Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

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 $\mathbf{M} = \mathbf{L}^\top\mathbf{L}$) by minimizing the pair-wise Mahalanobis distance of training samples. It is assumed that, by applying metric $\mathbf{M}$ (or $\mathbf{L}$) 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 $\mathbf{M}$ (or $\mathbf{L}$) 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]

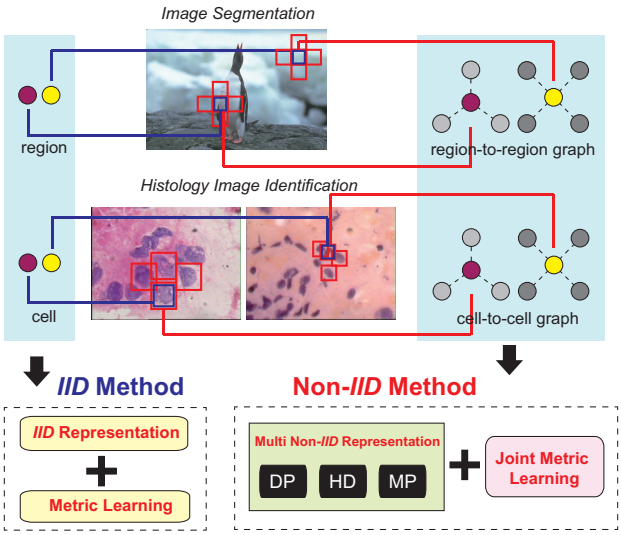


Figure 1: The illustration of comparing IID v.s. non-IID metric learning in two different applications.

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.

[1]Note 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 ($i = 1, 2, ..., N_{\text{tr}}$) is encoded as $\mathbf{x}_i \in \mathbb{R}^d$, where d is the feature dimensionality in the original feature space, and N_{tr} is the number of training samples. For $\mathbf{x}_i$, its neighbors are denoted as $\tilde{\mathbf{x}}_{i,1}$, $\tilde{\mathbf{x}}_{i,2}$, ..., $\tilde{\mathbf{x}}_{i,m_i} \in \mathbb{R}^d$, where m_i is the total number of neighbors of training sample $\mathbf{x}_i$. Similarly, we denote the i-th testing sample ($i = 1, 2, ..., N_{\text{te}}$) as $\mathbf{z}_i \in \mathbb{R}^d$, where N_{te} is the number of testing samples. For $\mathbf{z}_i$, its neighbors are denoted as $\tilde{\mathbf{z}}_{i,1}$, $\tilde{\mathbf{z}}_{i,2}$, ..., $\tilde{\mathbf{z}}_{i,n_i} \in \mathbb{R}^d$, where n_i is the total number of neighbors of testing sample $\mathbf{z}_i$.

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 $\mathbf{M}$ (or $\mathbf{L}$) in the Mahalanobis distance space can be formulated as learning metric in the corresponding graph kernel space, such as in (Jain, 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 ($\mathbf{K}_{\text{DP}}$), Hausdorff distance kernel ($\mathbf{K}_{\text{HD}}$), and max pooling kernel ($\mathbf{K}_{\text{MP}}$), obtained on all the training and testing samples.

The simplest method to compute the similarity of two undirected graphs is to first 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 ($\mathbf{K}_{\text{DP}}$). 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 ($\mathbf{K}_{\text{HD}}$). 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 ($\mathbf{K}_{\text{MP}}$).

Definition 1 *Given two samples $\mathbf{x}_i$ and $\mathbf{x}_j$, their respective m_i and m_j neighbors are denoted as $\tilde{\mathbf{x}}_{i,p}$ ($p = 1, ..., m_i$) and $\tilde{\mathbf{x}}_{j,q}$ ($q = 1, ..., m_j$) respectively, and their DP $\mathbf{K}_{\text{DP}}(i, j)$, HD $\mathbf{K}_{\text{HD}}(i, j)$, and MP $\mathbf{K}_{\text{MP}}(i, j)$ distances are defined below:*

$$\mathbf{K}_{\text{DP}}(i,j) = f(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{m_i \cdot m_j} \sum_{p=1}^{m_i} \sum_{q=1}^{m_j} f(\tilde{\mathbf{x}}_{i,p}, \tilde{\mathbf{x}}_{j,q}). \tag{1}$$

$$\mathbf{K}_{\text{HD}}(i,j) = f(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{m_i \cdot m_j} h(\mathcal{X}_i, \mathcal{X}_j). \tag{2}$$

$$\mathbf{K}_{\text{MP}}(i,j) = f(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{m_i} \sum_{p=1}^{m_i} \max_{q=1,...,m_j} f(\tilde{\mathbf{x}}_{i,p}, \tilde{\mathbf{x}}_{j,q}) + \frac{1}{m_j} \sum_{q=1}^{m_j} \max_{p=1,...,m_i} f(\tilde{\mathbf{x}}_{i,p}, \tilde{\mathbf{x}}_{j,q}). \tag{3}$$

where f is a positive semi-definite kernel. In Eqn.(2), $\mathcal{X}_i = \{\tilde{\mathbf{x}}_{i,1}, \tilde{\mathbf{x}}_{i,2}, ..., \tilde{\mathbf{x}}_{i,m_i}\}$ is the set including all the neighbours of $\mathbf{x}_i$, 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 $f(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\alpha_1 \|\mathbf{x}_i - \mathbf{x}_j\|^2)$ for simplicity, where α_1 is the parameter. In Eqn.(2), the second term calculates the HD between two sets $\mathcal{X}_i$ and $\mathcal{X}_j$. For two sets $\mathcal{A}$ and $\mathcal{B}$, HD kernel function h is defined below.

$$h(\mathcal{A}, \mathcal{B}) = \exp\Big(- \alpha_2 \cdot \max\big(H(\mathcal{A}, \mathcal{B}), H(\mathcal{B}, \mathcal{A})\big)\Big), \tag{4}$$

where α_2 is the parameter, and

$$H(\mathcal{A}, \mathcal{B}) = \max_{\mathbf{a} \in \mathcal{A}} \min_{\mathbf{b} \in \mathcal{B}} \|\mathbf{a} - \mathbf{b}\|, \; H(\mathcal{B}, \mathcal{A}) = \max_{\mathbf{b} \in \mathcal{B}} \min_{\mathbf{a} \in \mathcal{A}} \|\mathbf{b} - \mathbf{a}\|.$$

Eqns.(1), (2) and (3) are easy to implement and fast to compute. All the graph kernels $\mathbf{K}_{\text{DP}}$, $\mathbf{K}_{\text{HD}}$ and $\mathbf{K}_{\text{MP}}$ are computed on all the training samples ($\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{N_{\text{tr}}}$) and testing samples ($\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_{N_{\text{te}}}$), hence, $\mathbf{K}_{\text{DP}}$, $\mathbf{K}_{\text{HD}}$, $\mathbf{K}_{\text{MP}} \in \mathbb{R}^{(N_{\text{tr}}+N_{\text{te}}) \times (N_{\text{tr}}+N_{\text{te}})}$. Also, all the graph kernels should be processed by using kernel alignment, i.e., for a kernel function $\mathbf{K}$, rescaling each $\mathbf{K}(i, j)$ as $\mathbf{K}(i,j) = \frac{\mathbf{K}(i,j)}{\sqrt{\mathbf{K}(i,i)} \cdot \sqrt{\mathbf{K}(j,j)}}$.

For better representation, without loss of generality we put all the training samples before the testing samples. Taking $\mathbf{K}_{\text{DP}}$ as an example, we split the graph kernel matrix as follows:

$$\mathbf{K}_{\text{DP}} = \begin{pmatrix} \mathbf{K}_{\text{DP}}^{\text{tr}} & \mathbf{G}_{\text{DP}} \\ \mathbf{G}_{\text{DP}}^{\top} & \mathbf{K}_{\text{DP}}^{\text{te}} \end{pmatrix}, \tag{5}$$

where $\mathbf{K}_{\text{DP}}^{\text{tr}} \in \mathbb{R}^{N_{\text{tr}} \times N_{\text{tr}}}$ and $\mathbf{K}_{\text{DP}}^{\text{te}} \in \mathbb{R}^{N_{\text{te}} \times N_{\text{te}}}$ are the parts of graph kernel matrix calculated on the training samples and testing samples, respectively. $\mathbf{G}_{\text{DP}} \in \mathbb{R}^{N_{\text{tr}} \times N_{\text{te}}}$ 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* $\mathbf{L}$ *(a learned metric in original feature space), and a nonlinear mapping* $\mathbf{\Phi}$*, the projection of feature vector transformation is equivalent to*

$$\mathbf{L}\mathbf{\Phi} = \mathbf{\Omega}\mathbf{K}, \tag{6}$$

where $\mathbf{\Omega}$ *is the matrix that represents* $\mathbf{L}$ *as a linear combination of the feature vectors, and* $\mathbf{K}$ *is a corresponding kernel matrix corresponding to the nonlinear mapping* $\mathbf{\Phi}$*.*

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

We now discuss how to learn metric $\mathbf{\Omega}$. First, we introduce the method to learn the metric by only 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 $\mathbf{K}$ (here $\mathbf{K}$ can be $\mathbf{K}^{\text{tr}}_{\text{DP}}$, $\mathbf{K}^{\text{tr}}_{\text{HD}}$, or $\mathbf{K}^{\text{tr}}_{\text{MP}}$) calculated on all the training samples with known labels, the task of learning metric $\mathbf{\Omega}$ with single non-*IID* representation is to minimize the following objective function:

$$\mathcal{E}(\mathbf{\Omega};\mathbf{K}) = \sum_{i,j}\eta_{ij}\|\mathbf{\Omega}(\mathbf{k}_i-\mathbf{k}_j)\|^2 + \lambda\sum_{i,j,l}\eta_{ij}(1-y_{il})$$
$$h\Big(\|\mathbf{\Omega}(\mathbf{k}_i-\mathbf{k}_j)\|^2 - \|\mathbf{\Omega}(\mathbf{k}_i-\mathbf{k}_l)\|^2 + 1\Big).$$

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 loss function. λ is the weight parameter of the second term. $\mathbf{k}_i$ is the i-th column of $\mathbf{K}$.

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

$$\begin{aligned}\frac{\partial\mathcal{E}(\mathbf{\Omega};\mathbf{K})}{\partial\mathbf{\Omega}} =& \Big(2\mathbf{\Omega}\sum_{i,j}\eta_{ij}\Big[\mathbf{E}_i^{(\mathbf{k}_i-\mathbf{k}_j)} - \mathbf{E}_j^{(\mathbf{k}_i-\mathbf{k}_j)}\Big]\Big)\mathbf{\Phi} \\ &+\Big(2\lambda\mathbf{\Omega}\sum_{i,j,l}\eta_{ij}(1-y_{il})h'(s_{ijl}) \\ &\Big[\mathbf{E}_i^{(\mathbf{k}_i-\mathbf{k}_j)} - \mathbf{E}_j^{(\mathbf{k}_i-\mathbf{k}_j)} - \mathbf{E}_i^{(\mathbf{k}_i-\mathbf{k}_l)} + \mathbf{E}_l^{(\mathbf{k}_i-\mathbf{k}_l)}\Big]\Big)\mathbf{\Phi}.\end{aligned} \tag{7}$$

where $\mathbf{E}_i^{\mathbf{v}} = [0,...,0,\mathbf{v},0,...,0]$ is the square matrix with i-th column as vector $\mathbf{v}$ 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 $\partial\mathcal{E}(\mathbf{\Omega};\mathbf{K})/\partial\mathbf{\Omega}$ as $\mathbf{\Gamma}\mathbf{\Phi}$. Thus, at the $t+1$-th iteration, we employ the gradient descent to iteratively update $\mathbf{\Omega}_{t+1}$ as below:

$$\mathbf{\Omega}_{t+1} = \mathbf{\Omega}_t - \rho\left.\frac{\partial\mathcal{E}(\mathbf{\Omega};\mathbf{K})}{\partial\mathbf{\Omega}}\right|_{\mathbf{\Omega}=\mathbf{\Omega}_t} = \mathbf{\Omega}_t - \rho\mathbf{\Gamma}_t, \tag{8}$$

where ρ is the step size for the iterations.

By replacing $\mathbf{K}$ with $\mathbf{K}^{\text{tr}}_{\text{DP}}$, $\mathbf{K}^{\text{tr}}_{\text{HD}}$, or $\mathbf{K}^{\text{tr}}_{\text{MP}}$ for learning metric, we obtain three methods NIME-DP, NIME-HD, and NIME-MP. Their corresponding learned metrics are denoted as $\mathbf{\Omega}_{\text{DP}}$, $\mathbf{\Omega}_{\text{HD}}$, and $\mathbf{\Omega}_{\text{MP}}$, 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 $\mathbf{\Omega}$ to map testing samples from the kernel space to a transformed space. Taking NIME-DP as an example, let i-th column in $\mathbf{G}$ as $\mathbf{g}_i$ (here $\mathbf{G}$ can be $\mathbf{G}_{\text{DP}}$, $\mathbf{G}_{\text{HD}}$, or $\mathbf{G}_{\text{MP}}$), $\mathbf{\Omega}\mathbf{g}_i$ 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., λ), 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:

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

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 $\mathbf{\Omega}$ by fixing w^p. The process is similar as the aforementioned way (see Eqn.(8)).

Solving w^p by fixing $\mathbf{\Omega}$. This problem can be represented as follows:

$$\begin{aligned}\arg\min_{w^p}&\sum_{i,j}\psi_{ij}\|\mathbf{\Omega}\Big(\sum_p w^p\mathbf{k}_i^p - \sum_p w^p\mathbf{k}_j^p\Big)\|^2 + \\ &\lambda\sum_{i,j,l}\psi_{ij}(1-y_{il})h\Big[\|\mathbf{\Omega}\Big(\sum_p w^p\mathbf{k}_i^p - \sum_p w^p\mathbf{k}_j^p\Big)\|^2 \\ &- \|\mathbf{\Omega}\Big(\sum_p w^p\mathbf{k}_i^p - \sum_p w^p\mathbf{k}_l^p\Big)\|^2 + 1\Big]. \\ \text{s.t.}&\sum_p w^p = 1, w^p \geq 0.\end{aligned} \tag{10}$$

For non-*IID* representation $\sum_p w^p\mathbf{K}^p$, ψ_{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 $\sum_p w^p\mathbf{K}^p$, we can classify the testing samples using the aforementioned steps.

Table 1: The error rates of all comparison methods.

Method	GMMRF	Geodesic Cut	Random Walker	Lazy Snapping	Graph Cut	Grab Cut
Error Rate (%)	7.9	4.8	5.4	6.7	6.7	5.7
(Nguyen et al. 2012)	LMNN	LMCA	NIME-DP	NIME-HD	NIME-MP	NIME-MK
3.8	4.83	5.03	4.15	3.56	3.31	3.27

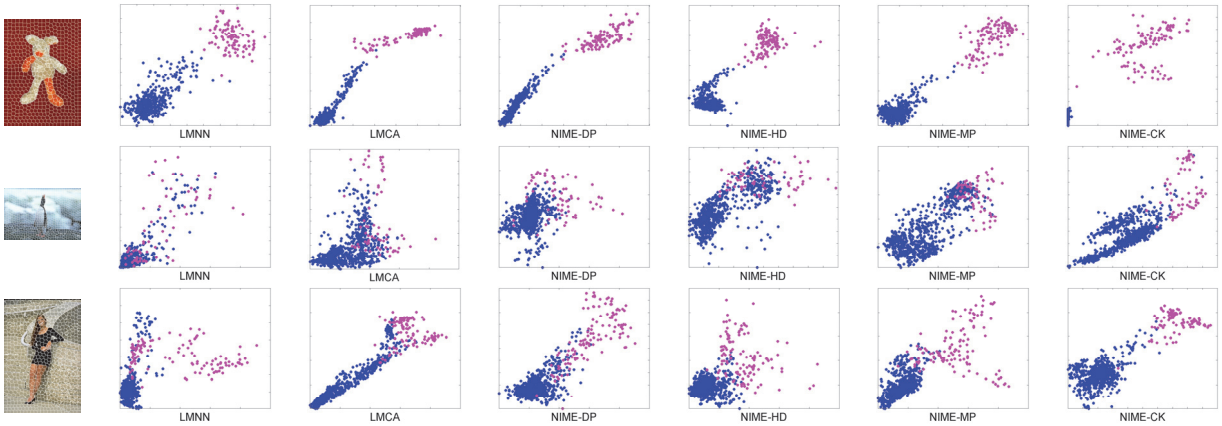


Figure 2: Visualization of feature transformation in segmentation. First column: original images, second to last columns: LMNN, LMCA, NIME-DP, NIME-HD, NIME-MP, NIME-CK. Pink points belong to foreground, and blue points belong to background.

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: $\mathbf{K}^p$, ϕ_{ij} and y_{ij}.
Output: $\mathbf{\Omega}$, w^p $(p = 1, ..., P)$.

1: $w_1^p \leftarrow \frac{1}{P}$
2: $\mathbf{\Omega}_1 \leftarrow$ Kernel PCA Initialization (2007)
3: **while** not converge **do**
4: $\quad \mathbf{\Omega}_{t+1} \leftarrow \mathbf{\Omega}_t - \rho \mathbf{\Gamma}_t$ in Eqn.(8)
5: $\quad w_{t+1}^p \leftarrow$ 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., λ) 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 2: The results of Accuracy (AC), Specificity (SP), Sensitivity (SE), F1 Score, and AUC of all methods. HC and DL denote hand-crafted and deeply-learned features, respectively.

Method	(Lee 2010)	CKNN	PCA+RF	KPCA	GPLVM	mSRC	LMNN	LMCA	NIME-DP	NIME-HD	NIME-MP	NIME-MK
AC_{HC}	82.0	85.0	79.0	75.0	81.0	87.0	80.0	77.0	86.0	83.0	84.0	**89.0**
SP_{HC}	80.8	83.0	76.4	76.6	78.2	87.8	78.9	76.5	84.6	85.1	88.6	**91.5**
SE_{HC}	83.3	87.2	82.2	73.6	84.4	86.3	81.3	77.6	**87.5**	81.1	80.4	86.8
$F1_{HC}$	81.6	84.5	77.9	75.7	80.0	87.1	79.6	76.8	85.7	83.5	84.9	**89.3**
AUC_{HC}	87.9	91.6	84.2	79.1	86.8	93.8	85.3	81.6	92.7	89.1	90.6	**96.0**
AC_{DL}	86.0	84.0	82.0	79.0	81.0	86.0	81.0	79.0	88.0	85.0	84.0	**90.0**
SP_{DL}	89.1	84.0	83.3	76.4	81.6	89.1	81.6	80.9	**89.6**	85.7	79.3	88.5
SE_{DL}	83.3	84.0	80.8	82.2	80.4	83.3	80.4	77.4	86.6	84.3	90.5	**91.7**
$F1_{DL}$	86.5	84.0	82.4	77.9	81.2	86.5	81.2	79.6	88.2	85.2	82.6	**89.8**
AUC_{DL}	92.8	90.3	87.9	84.2	86.6	92.8	86.6	84.1	95.0	91.5	90.8	**96.9**

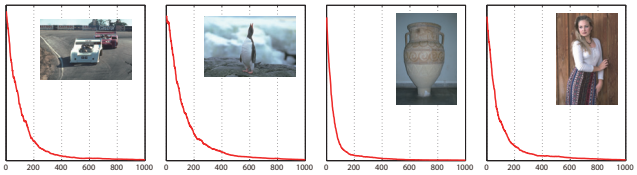


Figure 3: Illustration of the convergence of NIME-CK.

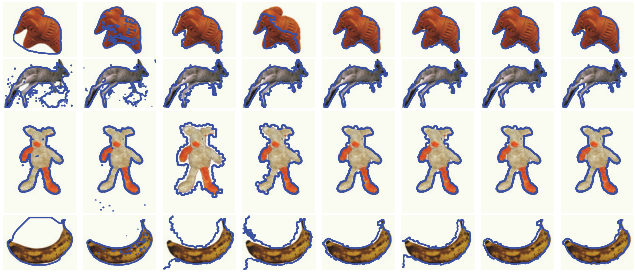


Figure 4: Typical results. First to last columns: Graph Cut, Grab Cut, LMNN, LMCA, NIME-DP, NIME-HD, NIME-MP, NIME-CK.

Kolmogorov, and Blake 2004), LMNN (Weinberger, Blitzer, and Saul 2005) and LMCA (Torresani and Lee 2007) (*IID* methods). For other methods including Nguyen *et al.* (Nguyen et al. 2012), Random Walker (Grady 2006), Geodesic Cut (Price, Morse, and Cohen 2010), and GMMRF (Blake et al. 2004), we simply cite their best performance on MSRC image set from the respective literatures. Note that the same labeled seed points are used for all these methods.

Since pixel-level error rate is widely used for performance evaluation (i.e., the number of misclassified pixels divided by the number of pixels in the unclassified region), we report it in Table 1. It shows that the non-*IID* metric learning methods outperform the *IID* metric learning LMNN and LMCA. Our methods achieve better segmentation performance than GMMRF, Random Walker, Geodesic Cut, Grab Cut, Graph Cut, and Lazy Snapping. Moreover, compared with (Nguyen et al. 2012) that was specifically designed for image segmentation, our methods still have comparable or even better results. The results are illustrated in Figure 4, with the non-light yellow region as the true foreground, and the blue contour as the predicted results of respective methods.

It is noteworthy that we did not compare ours with deep methods in image segmentation. First, during segmentation, the training and testing samples come from the same image, i.e., we do not use additional samples (or images) for model training as that in deep methods. Second, the extracted local regions by super-pixel partition are with irregular shape, rather than squared patches in deep methods. Also for NIME, we have investigated different initializations of seed points. The performance normally remains stable, since the non-*IID* assumption preserves spatial relations to avoid possible labelling noise.

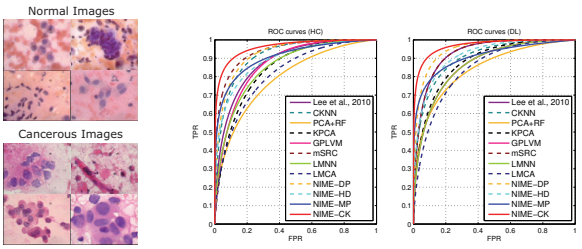
Histology Image Identification

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 used.

• *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



(a) Images. (b) The ROC curves.

Figure 5: Application to histology image identification. HC and DL denotes hand-crafted and deeply-learned features, respectively.

region.

The experiments were carried on 100 real histology images (each image normally contains 10-300 cell regions) provided by a cancer research center, including 50 cancerous and 50 normal images, respectively. Each histology image is labeled by a pathologist for experimental evaluation. 10-fold cross validation was taken for all the baseline and our methods. All the parameters (e.g., λ) were experimentally determined by inner cross validation.

In non-*IID* metric learning, for each current cell, we empirically choose its top 5 closest cells as the spatial neighbors. Thus, the constructed non-*IID* graph is with one central node (current cell) and 5 surrounding nodes (5 spatially neighboring cells).

We choose six related methods as the baseline, including (1) Citation-kNN (CKNN) (Wang and Zucker 2000), a classical multi-instance learning method assuming that the cells in each image follow the bag-of-instance setting; (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 baselines. 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-*IID*ness 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

Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; and Susstrunk, S. 2012. Slic superpixels compared to state-of-the-art superpixel methods. *TPAMI* 34:2274–2282.

Bellet, A.; Habrard, A.; and Sebban, M. 2013. A survey on metric learning for feature vectors and structured data. *CoRR* abs/1306.6709.

Blake, A.; Rother, C.; Brown, M.; Perez, P.; and Torr, P. 2004. Interactive image segmentation using an adaptive gmmrf model. *ECCV* 428–441.

Bottou, L. 2010. Large-scale machine learning with stochastic gradient descent. *COMPSTAT* 177–187.

Boykov, Y. Y., and Jolly, M.-P. 2001. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. *ICCV* 105–112.

Cao, L.; Ou, Y.; and Yu, P. 2012. Coupled behavior analysis with applications. *IEEE Trans. Knoweldge and Data Engineering* 24:1378–1392.

Cao, L. 2014. Non-iidness learning in behavioral and social data. *The Computer Journal* 57(9):1358–1370.

Cao, L. 2015. Coupling learning of complex interactions. *Information Processing & Management* 51(2):167–186.

Davis, J.; Kulis, B.; Jain, P.; Sra, S.; and Dhillon, I. S. 2007. Information-theoretic metric learning. *ICML*.

Gao, X.; Hoi, S.; Zhang, Y.; and ang J. Li, J. W. 2014. Soml: Sparse online metric learning with application to image retrieval. *AAAI*.

Globerson, A., and Roweis, S. 2005. Metric learning by collapsing classes. *NIPS*.

Goldberger, J.; Roweis, S.; Hinton, G.; and Salakhutdinov, R. 2004. Neighborhood components analysis. *NIPS*.

Grady, L. 2006. Random walks for image segmentation. *TPAMI* 28:1768–1783.

Guillaumin, M.; Mensink, T.; Verbeek, J.; and Schmid, C. 2009. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. *ICCV*.

Guillaumin, M.; Verbeek, J.; and Schmid, C. 2009. Is that you? metric learning approaches for face identification. *ICCV* 498–505.

Huang, Z.; Wang, R.; Shan, S.; and Chen, X. 2015. Projection metric learning on grassmann manifold with application to video based face recognition. *CVPR*.

Jain, P.; Kulis, B.; and Dhillon, I. 2010. Inductive regularized learning of kernel functions. *NIPS* 946–954.

Jiang, N.; Liu, W.; and Wu, Y. 2012. Order determination and sparsity-regularized metric learning adaptive visual tracking. *CVPR* 1956–1963.

Kondor, R., and Jebara, T. 2003. A kernel between sets of vectors. *ICML*.

Kovalev, V.; Harder, N.; Neumann, B.; Held, M.; Liebel, U.; Erfle, H.; Ellenberg, J.; Neumann, B.; Eils, R.; and Rohr, K. 2006. Feature selection for evaluating fluorescence microscopy images in genome-wide cell screens. *CVPR*.

Lajugie, R.; Arlot, S.; and Bach, F. 2014. Large-margin metric learning for constrained partitioning problems. *ICML*.

Lee, M. C. 2010. Computer-aided diagnosis of pulmonary nodules using a two-step approach for feature selection and classifier ensemble construction. *AIM* 50:43–53.

Li, X.; Shen, C.; Shi, Q.; Dick, A.; and van den Hengel, A. 2012. Linear representations for visual tracking with online reservoir metric learning. *CVPR* 1760–1767.

Liao, S., and Li, S. 2015. Efficient psd constrained asymmetric metric learning for person re-identification. *ICCV*.

Lyu, S. 2010. Mercer kernels for object recognition with local features. *CVPR* 223–229.

Mensink, T.; Verbeek, J.; Perronnin, F.; and Csurka, G. 2012. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. *ECCV*.

Nguyen, T.; Cai, J.; Zhang, J.; and Zheng, J. 2012. Robust interactive image segmentation using convex active contours. *TIP* 21:3734–3743.

Peng, H.; Long, F.; and Ding, C. 2005. Feature selection based on mutualinformation: criteria of max-dependency, max-relevance, and min-redundancy. *TPAMI* 27:1226–1238.

Price, B.; Morse, B.; and Cohen, S. 2010. Geodesic graph cut for interactive image segmentation. *CVPR*.

Rennie, J. D. M., and Srebro, N. 2005. Fast maximum margin matrix factorization for collaborative prediction. *ICML* 713–719.

Rother, C.; Kolmogorov, V.; and Blake, A. 2004. Grabcut: Interactive foreground extraction using iterated graph cuts. *SIGGRAPH* 309–314.

Shental, N.; Hertz, T.; Weinshall, D.; and Pavel, M. 2002. Adjustment learning and relevant component analysis. *ECCV* 776–790.

Shi, Y.; Gao, Y.; Yang, Y.; Wang, D.; and Zhang, Y. 2013. Multi-modal sparse representation-based classification for lung needle biopsy images. *TBME* 60:2675–2684.

Teney, D.; Brown, M.; Kit, D.; and Hall, P. 2015. Learning similarity metrics for dynamic scene segmentation. *CVPR*.

Torresani, L., and Lee, K. 2007. Large margin component analysis. *NIPS* 1385–1392.

Verma, N.; Mahajan, D.; Sellamanickam, S.; and Nair, V. 2012. Learning hierarchical similarity metrics. *CVPR*.

Wang, C., and Cao, L. 2011. Coupled nominal similarity in unsupervised learning. *CIKM* 973–978.

Wang, D., and Tan, X. 2014. Robust distance metric learning in the presence of label noise. *AAAI*.

Wang, J., and Zucker, J.-D. 2000. Solving the multiple-instance problem: A lazy learning approach. *ICML* 1119–1126.

Wang, J.; Do, H.; Woznica, A.; and Kalousis, A. 2011. Metric learning with multiple kernels. *NIPS*.

Wang, H.; Wang, W.; Zhang, C.; and Xu, F. 2014. Cross-domain metric learning based on information theory. *AAAI*.

Weinberger, K.; Blitzer, J.; and Saul, L. 2005. Distance metric learning for large margin nearest neighbor classification. *NIPS* 1473–1480.

Woznica, A.; Kalousis, A.; and Hilario, M. 2006. Distances and (indefinite) kernels for sets of objects. *ICDM*.

Xing, E.; Jordan, M.; Russell, S.; and Ng, A. 2002. Distance metric learning with application to clustering with side-information. *NIPS* 505–512.

Zhou, Z.-H.; Sun, Y.-Y.; and Li, Y.-F. 2009. Multi-instance learning by treating instances as non-iid samples. *ICML* 1249–1256.