve the parameters. As a result, Re-KISSME can learn the underlying distribution in the presence of label noise.

We conduct experiments on UCI datasets and two person re-identification datasets with synthetic label noise. First, with different level of noise on these three datasets, the result of performance and Friedman test validate that Re-KISSME outperforms other methods on seven UCI datasets. Second the experiments show that our proposal improves KISSME and reduces the negative influence of label noise on two person re-identification datasets. Finally, further work should take DML methods based on triplet

constraints into account as Re-KISSME only considers the pairwise constraints.

Acknowledgments

The authors would like to thank the anonymous reviewers for their critical comments and constructive suggestions. The authors would like to express their gratitude to their colleagues for their earnest help. This work was supported by the National Key R&D Program of China (No. 2017YFC0803700), the National Natural Science Foundation of China (U1636220, 61602482 and 61501463), and the Beijing Natural Science Foundation (Grant no. 4172063).

References

[1] B. Mcfee, G. Lanckriet, Metric learning to rank, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 775–782.
[2] S.C.H. Hoi, W. Liu, M.R. Lyu, W.-Y. Ma, Learning distance metric with contextual constraints for image retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 2072–2078.
[3] M. Guillaumin, J. Verbeek, C. Schmid, Is that you? Metric learning approaches for face identification, in: Proceedings of the IEEE International Conference on Computer Vision, 2009, pp. 498–505.
[4] M. Guillaumin, T. Mensink, J. Verbeek, C. Schmid, TagProp: discriminative metric learning in nearest neighbor models for image auto-annotation, in: Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, 2009, pp. 309–316.
[5] X. Wang, G. Hua, T.X. Han, Discriminative tracking by metric learning, in: Proceedings of the European Conference on Computer Vision, 2010, pp. 200–214.
[6] S. Liao, Y. Hu, X. Zhu, S.Z. Li, Person re-identification by local maximal occurrence representation and metric learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 07-12-June, 2014, pp. 2197–2206.
[7] W.S. Zheng, S. Gong, T. Xiang, Reidentification by relative distance comparison, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 649–656.
[8] A. Bellet, A. Habrard, M. Sebban, A Survey on Metric Learning for Feature Vectors and Structured Data, arXiv preprint arXiv:1306.6709. 2013.
[9] B. Kulis, Metric learning: a survey, Found. Trends® Mach. Learn. 5 (4) (2013) 287–364.
[10] A.L.B. Miranda, L.P.F. Garcia, A.C.P.L.F. Carvalho, A.C. Lorena, Use of classification algorithms in noise detection and elimination, in: Proceedings of the International Conference on Hybrid Artificial Intelligence Systems, 2009, pp. 417–424.
[11] C.E. Brodley, M.A. Friedl, Identifying mislabeled training data, J. Artif. Intell. Res. 11 (1999) 131–167.
[12] D. Wang, X. Tan, Robust distance metric learning in the presence of label noise, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014, pp. 1321–1327.
[13] T. Yang, R. Jin, A.A.K. Jain, Learning from noisy side information by generalized maximum entropy model, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 1199–1206.
[14] M. Kostinger, M. Hirzer, P. Wohlhart, P.M. Roth, H. Bischof, Large scale metric learning from equivalence constraints, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2288–2295.
[15] V.C.R. Raykar, S. Yu, L.H. Zhao, G.H. Valadez, C. Florin, L. Bogoni, L. Moy, Learning from crowds, J. Mach. Learn. Res. 11 (2010) 1297–1322.
[16] E.P. Xing, A.Y. Ng, M.I. Jordan, S. Russell, Distance metric learning with application to clustering with side-information, in: Proceedings of the Advances in neural information processing systems, 2003, pp. 521–528.
[17] J. Goldberger, S. Roweis, G. Hinton, R. Salakhutdinov, Neighbourhood Components Analysis Jacob, in: Proceedings of the Advances in neural information processing systems, 2005, pp. 1891–1901.
[18] K.K.Q. Weinberger, L.K.L. Saul, Distance metric learning for large margin nearest neighbor classification, J. Mach. Learn. 10 (2009) 207–244.
[19] J.V. Davis, B. Kulis, P. Jain, S. Sra, I.S. Dhillon, Information-theoretic metric learning, in: Proceedings of the 24th International Conference on Machine Learning - ICML '07, 2007, pp. 209–216.
[20] J. Wang, A. Woznica, A. Kalousis, Parametric local metric learning for nearest neighbor classification, in: Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 1601–1609.
[21] D. Kedem, S. Tyree, K.Q. Weinberger, F. Sha, G. Lanckriet, Non-linear metric learning, in: Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 2573–2581.
[22] H.-J. Ye, D.-C. Zhan, X.-M. Si, Y. Jiang, Z.-H. Zhou, What makes objects similar : a unified multi-metric learning approach, in: Proceedings of the Advances in Neural Information Processing Systems, 2016, pp. 1235–1243.
[23] A. Bar-Hillel, T. Hertz, N. Shental, D. Weinshall, Learning distance functions using equivalence relations, in: Proceedings of the Twentieth International Conference on Machine Learning ICML-2003, 2003, pp. 11–18.
[24] S. Xiang, F. Nie, C. Zhang, Learning a Mahalanobis distance metric for data clustering and classification, Pattern Recognit. 41 (12) (2008) 3600–3612.
[25] J. Yi, R. Jin, A.K. Jain, S. Jain, T. Yang, Semi-crowdsourced clustering: generalizing crowd labeling by robust distance metric learning, in: Proceedings of the Advances in neural information processing systems, 2012, pp. 1772–1780.
[26] B. Frénay, M. Verleysen, Classification in the presence of label noise: a survey, IEEE Trans. Neural Netw. Learn. Syst. 25 (5) (2014) 845–869.
[27] I. Guyon, N. Matic, V. Vapnik, Discovering informative patterns and data cleaning, in: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, 1996, pp. 181–203.
[28] D. Gamberger, N. Lavrac, S. Dzeroski, Noise detection and elimination in data preprocessing: experiments in medical domains, Appl. Artif. Intell. 14 (2) (2000) 205–223.
[29] F. Muhlenbach, S. Lallich, D.A. Zighed, Identifying and handling mislabelled instances, J. Intell. Inf. Syst. 22 (1) (2004) 89–109.
[30] N. Natarajan, I.S. Dhillon, P. Ravikumar, A. Tewari, Learning with noisy labels, in: Proceedings of the Advances in Neural Information Processing Systems, 2013, pp. 1196–1204.
[31] T. Liu, D. Tao, Classification with noisy labels by importance reweighting, IEEE Trans. Pattern Anal. Mach. Intell. 38 (3) (2016) 447–461.
[32] Z.H. Zhou, W. Gao, L. Wang, Y.F. Li, Risk minimization in the presence of label noise, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 1575–1581.
[33] A. Ghosh, H. Kumar, P.S. Sastry, Robust loss functions under label noise for deep neural networks, in: Proceedings of the In Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 1919–1925.
[34] N.D. Lawrence, B. Scholkopf, Estimating a Kernel–Fisher discrimant in the presence of label noise, in: Proceedings of the 18th International Conference on Machine Learning, 1, 2001, pp. 306–313.
[35] Y. Li, L.F. Wessels, D. de Ridder, M.J. Reinders, Classification in the presence of class noise using a probabilistic Kernel–Fisher method, Pattern Recognit. 40 (12) (2007) 3349–3357.
[36] J. Bootkrajang, A. Kabán, Label-noise robust logistic regression and its applications, in: Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2012, pp. 143–158.
[37] J. Bootkrajang, A. Kabán, Boosting in the presence of label noise, in: Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI 2013), 71, 2013, pp. 82–90.
[38] D. Tao, L. Jin, Y. Wang, X. Li, Person reidentification by minimum classification error-based KISS metric learning, IEEE Trans. Cybern. 45 (2) (2015) 242–252.
[39] D. Tao, Y. Guo, M. Song, Y. Li, Z. Yu, Y.Y. Tang, Person re-identification by dual-regularized, IEEE Trans. Image Process. 25 (6) (2016) 2726–2738.
[40] W. Zhang, R. Rekaya, K. Bertrand, A method for predicting disease subtypes in presence of misclassification among training samples using gene expression: application to human breast cancer, Bioinformatics 22 (3) (2005) 317–325.
[41] M. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, J. Res. Natl. Bur. Stand. 49 (1) (1952) 409–436.
[42] D. Dheeru, E. Karra Taniskidou, UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences, 2017. Available: http://archive.ics.uci.edu/ml.
[43] M.A. Little, P.E. McSharry, S.J. Roberts, D.A. Costello, I.M. Moroz, Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection, Biomed. Eng. Online 6 (1) (2007) 23.
[44] Y. Ying, Distance metric learning with eigenvalue optimization, J. Mach. Learn. Res. 13 (2012) 1–26.
[45] W.-S. Zheng, S. Gong, T. Xiang, Associating groups of people, in: Proceedings of the British Machine Vision Conference, 2, 2009, pp. 91–110.
[46] Z. Li, S. Chang, F. Liang, T.S. Huang, L. Cao, J.R. Smith, Learning locally-adaptive decision functions for person verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3610–3617.
[47] A. Mignon, A. Mignon, A. Mignon, PCCA : a new approach for distance learning from sparse pairwise constraints, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2666–2672.
[48] F. Xiong, M. Gou, O. Camps, M. Sznaier, Person re-identification using kernel-based metric learning methods, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 1–16.
[49] S. Pedagadi, J. Orwell, S. Velastin, B. Boghossian, Local fisher discriminant analysis for pedestrian re-identification, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2013) 3318–3325.
[50] S. Yan, D. Xu, B. Zhang, H.J. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: a general framework for dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell. 29 (1) (2007) 40–51.
[51] A. Das, A. Chakraborty, A.K. Roy-Chowdhury, Consistent re-identification in a camera network, in: Proceedings of the European Conference on Computer Vision, 8690, 2014, pp. 330–345. LNCS

Fanxia Zeng received her B.Sc. and M.Sc. degrees in Applied Mathematics from the Chengdu University of Technology, in 2010 and 2013, respectively. She is currently a Ph.D. candidate at the Institute of Automation, Chinese Academy of Sciences (CAS). Her research interests include machine learning, pattern recognition, distance metric learning.

Wensheng Zhang Wensheng Zhang received a Ph.D. degree in Pattern Recognition and Intelligent Systems from the Institute of Automation, Chinese Academy of Sciences (CAS), in 2000. He joined the Institute of Software, CAS, in 2001. He is a Professor of Machine Learning and Data Mining and the Director of Research and Development Department, Institute of Automation, CAS. His research interests include computer vision, pattern recognition and artificial intelligence.

Siheng Zhang is currently a Ph.D. candidate at the Institute of Automation, Chinese Academy of Sciences (CAS). He received a B.S degree in Department of Automation, Tsinghua University, Beijing, China, in 2015. His research interests include deep learning and knowledge graph.

Nan Zheng is an associate professor at the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. She received the Ph.D. degree from Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2012. Her research interests include data mining, information retrieval and machine learning.