Beyond IID: Learning to Combine Non-IID Metrics for Vision Tasks

Yinghuan Shi,† Wenbin Li,† Yang Gao,‡ Longbing Cao,‡ Dinggang Shen§
†State Key Laboratory for Novel Software Technology, Nanjing University, China
‡Advanced Analytics Institute, University of Technology at Sydney, Australia
§Department of Radiology and BRIC, UNC-Chapel Hill, USA

Abstract

Metric learning has been widely employed, especially in various computer vision tasks, with the fundamental assumption that all samples (e.g., regions/superpixels in images/videos) are independent and identically distributed (IID). However, since the samples are usually spatially-connected or temporally-correlated with their physically-connected neighbours, they are not IID (non-IID for short), which cannot be directly handled by existing methods. Thus, we propose to learn and integrate non-IID metrics (NIME). To incorporate the non-IID spatial/temporal relations, instead of directly using non-IID features and metric learning as previous methods, NIME first builds several non-IID representations on original (non-IID) features by various graph kernel functions, and then automatically learns the metric under the best combination of various non-IID representations. NIME is applied to solve two typical computer vision tasks: interactive image segmentation and histology image identification. The results show that learning and integrating non-IID metrics improves the performance, compared to the IID methods. Moreover, our method achieves results comparable or better than that of the state-of-the-arts.

Introduction

In recent years, metric learning, aiming to transform the samples from their original feature space to a more informative and discriminative one, has been widely adopted in various computer vision tasks, e.g., face recognition (Wang et al. 2014; Guillaumin, Verbeek, and Schmid 2009; Huang et al. 2015), image classification and retrieval (Wang and Tan 2014; Gao et al. 2014; Mensink et al. 2012), object recognition and segmentation (Lajugie, Arlot, and Bach 2014; Teney et al. 2015; Verma et al. 2012), person re-identification (Liao and Li 2015), and visual object tracking (Li et al. 2012; Jiang, Liu, and Wu 2012).

The main goal of metric learning is to learn the optimal metric (i.e., positive semi-definite matrix $M = L^\top L$) by minimizing the pair-wise Mahalanobis distance of training samples. It is assumed that, by applying metric $M$ (or $L$) for feature space transformation, the performance of subsequent classification or clustering can be improved, compared with only using the original features.

Most of existing metric learning methods developed for computer vision tasks take the independent and identically distributed (IID) assumption, i.e., regarding all samples (e.g., regions/superpixels in images/videos) as IID. Consequently, the obtained metric $M$ (or $L$) learned under the IID assumption ignores the relations between different samples. This essentially follows the general metric learning practices, i.e., converting a specific computer vision task to a general metric learning problem with the IID assumption, although sometimes in previous methods the non-IID features are first utilized to capture spatial/temporal information (Teney et al. 2015).

The IID metric learning methods are not consistent with the underlying real-life problem nature, i.e., different samples are often spatially-connected or temporally-correlated with their neighbours in certain ways. Accordingly, existing IID methods cannot handle the non-IID complexities, as illustrated in Figure 1, and it is essential for us to develop new and effective metric learning methods to learn the non-IID samples.

In this work, we propose to learn and integrate non-IID metrics (NIME), to automatically capture the underlying non-IID relations for different tasks. To prevent the ad-hoc setting, NIME first builds non-IID representations by modeling each sample along with its corresponding neighbors with three simple and effective graph kernels (i.e., direct product, Hausdorff distance, and max pooling) on the original feature space, in which the respective feature vectors of different samples are regarded as nodes in the graph. With the obtained non-IID representations, we further regard the subsequent non-IID metric learning as a kernel-based learning problem, the optimal metric is learned by minimizing the corresponding empirical errors w.r.t. different non-IID representations. The stochastic gradient descent (SGD) (Bottou 2010) is performed to accelerate the optimization process. Also, previous methods normally design the ad-hoc setting for specific tasks, to fully make the various non-IID representations adapt to different tasks, the best linear combination of multiple non-IID representations (named NIME-CK) is automatically learned. Please note that, instead of introducing additional parameters (e.g., imposing other regularizer) or assumption (e.g., projecting onto a unique subspace) which might cause extra parameter tuning or heavy computational burden, NIME-CK only requires to tune one param-
eter, thus keeping the same property as a single non-IID representation, which can be solved by stochastic optimization. Finally, the learned metric transforms the samples from their original feature space to another space. In the transformed feature space, many commonly-used classifiers can be directly applied to predict new samples. 1

Figure 1: The illustration of comparing IID v.s. non-IID metric learning in two different applications.

In summary, the major highlights of our work include:

- Our method is not (non-IID) features + metric learning as previous methods, but three phases as (non-IID) features + various non-IID representations + joint metric learning.
- Our method can be easily adapted to different applications which are not ad-hoc, with the best combination of various non-IID representations automatically learned.
- Our method is easy to implement, and has a nice property that can be solved by stochastic optimization.
- Many features (e.g., deep features), non-IID representations (e.g., Hausdorff), and metric-learning methods (e.g., LMNN) can be easily integrated into our method.

1Note that, although the existing IID metric learning methods, e.g., LMNN (Weinberger, Blitzer, and Saul 2005) and NCA (Goldberger et al. 2004)), also impose the constraint on k-neighborhood samples, their data assumption and learning goal are fundamentally different from ours. (1) They assume that the distance, measured in Mahalanobis space, between neighboring samples with the same label should be small. Instead, we assume that the sample with its physically-connected (i.e., spatially and temporally) neighbors (with either the same, different or even unknown labels) are not IID, which makes the corresponding comparison of different samples becoming a similarity measure on different undirected graphs. (2) Their goal is to learn the metric on a specifically-designed function, while ours is to learn the intrinsic non-IID representation and then the baseline metric learning methods, including LMCA (Torresani and Lee 2007), LMNN and NCA (Jain, Kulis, and Dhillon 2010), can be applied. In this way, our work supports to transfer existing IID metric learning methods for non-IID problems.

NIME is evaluated for handling two typical computer vision tasks to validate the effect of considering non-IID relations compared with pure IID assumption, which shows its great potential.

Related Work

Metric learning learns a linear, nonlinear, or a local feature transformation to achieve better feature representation. Depending on whether labels (or side information) are used, metric learning methods can be roughly classified into three categories: supervised, semi-supervised, and weakly (un-)supervised methods. As our proposed NIME method is for supervised metric learning, we review the relevant works accordingly. Xing et al. proposed MMC (Xing et al. 2002) to maximize the sum of pairwise distance for similar samples and add a constraint to keep large distance for dissimilar samples. Goldberger et al. reported NCA (Goldberger et al. 2004) to optimize the leave-one-out error. Weinberger et al. presented LMNN (Weinberger, Blitzer, and Saul 2005) requiring that (1) the k-nearest neighbors with the same label to be close to each other, and (2) samples of different labels be separated by imposing a large margin. Other supervised metric learning methods include MCML (Globerson and Roweis 2005), RCA (Shental et al. 2002), ITML (Davis et al. 2007). Detailed comparisons of existing methods can be referred to (Bellet, Habrard, and Sebban 2013).

Increasing recent efforts have been made to metric learning for computer vision. For example, for face identification and recognition, two methods, LDML and MKNN, were reported in (Guillaumin, Verbeek, and Schmid 2009) by learning the metric using the image-pair information. For person re-identification, a novel method was designed in (Liao and Li 2015) for cross-view metric learning problem; a set of hierarchical metrics were learned in (Verma et al. 2012) to reflect the underlying class taxonomy for multi-class classification. In addition, TagProp (Guillaumin et al. 2009) was introduced for image auto-annotation, which maximizes the log-likelihood of tag predictions in the training samples by combining a set of metrics trained with different features, e.g., local shape descriptors and global color histograms. In (Mensink et al. 2012), metric learning was incorporated into two classifiers, kNN and Nearest Class Mean (NCM), for large-scale image annotation. For dynamic scene segmentation, a metric learning framework was developed in (Teney et al. 2015) to jointly optimize the representation from various perspectives. For visual object tracking, both (Li et al. 2012) and (Jiang, Liu, and Wu 2012) presented respective metric learning methods.

Learning from non-IID data is a recent topic (Cao 2014)(Cao 2015) to address the intrinsic data complexities, with preliminary work reported such as for clustering (Wang and Cao 2011), group behavior analysis (Cao, Ou, and Yu 2012), and multi-instance learning (Zhou, Sun, and Li 2009). However, the non-IID metric learning is seldom exploited, especially for computer vision tasks.
Non-IID Representation via Graph

To incorporate the non-IID relations, we represent each sample with its corresponding physically-connected neighbors from spatial/temporal perspectives as an undirected graph, where different nodes denote different samples. In the constructed graph, the central node denotes the current sample under comparison, and the surrounding nodes denote the neighboring samples (see Figure 1). Note our concept of neighbours are different from the previous, here neighbours refer to the physically-connected samples.

Formally, the \( i \)-th training sample \((i = 1, 2, ..., N_t)\) is encoded as \( x_i \in \mathbb{R}^d \), where \( d \) is the feature dimensionality in the original feature space, and \( N_t \) is the number of training samples. For \( x_i \), its neighbors are denoted as \( x_{i,1}, x_{i,2}, ..., x_{i,m_i} \in \mathbb{R}^d \), where \( m_i \) is the total number of neighbors of training sample \( x_i \). Similarly, we denote the \( i \)-th testing sample \((i = 1, 2, ..., N_t)\) as \( z_i \in \mathbb{R}^d \), where \( N_t \) is the number of testing samples. For \( z_i \), its neighbors are denoted as \( z_{i,1}, z_{i,2}, ..., z_{i,n_i} \in \mathbb{R}^d \), where \( n_i \) is the total number of neighbors of testing sample \( z_i \).

By connecting each sample with its neighbors as an undirected graph, the classic Mahalanobis distance-based pairwise metric function can be regarded as the pairwise similarity in the defined graph kernel representations. Hence, learning metric \( M \) (or \( L \)) in the Mahalanobis distance space can be formulated as learning metric in the corresponding graph kernel space, such as in (Junior, Kulis, and Dhillon 2010)(Torresani and Lee 2007)(Wang et al. 2011).

Although previous work also introduced similar kernels (Lyu 2010)(Woznica, Kalousis, and Hilario 2006)(Kondor and Jebara 2003), they usually treated all the nodes equally. Accordingly, we define the graph kernel function by representing and measuring the constructed undirected graph for different samples. To fully represent the non-IID relations between each sample and its neighbors, we define the following three specific non-IID graph kernel functions, direct product kernel \( K_{DP} \), Hausdorff distance kernel \( K_{HD} \), and max pooling kernel \( K_{MP} \), obtained on all the training and testing samples.

The simplest method to compute the similarity of two undirected graphs is first to calculate the distance between two central nodes (i.e., two samples), and then calculate the averaged distance for pairwise combinations of all surrounding nodes (i.e., neighboring samples), which contributes to the direct product (DP) kernel \( K_{DP} \). In some cases, it is natural to incorporate the overall distributions of these two sets of surrounding nodes into calculating distances. Therefore, the Hausdorff distance is a good choice to compute the overall distance between two discrete sets, resulting in the Hausdorff distance (HD) kernel \( K_{HD} \). In addition, the max pooling function, widely used in computer vision tasks, also shows success for finding the near optimal matching between two discrete sets, contributing to the max pooling (MP) kernel \( K_{MP} \).

**Definition 1** Given two samples \( x_i \) and \( x_j \), their respective \( m_i \) and \( m_j \) neighbors are denoted as \( x_{i,p} \ (p = 1, ..., m_i) \) and \( x_{j,q} \ (q = 1, ..., m_j) \) respectively, and their DP \( K_{DP}(i,j) \), HD \( K_{HD}(i,j) \), and MP \( K_{MP}(i,j) \) distances are defined below:

\[
K_{DP}(i,j) = f(x_i, x_j) + \frac{1}{m_i \cdot m_j} \sum_{p=1}^{m_i} \sum_{q=1}^{m_j} f(x_{i,p}, x_{j,q}).
\]

\[
K_{HD}(i,j) = f(x_i, x_j) + \frac{1}{m_i \cdot m_j} h(X_i, X_j).
\]

\[
K_{MP}(i,j) = f(x_i, x_j) + \frac{1}{m_i} \max_{p=1}^{m_i} f(x_{i,p}, x_{j,q}) + \frac{1}{m_j} \max_{q=1}^{m_j} f(x_{i,p}, x_{j,q}).
\]

where \( f \) is a positive semi-definite kernel. In Eqn.(2), \( X_i = \{ x_{i,1}, x_{i,2}, ..., x_{i,m_i} \} \) is the set including all the neighbours of \( x_i \), and \( h \) is the HD kernel function.

Note that there are many alternatives to define \( f \). In this paper, we choose the Gaussian RBF kernel as \( f(x_i, x_j) = \exp(-\alpha_1 \| x_i - x_j \|^2) \) for simplicity, where \( \alpha_1 \) is the parameter. In Eqn.(2), the second term calculates the HD between two sets \( X_i \) and \( X_j \). For two sets \( A \) and \( B \), HD kernel function \( h \) is defined below.

\[
h(A, B) = \exp \left( -\alpha_2 \cdot \max \left( H(A, B), H(B, A) \right) \right),
\]

where \( \alpha_2 \) is the parameter, and \( H(A, B) = \max \min_{a \in A, b \in B} \| a - b \|, \ H(B, A) = \max \min_{b \in B, a \in A} \| b - a \| \).

Eqns.(1), (2) and (3) are easy to implement and fast to compute. All the graph kernels \( K_{DP}, K_{HD} \) and \( K_{MP} \) are computed on all the training samples \((x_1, x_2, ..., x_{N_t})\) and testing samples \((z_1, z_2, ..., z_{N_t})\), hence, \( K_{DP}, K_{HD}, K_{MP} \in \mathbb{R}^{(N_t+N_t) \times (N_t+N_t)} \). Also, all the graph kernels should be processed by using kernel alignment, i.e., for a kernel function \( K \), rescaling each \( K(i,j) = \frac{K(i,j)}{\sqrt{K(i,i) \cdot K(j,j)}} \).

For better representation, without loss of generality we put all the training samples before the testing samples. Taking \( K_{DP} \) as an example, we splat the graph kernel matrix as follows:

\[
K_{DP} = \begin{pmatrix}
K_{DP}^{tr} & G_{DP}^	op \\
G_{DP} & K_{DP}^{te}
\end{pmatrix},
\]

where \( K_{DP}^{tr} \in \mathbb{R}^{N_t \times N_t} \) and \( K_{DP}^{te} \in \mathbb{R}^{N_t \times N_t} \) are the parts of graph kernel matrix calculated on the training samples and testing samples, respectively. \( G_{DP} \in \mathbb{R}^{N_t \times N_t} \) denotes the kernel similarity matrix between training and testing samples.

**Learning to Combine Metrics**

To learn the metric (transformation) on the obtained non-IID representations, the original representation is equivalently transformed into the kernel-based representation according to the following Lemma (Torresani and Lee 2007).
Lemma 1 Given \( L \) (a learned metric in original feature space), and a nonlinear mapping \( \Phi \), the projection of feature vector transformation is equivalent to

\[
L\Phi = \Omega K,
\]

where \( \Omega \) is the matrix that represents \( L \) as a linear combination of the feature vectors, and \( K \) is a corresponding kernel matrix corresponding to the nonlinear mapping \( \Phi \).

Accordingly, in the kernel space, the original optimization of solving the metric \( L \) in original feature space (using the nonlinear mapping \( \Phi \)), is transformed to an equivalent form by solving \( \Omega \) (using the kernel matrix \( K \)). With the obtained non-IID representations, to learn the optimal metric on all the training samples, \( K \) can be \( K_{\text{DP}}^\nu, K_{\text{HD}}^\nu, K_{\text{MP}}^\nu \), or any positive semi-definite combination of them.

We now discuss how to learn metric \( \Omega \). First, we introduce the method to learn the metric by using the single non-IID representation. For simplicity, NIME-DP, NIME-HD, and NIME-MP denote the respective single non-IID representation-based metric learning methods. Then, the NIME-CK is introduced as a simple yet effective method to combine multiple non-IID representations.

### Learning with Single Non-IID Representation

Inspired by (Torresani and Lee 2007), for a given \( K \) (here \( K \) can be \( K_{\text{DP}}^\nu, K_{\text{HD}}^\nu \) or \( K_{\text{MP}}^\nu \)) calculated on all the training samples with known labels, the task of learning metric \( \Omega \) with single non-IID representation is to minimize the following objective function:

\[
E(\Omega; K) = \sum_{i,j} \eta_{ij} \|\Omega(k_i - k_j)\|^2 + \lambda \sum_{i,j,l} \eta_{ij} (1 - y_{il}) \bigg(\|\Omega(k_i - k_j)\|^2 + 1\bigg).
\]

where \( \eta_{ij} \in \{0, 1\} \) indicates that if \( j \)-th training sample is one of the \( k \)-closest neighbours of \( i \)-th training sample with the same label \( y_i \). \( y_{il} \in \{0, 1\} \) indicates whether \( i \)-th training sample and \( l \)-th training sample are with the same label. \( h(z) = \max(z, 0) \) is the hinge function. \( \lambda \) is the weight parameter of the second term. \( k_i \) is the \( i \)-th column of \( K \).

To obtain \( \Omega \), the gradient descent method is applied, with the derivative of \( E(\Omega; K) \) calculated as follows:

\[
\frac{\partial E(\Omega; K)}{\partial \Omega} = \left(2\sum_{i,j} \eta_{ij} \left[ E^i_j(k_i - k_j) - E^j_i(k_i - k_j) \right] \right) \Phi + 2\lambda \sum_{i,j,l} \eta_{ij} (1 - y_{il}) h'(s_{ij}) \left[ E^i_j(k_i - k_j) - E^j_i(k_i - k_j) + E^j_i(k_j - k_i) - E^i_j(k_j - k_i) \right] \Phi
\]

where \( E^i_j = [0, ..., 0, v, 0, ..., 0] \) is the square matrix with \( i \)-th column as vector \( v \) and other columns as all 0. For \( h'(z) \), we adopt the smooth hinge function that can handle the non-differentiability at \( z = 0 \), as suggested in (Rennie and Srebro 2005). For simplicity, we denote \( \frac{\partial E(\Omega; K)}{\partial \Omega} \) as \( \Gamma \Phi \).

Thus, at the \( t + 1 \)-th iteration, we employ the gradient descent to iteratively update \( \Omega_{t+1} \) as below:

\[
\Omega_{t+1} = \Omega_t - \rho \frac{\partial E(\Omega_t; K)}{\partial \Omega} \bigg|_{\Omega=\Omega_t} = \Omega_t - \rho \Gamma_t,
\]

where \( \rho \) is the step size for the iterations.

By replacing \( K \) with \( K_{\text{DP}}^\nu, K_{\text{HD}}^\nu \), or \( K_{\text{MP}}^\nu \) for learning metric, we obtain three methods NIME-DP, NIME-HD, and NIME-MP. Their corresponding learned metrics are denoted as \( \Omega_{\text{DP}}, \Omega_{\text{HD}}, \) and \( \Omega_{\text{MP}} \), respectively.

To further accelerate the learning process, we here adopt the SGD to calculate Eqn.(7). To predict the labels for testing samples, we use the learned metric \( \Omega \) to map testing samples from the kernel space to a transformed space. Taking NIME-DP as an example, let \( i \)-th column in \( G \) as \( g_i \) (here \( G \) can be \( G_{\text{DP}}, G_{\text{HD}}, \) or \( G_{\text{MP}} \)), \( \Omega g_i \) denotes the transformed feature vector of the \( i \)-th testing samples. Subsequently, in the transformed feature space, many alternative classifiers can be employed to predict the labels of testing samples. The processes for NIME-HD, NIME-MP and the following NIME-CK are similar.

### Combining Multiple Non-IID Representations

Instead of introducing additional parameters (e.g., imposing other regularizer) or assumption (e.g., projecting onto a unique subspace) which might lead to extra parameter tuning or heavy computational burden, NIME-CK learns the optimal metric by automatically finding the best linear combination of multiple non-IID representations. The major advantage is that NIME-CK only requires one parameter to be tuned (i.e., \( \lambda \)), and also holds the property as a single kernel method (e.g., NIME-HD), thus solvable by SGD. The objective function of NIME-CK is below:

\[
\arg \min_{\Omega, w^p} E(\Omega; \sum_p w^p K^p) \quad \text{s.t.} \quad \sum_p w^p = 1, w^p \geq 0. \tag{9}
\]

\( w^p \) is the non-negative weight for \( p \)-th (\( p = 1, 2, ..., P \)) non-IID representations. Eqn.(9) can be effectively solved by alternating optimization strategy to iteratively solve the following two sub-problems until convergence.

**Solving \( \Omega \) by fixing \( w^p \).** The process is similar as the aforementioned way (see Eqn.(8)).

**Solving \( w^p \) by fixing \( \Omega \).** This problem can be represented as follows:

\[
\arg \min_{w^p} \sum_{i,j} \psi_{ij} \|\Omega \left( \sum_p w^p k_i^p - \sum_p w^p k_j^p \right) \|^2 + \lambda \sum_{i,j,l} \psi_{ij} (1 - y_{il}) h' \left(\|\Omega \left( \sum_p w^p k_i^p - \sum_p w^p k_j^p \right) \|^2 + 1\right) \bigg\| \sum_p w^p k_i^p \bigg\|^2 + 1 \bigg]\bigg| \sum_p w^p = 1, w^p \geq 0. \tag{10}
\]

For non-IID representation \( \sum_p w^p K^p \), \( \psi_{ij} \) indicates whether \( j \)-th training sample is one of the \( k \)-closest neighbours of \( i \)-th training sample with the same label \( y_i \).

To obtain the approximated solution of \( w^p \) in a reasonable time period, we employ the Simulated Annealing (SA) for optimization. With the new kernel \( \sum_p w^p K^p \), we can classify the testing samples using the aforementioned steps.
Table 1: The error rates of all comparison methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>GMMRF</th>
<th>Geodesic Cut</th>
<th>Random Walker</th>
<th>Lazy Snapping</th>
<th>Graph Cut</th>
<th>Grab Cut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Rate (%)</td>
<td>7.9</td>
<td>4.8</td>
<td>5.4</td>
<td>6.7</td>
<td>6.7</td>
<td>5.7</td>
</tr>
<tr>
<td>(Nguyen et al. 2012)</td>
<td>3.8</td>
<td>4.83</td>
<td>5.03</td>
<td>4.15</td>
<td>3.56</td>
<td>3.31</td>
</tr>
</tbody>
</table>

Figure 2: Visualization of feature transformation in segmentation. First column: original images, second to last columns: LMNN, LMCA, NIME-DP, NIME-HD, NIME-MP, NIME-MK. Pink points belong to foreground, and blue points belong to background.

Figure 2 shows the typical examples of feature transformation for different methods by applying respectively learned metrics, in which two features with the highest Mutual Information (MI) values (Peng, Long, and Ding 2005) calculated from ground truth labels after feature transformation are selected for visualization. In Figure 3, four examples of the objective function values of NIME-CK are shown for each iteration step, showing that solving NIME-CK can converge. Also, by performing SGD, only one triplet is used to calculate the gradient in Eqn.(7), resulting in low computational cost.

Algorithm 1 shows the pseudo code of NIME-CK.

Algorithm 1 NIME-CK

Input: $K^p, \phi_{ij}$ and $y_{ij}$.  
Output: $\Omega, w_p (p = 1, \ldots, P)$. 

1: $w^1_p \leftarrow \frac{1}{P}$ 
2: $\Omega_1 \leftarrow$ Kernel PCA Initialization (2007) 
3: while not converge do 
4: $\Omega_{t+1} \leftarrow \Omega_t - \rho \Gamma_t$ in Eqn.(8) 
5: $w_{t+1}^p \leftarrow$ Solved in Eqn.(10) by SA 
6: end while

Natural Image Segmentation

We evaluate the NIME models against various segmentation methods on the MSRC image set (Rother, Kolmogorov, and Blake 2004), which is a challenging and commonly-used image set in image segmentation and with results available from many existing methods for comparison. In the MSRC image set, a single foreground needs to be segmented from the background for each image and the ground-truth segmentations are available.

The major pipeline of interactive segmentation in our work includes: (1) For each image before segmentation, we first perform superpixel over-segmentation and also extract features for each segmented superpixel; (2) Then, the seed points are given by a user, where each superpixel that contains the seed points is assigned the corresponding label (i.e., background or foreground); (3) Finally, the image segmentation is treated as a non-IID metric learning problem, in which each superpixel with its neighboring superpixels are modeled by the non-IID representation. To segment an image, the labeled superpixels are used to learn the non-IID metric; after the feature space transformation, we classify the unlabeled superpixels (using SVM with a linear kernel) to complete the segmentation.

For experimental settings, for each superpixel, we choose its adjacent superpixels as the spatial neighbors. For the step of superpixel over-segmentation, we employ SLIC (Achanta et al. 2012) to over-segment each image into a number of non-overlapping superpixels (typically 500-1500 superpixels). For feature representation, we extract LBP (30-dimensional), Gabor (48-dimensional), color (including histogram, mean, variance, with totally 66-dimensional), and intensity (4-dimensional). In total, to represent a superpixel, we extract 148-dimensional features. All the parameters (e.g., $\lambda$) are experimentally determined by inner cross validation.

Several state-of-the-art interactive image segmentation methods are chosen as baselines for comparison, including Graph Cut (Boykov and Jolly 2001), Grab Cut (Rother,
<table>
<thead>
<tr>
<th>Method</th>
<th>AC</th>
<th>SP</th>
<th>SE</th>
<th>F1</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>CKNN</td>
<td>82.0</td>
<td>83.0</td>
<td>87.0</td>
<td>80.0</td>
<td>84.0</td>
</tr>
<tr>
<td>PCA+RF</td>
<td>81.6</td>
<td>75.0</td>
<td>76.8</td>
<td>79.6</td>
<td>84.0</td>
</tr>
<tr>
<td>KPCA</td>
<td>81.5</td>
<td>79.6</td>
<td>85.3</td>
<td>81.6</td>
<td>90.0</td>
</tr>
<tr>
<td>GPLVM</td>
<td>80.0</td>
<td>76.9</td>
<td>84.0</td>
<td>87.9</td>
<td>88.6</td>
</tr>
<tr>
<td>mSRC</td>
<td>87.0</td>
<td>81.3</td>
<td>89.0</td>
<td>94.0</td>
<td>96.0</td>
</tr>
<tr>
<td>LMNN</td>
<td>84.6</td>
<td>81.5</td>
<td>86.8</td>
<td>89.3</td>
<td></td>
</tr>
<tr>
<td>LMCA</td>
<td>87.5</td>
<td>81.1</td>
<td>86.8</td>
<td>89.3</td>
<td></td>
</tr>
<tr>
<td>NIME-DP</td>
<td>85.7</td>
<td>83.5</td>
<td>87.9</td>
<td>89.3</td>
<td></td>
</tr>
<tr>
<td>NIME-HD</td>
<td>92.7</td>
<td>89.1</td>
<td>96.0</td>
<td>96.0</td>
<td></td>
</tr>
<tr>
<td>NIME-MP</td>
<td>88.0</td>
<td>85.0</td>
<td>90.0</td>
<td>96.0</td>
<td></td>
</tr>
<tr>
<td>NIME-MK</td>
<td>90.0</td>
<td>85.0</td>
<td>96.0</td>
<td>96.0</td>
<td></td>
</tr>
</tbody>
</table>

Histology Image Identification

Histology image identification is a crucial and challenging task in medical image analysis. Each histology image consists of several cells, which are spatially connected with each other. The goal of histology image identification is to classify a new image to be cancerous or normal, according to other images pre-diagnosed by pathologists (a.k.a. training images). See Figure 5(a) for more details about the cancerous and normal images.

All histological images were collected per a needle biopsy procedure. Each image was preprocessed per the following steps: image denoising and smoothing by a Gaussian kernel, gray scale conversion, and cell region segmentation by a threshold-based method. To represent the segmented cell regions, we adopt hand-crafted and deeply-learned features, respectively, to fully investigate the performance of using different feature types:

- **Hand-crafted features**: Those features whose effectiveness are already validated in (Kovalev et al. 2006)(Shi et al. 2013) are chosen, including height, width, RGB, HSI, area, circumference, Fourier descriptor, entropy, and central moment. In total, to represent a cell region, 37-dimensional features are extracted.

- **Deeply-learned features**: Regarding the relative small-scale cell regions compared with natural images, we here first use the bounding box to bound the irregular segmented cell regions, resize them into 32×32 patches, and then employ the LeNet model to automatically learn the deep features, which will form a 64-dimensional feature for each cell.
were specifically designed for histology image identification with LMNN and LMCA, our non-IID and deeply-learned features. It is observed that (1) comparison methods are listed in Table 2, and the ROC otherwise, it is diagnosed as cancerous. (2) Lee et al.’s method (Lee 2010), a two-stage diagnosis framework by first automatically selecting the most informative features and then training the classifier for diagnosis; (3) mSRC (Shi et al. 2013), a multi-modal sparse learning framework generating the dictionaries from cells, (4) Kovalev et al.’s method (Kovalev et al. 2006), a comprehensive study that experimentally compares many combinations of feature reduction methods and classifiers. We use the PCA + Random Forest (obtain the best performance in (Kovalev et al. 2006)), (5) KPCA + LMCA, and (6) GPLVM + LMCA as the baseline. Also, LMNN (Weinberger, Blitzer, and Saul 2005) and LMCA (Torresani and Lee 2007) are included, representing the metric learning methods assuming cells are IID. While our method only obtains the labels of each cell, our task is to classify the whole testing image instead of cells. A majority voting strategy is thus taken: a testing image is eventually diagnosed as normal if all the cells are classified as normal; otherwise, it is diagnosed as cancerous.

The accuracy, specificity, sensitivity, and F1 score of all comparison methods are listed in Table 2, and the ROC curves are in Figure 5(b), including both the hand-crafted and deeply-learned features. It is observed that (1) comparing with LMNN and LMCA, our non-IID metric obtains the superior results; (2) comparing with the baselines which were specifically designed for histology image identification, our methods achieve the comparable or even better performance; (3) different types of features can be easily employed in our model.

Furthermore, we have investigated the performance by choosing different numbers of spatial neighbors. By varying the number of spatial neighbors from 3 to 7, the accuracy of using hand-crafted features are 0.84, 0.89, 0.89, 0.87 and 0.87, while the accuracy of using deeply-learned features are 0.89, 0.91, 0.90, 0.89 and 0.88, respectively.

Discussions
For the relations between specific non-IID representation and application. Indeed, different applications prefer different non-IID representations, e.g., in histological image identification, neighboring cells are usually similar, thus NIME-DP can identify the grouping cancerous cells. In segmentation, max-pooling in NIME-MP can preserve the maximum matching w.r.t. object rotation. To automatically learn the best combinations when no prior knowledge is available, NIME-CK can be used to integrate various non-IID representations (but not limited to those three).

For the relation and difference with deep methods, first, the relation between layer/region in deep methods is fixed once whole architecture is designed, however, our method allows the best combination to reveal different non-IIDness can be learned; Second, our method can borrow deep features for performance improvement.

Conclusions
Metric learning for computer vision usually assumes data is IID, which is inconsistent with the fact that different neighboring regions/superpixels in images/videos are non-IID. This fundamentally challenges existing metric learning methods. Our work proposes an effective method NIME for learning and integrating multiple non-IID metrics for computer vision tasks. Its substantial comparisons with state-of-the-art baselines shows its superior performance in not only capturing the intrinsic non-IID data characteristics but being easy to implement by stochastic optimization and only needing to tune very few parameters. For future directions, our current work relies on pairwise computing, thus we are developing parallel-computing strategy and converting our method to a layer in deep models.

Acknowledgements
This research was supported by NSFC (Nos. 61673203, 61305068, 61432008, 61321491), Jiangsu Nature Science Foundation (JSNSF) (No. BK20130581). The authors would like to thank Wanqi Yang and Jing Huo for proofreading.

References


