ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-381

Embedding Heterogeneous Data by Preserving Multiple Kernels

Mehmet Gönen[1]

Abstract. Heterogeneous data may arise in many real-life applications under different scenarios. In this paper, we formulate a general framework to address the problem of modeling heterogeneous data. Our main contribution is a novel embedding method, called multiple kernel preserving embedding (MKPE), which projects heterogeneous data into a unified embedding space by preserving cross-domain interactions and within-domain similarities simultaneously. These interactions and similarities between data points are approximated with Gaussian kernels to transfer local neighborhood information to the projected subspace. We also extend our method for out-of-sample embedding using a parametric formulation in the projection step. The performance of MKPE is illustrated on two tasks: (i) modeling biological interaction networks and (ii) cross-domain information retrieval. Empirical results of these two tasks validate the predictive performance of our algorithm.

1 INTRODUCTION

In many real-life applications, data come from heterogeneous sources. These applications can be divided into two basic categories: (i) Heterogeneity may be coming from different representations (i.e., *modalities* or *views*) of the same domain, which is studied under the names of *multiview learning*, *transfer learning*, and *domain adaptation*. (ii) The task at hand may consider data from different domains, leading to heterogeneity, which is frequently used for recommender systems and modeling interaction networks because these work on objects from two domains by definition.

When we have multiple representations from the same domain, the most common strategy is to use *canonical correlation analysis* (CCA) [11], which finds a common subspace by maximizing correlation. CCA type of models are especially useful for cross-domain information retrieval tasks, where we have multiple representations of documents such as image and text. However, such models require having matching samples from these representations. When there is no one-to-one correspondence between samples, we need to use some additional information from the original data such as class membership to find correspondence between samples of different representations when learning the common subspace. Similarly, when we have samples from different domains, we again need to capture cross-domain interactions.

The most studied heterogeneous data problem is cross-domain information retrieval, where target documents are represented in different forms such as image and text. [19] addresses this task with a two-step learning algorithm: (i) They represent image documents using histograms obtained from k-means clustering on their SIFT features and text documents using topic probabilities obtained from latent Dirichlet allocation. (ii) They find a common subspace for these two extracted representations using CCA. [18] gives a multiview metric learning algorithm, which projects data points from different views into a shared subspace by trying to capture cross- and within-view similarities in this space. [27] proposes to define a similarity measure between cross-domain objects by looking at the class labels of their neighbors, which can be used to train standard learning algorithms.

Another popular solution strategy for cross-domain information retrieval tasks is to use hashing-based algorithms. These methods map documents from different domains into a common Hamming space (i.e., representing documents with binary vectors) instead of an Euclidean space and using a binary representation allows us to find relevant documents very fast for a new document and to reduce storage requirement drastically. [28] gives a hashing algorithm working on multiple views available for all samples, which limits the applicability to data sets with fully matching samples across domains. [1] formulates cross-domain hashing as a binary classification problem and use a boosting-based algorithm to find binary representations. [16] also proposes a cross-domain hashing algorithm that tries to map similar objects to similar codes across the views. [29] gives a probabilistic model to learn hash functions on different domains simultaneously using cross- and within-domain similarities.

Modeling heterogeneous data is also needed in transfer learning or domain adaptation settings, where we want to make use of available additional data (i.e., *source domain*) to improve the generalization performance on the task with limited data (i.e., *target domain*). [20] propose a domain adaptation method for images recorded under different conditions. [23] formulate a transfer learning algorithm using spectral embedding to find a unified subspace for both domains.

Heterogeneous data arise naturally in bioinformatics domain especially for biological interaction networks. Two well-known examples are drug–protein interaction networks [7, 25, 26] and host–pathogen protein–protein interaction networks [4], which consider two different domains (e.g., drug compounds and proteins) by definition. For drug–protein interaction, [25] and [26] find a common subspace for drugs and proteins using cross-domain interactions and within-domain similarities, and perform distance-based predictions using in this common subspace. [7] gives a Bayesian matrix factorization method that tries to reconstruct the cross-domain interaction network from the within-domain similarities. In a different application, [13] proposes a method to learn protein–protein interaction networks of multiple species using cross- and within-species similarities.

There are many embedding algorithms for single-domain applications in the literature and they mainly differ in the criteria they try to preserve while learning the embedding coordinates. We can choose

[1] Department of Computational Biology, Sage Bionetworks, Seattle, WA 98109, USA, email:mehmet.gonen@sagebase.org
Present address: Department of Biomedical Engineering, Oregon Health & Science University, Portland, OR 97239, USA, email: gonen@ohsu.edu

to preserve distances, dissimilarities, neighborhoods, or similarities. The most standard algorithm for preserving distances or dissimilarities is *multidimensional scaling* (MDS) method [3], which basically approximates the provided distances or dissimilarities in the original domain with Euclidean distances in the embedding space. There is also a non-metric version of MDS that tries to preserve the rank orders of given distances or dissimilarities [15]. [9] gives a formulation that approximates the input kernel calculated using the original representation with a standard kernel calculated in the embedding space. However, these methods are not applicable to heterogeneous data.

[6] gives an embedding method for objects from different domains using their cross- and within-domain co-occurrence statistics. They model the joint distributions as exponentials of Euclidean distances in the embedding space. [2] formulates a non-metric MDS variant that tries to place reference correspondence pairs, which share the same semantic meaning across different domains, close to each other. Their algorithm tries both to preserve within-domain relationships and to maximize alignment between domains using correspondences. Following these lines of research, we basically propose to preserve cross-domain interactions and within-domain similarities by approximating them with kernels.

In this paper, we address the problem of modeling heterogeneous data by formulating a general framework. The main idea behind our formulation is to model heterogeneous data by projecting them into a unified embedding space. This embedding step with its novel optimization formulation tries to preserve cross-domain interactions and within-domain similarities simultaneously by approximating them with multiple kernels. The proposed framework can be applied to different tasks after casting them into our formulation by defining score functions for cross-domain interactions and within-domain similarities. Note that our formulation is very different than combining multiple kernel functions to get a better one, which is known as *multiple kernel learning* [8].

Section 2 introduces the proposed embedding algorithm, called *multiple kernel preserving embedding* (MKPE), and gives detailed derivations of our optimization procedure. In Section 3, we extend our method towards out-of-sample embedding. Section 4 evaluates MKPE on two tasks: (i) modeling biological interaction networks and (ii) cross-domain information retrieval.

2 MULTIPLE KERNEL PRESERVING EMBEDDING

In order to model both cross-domain interactions and within-domain similarities, we assume that these are provided as scoring functions between objects and we want to approximate these values in the embedding space with kernel function values calculated between low-dimensional representations. Our algorithm is applicable to problems with more than two domains, but we give its details with two domains for simplicity. We first introduce the necessary notation for our method and then describe its optimization strategy in detail.

Our heterogeneous data come from two different domains, namely, $\mathcal{X}$ and $\mathcal{Z}$, and we are given two sets of objects $\mathbf{X} = \{\boldsymbol{x}_i \in \mathcal{X}\}_{i=1}^{N_\mathrm{x}}$ and $\mathbf{Z} = \{\boldsymbol{z}_i \in \mathcal{Z}\}_{i=1}^{N_\mathrm{z}}$. In standard applications, these objects have vectorial representations (i.e., $\mathcal{X}$ and $\mathcal{Z}$ are Euclidean spaces). However, these two domains may also contain non-vectorial but structured objects such as strings used for proteins and graphs used for chemical compounds in bioinformatics applications. In order to have a general formulation for both vectorial and non-vectorial data, we assume that the cross-domain interactions and the within-domain similarities are provided with three different scoring functions: (i) $s_{\mathrm{c},ij}^i \colon \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ gives the cross-domain interaction score between $\boldsymbol{x}_i$ and $\boldsymbol{z}_j$, (ii) $s_{\mathrm{x},ij}^i \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ gives the within-domain similarity score between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and (iii) $s_{\mathrm{z},ij}^i \colon \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ gives the within-domain similarity score between $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$. We also introduce three index sets, namely, $\mathcal{I}_\mathrm{c} = \{(i,j) \colon s_{\mathrm{c},ij}^i \text{ is known}\}$, $\mathcal{I}_\mathrm{x} = \{(i,j) \colon s_{\mathrm{x},ij}^i \text{ is known}\}$, and $\mathcal{I}_\mathrm{z} = \{(i,j) \colon s_{\mathrm{z},ij}^i \text{ is known}\}$, to represent available information coming from these scoring functions.

We map heterogeneous objects from two different domains into a unified embedding space. The objects in $\mathbf{X}$ and $\mathbf{Z}$ are converted into R-dimensional vectors of an Euclidean space, namely, $\mathbf{E}_\mathrm{x} = \{\boldsymbol{e}_{\mathrm{x},i} \in \mathbb{R}^R\}_{i=1}^{N_\mathrm{x}}$ and $\mathbf{E}_\mathrm{z} = \{\boldsymbol{e}_{\mathrm{z},i} \in \mathbb{R}^R\}_{i=1}^{N_\mathrm{z}}$. We try to approximate the scoring functions $s_{\mathrm{c},ij}^i$, $s_{\mathrm{x},ij}^i$, and $s_{\mathrm{z},ij}^i$ by three kernel functions, namely, $k_{\mathrm{c},ij}^i \colon \mathbb{R}^R \times \mathbb{R}^R \to \mathbb{R}$, $k_{\mathrm{x},ij}^i \colon \mathbb{R}^R \times \mathbb{R}^R \to \mathbb{R}$, and $k_{\mathrm{z},ij}^i \colon \mathbb{R}^R \times \mathbb{R}^R \to \mathbb{R}$. These three kernel functions in the embedding space have to be differentiable with respect to the embedding coordinates to be able to calculate the gradients required for the subsequent optimization step. We propose to use the Gaussian kernel (also known as radial basis function kernel or squared exponential kernel) in the embedding space to capture the local neighborhood information coming from the cross-domain interactions and within-domain similarities. The kernel functions in the embedding space can be written as

$$k_{\mathrm{c},ij}^i = \exp\left(-\frac{\|\boldsymbol{e}_{\mathrm{x},i} - \boldsymbol{e}_{\mathrm{z},j}\|_2^2}{\sigma_\mathrm{e}^2}\right) = \exp(\mathcal{Q}_{\mathrm{c},ij}^i) \qquad \forall (i,j)$$

$$k_{\mathrm{x},ij}^i = \exp\left(-\frac{\|\boldsymbol{e}_{\mathrm{x},i} - \boldsymbol{e}_{\mathrm{x},j}\|_2^2}{\sigma_\mathrm{e}^2}\right) = \exp(\mathcal{Q}_{\mathrm{x},ij}^i) \qquad \forall (i,j)$$

$$k_{\mathrm{z},ij}^i = \exp\left(-\frac{\|\boldsymbol{e}_{\mathrm{z},i} - \boldsymbol{e}_{\mathrm{z},j}\|_2^2}{\sigma_\mathrm{e}^2}\right) = \exp(\mathcal{Q}_{\mathrm{z},ij}^i) \qquad \forall (i,j),$$

where $\sigma_\mathrm{e} \in \mathbb{R}_{++}$ is the kernel width and the auxiliary variables, namely, $\mathcal{Q}_{\mathrm{c},ij}^i$, $\mathcal{Q}_{\mathrm{x},ij}^i$, and $\mathcal{Q}_{\mathrm{z},ij}^i$, are just for simplicity.

We propose to preserve the interaction and similarity scores simultaneously using a composite loss function:

$$\mathcal{L} = \frac{\lambda_\mathrm{c}}{|\mathcal{I}_\mathrm{c}|} \sum_{\mathcal{I}_\mathrm{c}} (k_{\mathrm{c},ij}^i - s_{\mathrm{c},ij}^i)^2 + \frac{\lambda_\mathrm{x}}{|\mathcal{I}_\mathrm{x}|} \sum_{\mathcal{I}_\mathrm{x}} (k_{\mathrm{x},ij}^i - s_{\mathrm{x},ij}^i)^2 + \frac{\lambda_\mathrm{z}}{|\mathcal{I}_\mathrm{z}|} \sum_{\mathcal{I}_\mathrm{z}} (k_{\mathrm{z},ij}^i - s_{\mathrm{z},ij}^i)^2,$$

where $|\cdot|$ gives the cardinality of the input set. We have separate mean squared error terms as loss functions and separate regularization parameters, namely, $\lambda_\mathrm{c} \in \mathbb{R}_+$, $\lambda_\mathrm{x} \in \mathbb{R}_+$, and $\lambda_\mathrm{z} \in \mathbb{R}_+$, to tune their weights.

The corresponding optimization problem is formulated as

$$\begin{aligned} &\text{minimize} && \mathcal{L} \\ &\text{with respect to} && \mathbf{E}_\mathrm{x} \in \mathbb{R}^{R \times N_\mathrm{x}},\ \mathbf{E}_\mathrm{z} \in \mathbb{R}^{R \times N_\mathrm{z}},\ \sigma_\mathrm{e} \in \mathbb{R}_{++} \\ &\text{subject to} && \mathbf{E}_\mathrm{x}\mathbf{E}_\mathrm{x}^\top = \mathbf{I}_R,\ \mathbf{E}_\mathrm{z}\mathbf{E}_\mathrm{z}^\top = \mathbf{I}_R, \end{aligned}$$

where we assume orthonormality of the embedding dimensions in each domain separately. This assumption enables us to avoid the scaling ambiguity and to capture useful information in each dimension of the embedding space. The objective function of our optimization problem is non-convex due to non-linearity introduced by the Gaussian kernels and global optimization is not possible. Instead, we formulate a gradient-based optimization strategy to find a local optimum. In our optimization procedure, we need to satisfy the orthonormality constraints on the embedding coordinates in addition to the non-negativity constraint on the kernel width.

We can find the gradients of $\mathcal{L}$ with respect to the embedding coordinates as

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{e}_{\mathrm{x},l}} = 2\frac{\lambda_{\mathrm{c}}}{|\mathcal{I}_{\mathrm{c}}|}\sum_{\mathcal{I}_{\mathrm{c}}} k^i_{\mathrm{c},j}(k^i_{\mathrm{c},j} - s^i_{\mathrm{c},j})\frac{\partial \mathcal{Q}^i_{\mathrm{c},j}}{\partial \boldsymbol{e}_{\mathrm{x},l}} + 2\frac{\lambda_{\mathrm{x}}}{|\mathcal{I}_{\mathrm{x}}|}\sum_{\mathcal{I}_{\mathrm{x}}} k^i_{\mathrm{x},j}(k^i_{\mathrm{x},j} - s^i_{\mathrm{x},j})\frac{\partial \mathcal{Q}^i_{\mathrm{x},j}}{\partial \boldsymbol{e}_{\mathrm{x},l}} \qquad \forall l$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{e}_{\mathrm{z},l}} = 2\frac{\lambda_{\mathrm{c}}}{|\mathcal{I}_{\mathrm{c}}|}\sum_{\mathcal{I}_{\mathrm{c}}} k^i_{\mathrm{c},j}(k^i_{\mathrm{c},j} - s^i_{\mathrm{c},j})\frac{\partial \mathcal{Q}^i_{\mathrm{c},j}}{\partial \boldsymbol{e}_{\mathrm{z},l}} + 2\frac{\lambda_{\mathrm{z}}}{|\mathcal{I}_{\mathrm{z}}|}\sum_{\mathcal{I}_{\mathrm{z}}} k^i_{\mathrm{z},j}(k^i_{\mathrm{z},j} - s^i_{\mathrm{z},j})\frac{\partial \mathcal{Q}^i_{\mathrm{z},j}}{\partial \boldsymbol{e}_{\mathrm{z},l}} \qquad \forall l.$$

The gradients of the auxiliary variables can be found as

$$\begin{aligned}
\frac{\partial \mathcal{Q}^i_{\mathrm{c},j}}{\partial \boldsymbol{e}_{\mathrm{x},l}} &= -\frac{2\delta^l_i(\boldsymbol{e}_{\mathrm{x},i} - \boldsymbol{e}_{\mathrm{z},j})}{\sigma^2_{\mathrm{e}}} && \forall l \\
\frac{\partial \mathcal{Q}^i_{\mathrm{x},j}}{\partial \boldsymbol{e}_{\mathrm{x},l}} &= -\frac{2(\delta^l_i - \delta^l_j)(\boldsymbol{e}_{\mathrm{x},i} - \boldsymbol{e}_{\mathrm{x},j})}{\sigma^2_{\mathrm{e}}} && \forall l \\
\frac{\partial \mathcal{Q}^i_{\mathrm{c},j}}{\partial \boldsymbol{e}_{\mathrm{z},l}} &= -\frac{2\delta^l_j(\boldsymbol{e}_{\mathrm{z},j} - \boldsymbol{e}_{\mathrm{x},i})}{\sigma^2_{\mathrm{e}}} && \forall l \\
\frac{\partial \mathcal{Q}^i_{\mathrm{z},j}}{\partial \boldsymbol{e}_{\mathrm{z},l}} &= -\frac{2(\delta^l_i - \delta^l_j)(\boldsymbol{e}_{\mathrm{z},i} - \boldsymbol{e}_{\mathrm{z},j})}{\sigma^2_{\mathrm{e}}} && \forall l,
\end{aligned}$$

where δ^l_i is 1 if $i = l$ and 0 otherwise. Due to the orthonormality constraints, the embedding coordinates of each domain are defined on a Stiefel manifold (i.e., $\mathcal{S}(R,N) = \{\mathbf{E} \in \mathbb{R}^{R\times N} \colon \mathbf{E}\mathbf{E}^\top = \mathbf{I}_R\}$). In order to satisfy these constraints, we need to use the modified gradient defined for Stiefel manifolds to update the embedding coordinates and to project the updated values back to the manifold using a QR decomposition [17].

When learning the kernel width, we need to operate on the logarithmic scale to satisfy the non-negativity constraint. We introduce a new variable for the logarithm of the kernel width (i.e., $\eta_{\mathrm{e}} = \log \sigma_{\mathrm{e}}$) and perform gradient-based optimization on this variable. The gradient of $\mathcal{L}$ with respect to η_{e} are

$$\frac{\partial \mathcal{L}}{\partial \eta_{\mathrm{e}}} = 2\frac{\lambda_{\mathrm{c}}}{|\mathcal{I}_{\mathrm{c}}|}\sum_{\mathcal{I}_{\mathrm{c}}} k^i_{\mathrm{c},j}(k^i_{\mathrm{c},j} - s^i_{\mathrm{c},j})\frac{\partial \mathcal{Q}^i_{\mathrm{c},j}}{\partial \eta_{\mathrm{e}}} + 2\frac{\lambda_{\mathrm{x}}}{|\mathcal{I}_{\mathrm{x}}|}\sum_{\mathcal{I}_{\mathrm{x}}} k^i_{\mathrm{x},j}(k^i_{\mathrm{x},j} - s^i_{\mathrm{x},j})\frac{\partial \mathcal{Q}^i_{\mathrm{x},j}}{\partial \eta_{\mathrm{e}}} + 2\frac{\lambda_{\mathrm{z}}}{|\mathcal{I}_{\mathrm{z}}|}\sum_{\mathcal{I}_{\mathrm{z}}} k^i_{\mathrm{z},j}(k^i_{\mathrm{z},j} - s^i_{\mathrm{z},j})\frac{\partial \mathcal{Q}^i_{\mathrm{z},j}}{\partial \eta_{\mathrm{e}}},$$

where the gradients of the auxiliary variables are found as

$$\begin{aligned}
\frac{\partial \mathcal{Q}^i_{\mathrm{c},j}}{\partial \eta_{\mathrm{e}}} &= \frac{2\|\boldsymbol{e}_{\mathrm{x},i} - \boldsymbol{e}_{\mathrm{z},j}\|^2_2}{\sigma^2_{\mathrm{e}}} \\
\frac{\partial \mathcal{Q}^i_{\mathrm{x},j}}{\partial \eta_{\mathrm{e}}} &= \frac{2\|\boldsymbol{e}_{\mathrm{x},i} - \boldsymbol{e}_{\mathrm{x},j}\|^2_2}{\sigma^2_{\mathrm{e}}} \\
\frac{\partial \mathcal{Q}^i_{\mathrm{z},j}}{\partial \eta_{\mathrm{e}}} &= \frac{2\|\boldsymbol{e}_{\mathrm{z},i} - \boldsymbol{e}_{\mathrm{z},j}\|^2_2}{\sigma^2_{\mathrm{e}}}.
\end{aligned}$$

Our complete algorithm is an alternating optimization scheme consisting of three main steps: (i) update $\mathbf{E}_{\mathrm{x}}$ given $\mathbf{E}_{\mathrm{z}}$ and σ_e, (ii) update $\mathbf{E}_{\mathrm{z}}$ given $\mathbf{E}_{\mathrm{x}}$ and σ_e, and (iii) update σ_e given $\mathbf{E}_{\mathrm{x}}$ and $\mathbf{E}_{\mathrm{z}}$. The optimization procedure sequentially updates the decision variables until convergence, which can be checked by monitoring the objective function value. The key issue for faster convergence is to select the step sizes of the update equations carefully. We use Armijo's rule, which is a line search method whose search procedure allows backtracking and does not use any curve fitting method, to speed up the convergence. Our algorithm is guaranteed to converge to one of the local optima in a finite number of iterations because there is no chance of increasing the objective value due to the line search.

The main motivation of approximating cross-domain interactions and within-domain similarities with Gaussian kernels in the embedding space is to capture local neighborhood information with the help of nonlinearity of the kernel. It is not easy to capture such information with distance-based strategies (e.g., using Euclidean distance). Some MDS variants integrate weight terms in their objective functions to ignore very large distances or dissimilarities in their learning phase [3], which is implicitly performed in the Gaussian kernel.

3 EXTENSION FOR OUT-OF-SAMPLE EMBEDDING

Our algorithm outlined in the previous section is not able to embed unseen objects, which are not used during training. We also formulate a variant of our algorithm to be able to do out-of-sample embedding. Instead of modeling the embedding coordinates as decision variables in our optimization problem, we can assume linear projections from the input domains to the embedding domain and optimize the projection matrices. The embedding coordinates are formulated as

$$\begin{aligned}
\boldsymbol{e}_{\mathrm{x},i} &= \mathbf{Q}^\top_{\mathrm{x}}\boldsymbol{x}_i \qquad \forall i \\
\boldsymbol{e}_{\mathrm{z},i} &= \mathbf{Q}^\top_{\mathrm{z}}\boldsymbol{z}_i \qquad \forall i,
\end{aligned}$$

where we assume that the objects from the two domains have vectorial representations (i.e., $\mathcal{X} \in \mathbb{R}^{D_{\mathrm{x}}}$ and $\mathcal{Z} \in \mathbb{R}^{D_{\mathrm{z}}}$). The modified optimization problem is

$$\begin{aligned}
&\text{minimize} && \mathcal{L} \\
&\text{with respect to} && \mathbf{Q}_{\mathrm{x}} \in \mathbb{R}^{D_{\mathrm{x}}\times R},\ \mathbf{Q}_{\mathrm{z}} \in \mathbb{R}^{D_{\mathrm{z}}\times R},\ \sigma_{\mathrm{e}} \in \mathbb{R}_{++} \\
&\text{subject to} && \mathbf{Q}^\top_{\mathrm{x}}\mathbf{Q}_{\mathrm{x}} = \mathbf{I}_R,\ \mathbf{Q}^\top_{\mathrm{z}}\mathbf{Q}_{\mathrm{z}} = \mathbf{I}_R,
\end{aligned}$$

where we assume orthonormality of the projection matrix columns in each domain separately.

We can use the same optimization strategy, but this time we need the gradients of $\mathcal{L}$ with respect to the projection matrices. These gradients can be calculated as

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{q}_{\mathrm{x},p}} = 2\frac{\lambda_{\mathrm{c}}}{|\mathcal{I}_{\mathrm{c}}|}\sum_{\mathcal{I}_{\mathrm{c}}} k^i_{\mathrm{c},j}(k^i_{\mathrm{c},j} - s^i_{\mathrm{c},j})\frac{\partial \mathcal{Q}^i_{\mathrm{c},j}}{\partial \boldsymbol{q}_{\mathrm{x},p}} + 2\frac{\lambda_{\mathrm{x}}}{|\mathcal{I}_{\mathrm{x}}|}\sum_{\mathcal{I}_{\mathrm{x}}} k^i_{\mathrm{x},j}(k^i_{\mathrm{x},j} - s^i_{\mathrm{x},j})\frac{\partial \mathcal{Q}^i_{\mathrm{x},j}}{\partial \boldsymbol{q}_{\mathrm{x},p}} \qquad \forall p$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{q}_{\mathrm{z},p}} = 2\frac{\lambda_{\mathrm{c}}}{|\mathcal{I}_{\mathrm{c}}|}\sum_{\mathcal{I}_{\mathrm{c}}} k^i_{\mathrm{c},j}(k^i_{\mathrm{c},j} - s^i_{\mathrm{c},j})\frac{\partial \mathcal{Q}^i_{\mathrm{c},j}}{\partial \boldsymbol{q}_{\mathrm{z},p}} + 2\frac{\lambda_{\mathrm{z}}}{|\mathcal{I}_{\mathrm{z}}|}\sum_{\mathcal{I}_{\mathrm{z}}} k^i_{\mathrm{z},j}(k^i_{\mathrm{z},j} - s^i_{\mathrm{z},j})\frac{\partial \mathcal{Q}^i_{\mathrm{z},j}}{\partial \boldsymbol{q}_{\mathrm{z},p}} \qquad \forall p,$$

where the gradients of the auxiliary variables are found as

$$\frac{\partial \mathcal{Q}^i_{\mathrm{c},j}}{\partial \boldsymbol{q}_{\mathrm{x},p}} = -\frac{2\boldsymbol{x}_i(\boldsymbol{q}^\top_{\mathrm{x},p}\boldsymbol{x}_i - \boldsymbol{q}^\top_{\mathrm{z},p}\boldsymbol{z}_j)}{\sigma^2_{\mathrm{e}}} \qquad \forall p$$

$$\frac{\partial \mathcal{Q}^i_{\mathrm{x},j}}{\partial \boldsymbol{q}_{\mathrm{x},p}} = -\frac{2(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{q}^\top_{\mathrm{x},p}\boldsymbol{x}_i - \boldsymbol{q}^\top_{\mathrm{x},p}\boldsymbol{x}_j)}{\sigma^2_{\mathrm{e}}} \quad \forall p$$
$$\frac{\partial \mathcal{Q}^i_{\mathrm{c},j}}{\partial \boldsymbol{q}_{\mathrm{z},p}} = -\frac{2\boldsymbol{z}_j(\boldsymbol{q}^\top_{\mathrm{z},p}\boldsymbol{z}_j - \boldsymbol{q}^\top_{\mathrm{x},p}\boldsymbol{x}_i)}{\sigma^2_{\mathrm{e}}} \quad \forall p$$
$$\frac{\partial \mathcal{Q}^i_{\mathrm{z},j}}{\partial \boldsymbol{q}_{\mathrm{z},p}} = -\frac{2(\boldsymbol{z}_i - \boldsymbol{z}_j)(\boldsymbol{q}^\top_{\mathrm{z},p}\boldsymbol{z}_i - \boldsymbol{q}^\top_{\mathrm{z},p}\boldsymbol{z}_j)}{\sigma^2_{\mathrm{e}}} \quad \forall p.$$

The linear projection formulation is quite restrictive because it assumes that we have vectorial representations for the objects of each domain. Instead, we can use the within-domain similarity functions to represent the objects in a vectorial form, which is known as *empirical kernel map* [22]:

$$\boldsymbol{x}_i = \begin{bmatrix} s^i_{\mathrm{x},1} & s^i_{\mathrm{x},2} & \dots & s^i_{\mathrm{x},N_\mathrm{x}} \end{bmatrix}^\top \quad \forall i$$
$$\boldsymbol{z}_i = \begin{bmatrix} s^i_{\mathrm{z},1} & s^i_{\mathrm{z},2} & \dots & s^i_{\mathrm{z},N_\mathrm{z}} \end{bmatrix}^\top \quad \forall i.$$

Even if we have vectorial representations for the objects, this strategy allows us to introduce nonlinearity into the embedding part as in kernel-based dimensionality reduction [21].

Most embedding algorithms can not handle unseen data points. Using parametric projection rules enables us to project unseen data points. In addition to out-of-sample embedding, this projection matrices can also be used for extracting feature importances. For example, if a particular row of $\mathbf{Q}_\mathrm{x}$ or $\mathbf{Q}_\mathrm{z}$ has values very close to zero, this means that the corresponding feature/sample is not important for the task at hand.

4 EXPERIMENTS

To show the performance of our algorithm MKPE, we test it on two tasks: (i) modeling biological interaction networks and (ii) cross-domain information retrieval. We implement our algorithms in Matlab, which is publicly available at https://github.com/mehmetgonen/mkpe/. We set the regularization parameters (λ_c, λ_x, λ_z) to $(1, 0.1, 0.1)$.

We use two different drug–protein interaction networks provided by [25], which are considering *G-protein-coupled receptors* (GPCR) and *nuclear receptors* (NR) from humans and are publicly available at http://web.kuicr.kyoto-u.ac.jp/supp/yoshi/drugtarget/. Table 1 summarizes the data sets in terms of numbers of drugs, proteins, and interactions, which contain both the within-domain similarity scores and the experimentally validated interactions.

Table 1. The drug–protein interaction data sets provided by [25].

Data Set	Number of Drugs	Number of Proteins	Number of Interactions
GPCR	223	95	635
NR	54	26	90

We cast the problem of modeling drug–protein interaction networks into our formulation as follows: The two domains $\mathcal{X}$ and $\mathcal{Z}$ correspond to drugs and proteins, respectively. The cross-domain interactions correspond to the given set of experimentally validated drug–protein interactions, which are usually represented in the form of a binary matrix (i.e., 1 for the interacting pairs and 0 for the non-interacting pairs). We construct our cross-domain interaction score from this binary interaction matrix with the following simple rule:

$$s^i_{\mathrm{c},j} = \begin{cases} 0.9 & \text{if } \boldsymbol{x}_i \text{ and } \boldsymbol{z}_j \text{ are interacting,} \\ \mathtt{NA} & \text{otherwise,} \end{cases}$$

where we set the interaction score to 0.9 for the interacting pairs because setting the score to 1 implies that their ideal embedding coordinates are equal, which is not a good idea for visualization. We leave the interaction score empty for the noninteracting pairs because some of them may be interacting in reality but not validated experimentally yet and setting the score to a low value may hurt the visualization.

The within-domain similarity score between drugs is found by representing them as graphs and calculating the Jaccard similarity coefficient over the substructures of the two graphs [10]. Given two drugs $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, chemical similarity between them can be found as

$$s^i_{\mathrm{x},j} = \frac{|\boldsymbol{x}_i \cap \boldsymbol{x}_j|}{|\boldsymbol{x}_i \cup \boldsymbol{x}_j|}.$$

The within-domain similarity score between proteins is found using a normalized version of Smith-Waterman score [24]. Given two proteins $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$, genomic similarity between them can be found as

$$s^i_{\mathrm{z},j} = \frac{\mathtt{SW}(\boldsymbol{z}_i, \boldsymbol{z}_j)}{\sqrt{\mathtt{SW}(\boldsymbol{z}_i, \boldsymbol{z}_i)\mathtt{SW}(\boldsymbol{z}_j, \boldsymbol{z}_j)}},$$

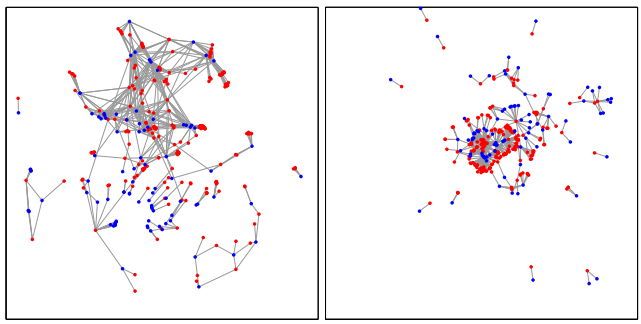
where $\mathtt{SW}(\cdot, \cdot)$ gives the canonical Smith-Waterman score between two proteins. Note that our choice of approximating both within-domain similarity scores using the Gaussian kernel is reasonable because they are guaranteed to take values between 0 and 1 similar to the Gaussian kernel.

In the first set of experiments, we project drugs and proteins into a unified two-dimensional (2-D) embedding space using our algorithm MKPE and *co-occurrence data embedding* (CODE) algorithm of [6]. For MKPE algorithm, we perform 100 iterations. CODE algorithm uses the co-occurrence statistics of objects to embed them into a unified embedding space. We provide the cross-domain interaction scores and within-domain similarities as the co-occurrence statistics and use the same values used as the regularization weights in MKPE for the cross- and within-domain likelihood weights in CODE. We use the Matlab implementation of CODE provided by [6] with its default parameters.

Figures 1 and 2 show the 2-D embeddings obtained by CODE and MKPE algorithms on the GPCR and NR data sets, respectively. We can see that MKPE finds more visually appealing embeddings than CODE on both data sets because MKPE has well-separated groups of nodes and fewer edge crossings compared to CODE.

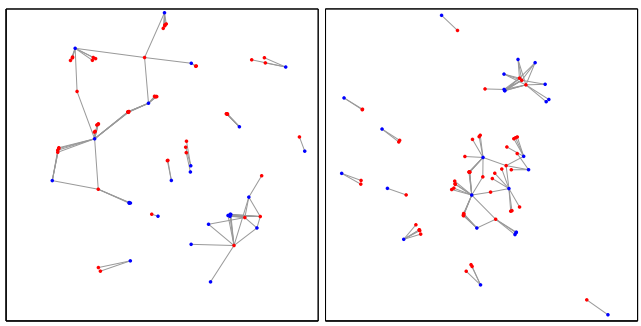
In addition to visual attractiveness, we also compare the algorithms in terms of their performances on unknown interaction prediction task. The drug–protein interactions we use are extracted by [25] from an earlier version of KEGG DRUG database [12]. Its latest online version or other databases may contain additional experimentally validated drug–protein interactions. On the NR data set, we rank the noninteracting pairs with respect to their Euclidean distances in the embedding space and extract the pairs with the five smallest distances. We check these interactions from the latest online versions of ChEMBL [5], DrugBank [14], and KEGG DRUG [12].

Table 2 lists the top five predicted interactions obtained by both algorithms on the NR data set. We see that the first four predictions of MKPE (marked with $\star$ in Table 2) are reported in at least one of the databases, whereas none of the predictions obtained by CODE. Note that these results are obtained using only two dimensions and this is a strong evidence for the practical relevance of our method.



(a) CODE (b) MKPE

Figure 1. The two-dimensional embeddings obtained on the GPCR data set. Red and blue points denote drugs and proteins, respectively.



(a) CODE (b) MKPE

Figure 2. The two-dimensional embeddings obtained on the NR data set. Red and blue points denote drugs and proteins, respectively.

Table 2. The top five predicted interactions obtained by CODE and MKPE algorithms on the NR data set.

	Rank	Pair	Rank	Pair
CODE	1	D00506 hsa:6095	1⋆	D01115 hsa:2908
	2	D00279 hsa:6095	2⋆	D00443 hsa:367
	3	D00565 hsa:6095	3⋆	D00443 hsa:2908
	4	D05341 hsa:2104	4⋆	D00075 hsa:5241
	5	D05341 hsa:2103	5	D00961 hsa:2101
MKPE				

In the second set of experiments, we illustrate the performance of our variant for out-of-sample embedding in predicting interactions for unseen drugs. For both data sets, we apply ten replications of ten-fold cross-validation over drugs to obtain robust results. We compare our algorithm with *kernelized Bayesian matrix factorization with twin kernels* (KBMF2K) algorithm of [7], which is proposed for modeling biological interaction networks and projects objects from different domains into a unified embedding space. We use the Matlab implementation of KBMF2K provided by [7] with its default parameters. We obtain the results of both methods by training them with changing subspace dimensionality parameters taken from $\{5, 10, 15, 20, 25\}$.

Figure 3 gives the average AUC (area under the receiver operating curve) values for KBMF2K and MKPE. When the subspace dimensionality is larger than ten, we see that MKPE achieves comparable average AUC values on the GPCR data set, whereas it significantly

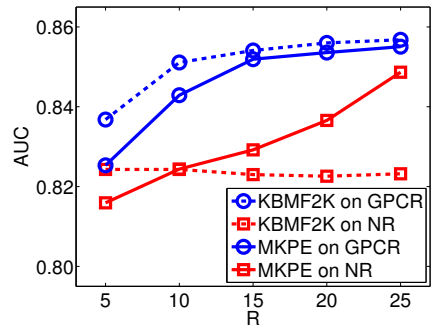


Figure 3. The prediction performances of KBMF2K and MKPE with changing subspace dimensionality on the GPCR and NR data sets in terms of average AUC values.

improves the results on the NR data set. These results validate the predictive performance of MKPE for out-of-sample embedding.

We perform cross-domain information retrieval experiments on an image classification data set provided by [20], which is publicly available at `http://www1.icsi.berkeley.edu/~saenko/projects.html#DA`. The classification task is to assign images to one of the 31 categories (e.g., backpack, bicycle, helmet, chair, etc.). The data points come from two domains: (i) images taken with a high-resolution DSLR camera (`dslr`) and (ii) images taken with a low-resolution webcam (`webcam`). Each category has images from five distinct objects (e.g., different backpacks). The `dslr` and `webcam` domains have 423 and 795 images, respectively. [20] use a codebook of size 800 to convert all images into histograms over visual words. Note that no spatial or color information is included in the image representation.

Following the experimental procedure of [20], we investigate domain transfer from the high-resolution DSLR images (i.e., source domain) to the low-resolution webcam images (i.e., target domain). Each category has eight training points in the source domain, whereas we have only three for the target domain. Training images are selected from the first three objects of each category and test images are selected from the remaining two.

We cast this cross-domain information retrieval task into our formulation as follows: The two domains $\mathcal{X}$ and $\mathcal{Z}$ correspond to `dslr` and `webcam`, respectively. We construct our cross-domain interaction score from the training data as

$$s^i_{c,j} = \begin{cases} 1 & \text{if } \boldsymbol{x}_i \text{ and } \boldsymbol{z}_j \text{ belong to the same object,} \\ 0.9 & \text{if } \boldsymbol{x}_i \text{ and } \boldsymbol{z}_j \text{ belong to the same class,} \\ 0 & \text{otherwise.} \end{cases}$$

The within-domain similarity scores are calculated as cosine similarities between image representations.

We train our variant for out-of-sample embedding by performing 100 iterations to classify unseen images from the target domain. We assign unseen images to the category of their nearest neighbors in the embedding space found by MKPE. We perform ten replications for each subspace dimensionality from $\{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$. We compare our algorithm with three baseline algorithms: (i) k-nearest neighbor (k-NN) classifier using only target domain (i.e., k-NN on $\mathcal{Z}$), (ii) k-NN classifier using only source domain (i.e., k-NN on $\mathcal{X}$), and (iii) k-NN classifier using both source and target domains (i.e., k-NN on $\mathcal{X}$+$\mathcal{Z}$). For baseline methods, we also set k to 1.

Figure 4 gives the classification performances of baseline algorithms and MKPE in terms of average test accuracy. We see that us-

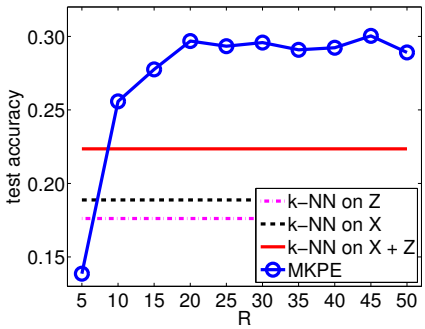


Figure 4. The classification performances of MKPE with changing subspace dimensionality and baseline methods in terms of average test accuracy.

ing only target domain (i.e., k-NN on $\mathcal{Z}$) gets the worst results due to small number of training samples per category. Using only source domain (i.e., k-NN on $\mathcal{X}$) or both domains (i.e., k-NN on $\mathcal{X}$+$\mathcal{Z}$) improves the classification performance. MKPE outperforms all baseline methods when the subspace dimensionality is larger than five. The performance of MKPE stabilizes after 20 dimensions and it is better than k-NN on $\mathcal{X}$+$\mathcal{Z}$ around seven per cent. These results show that our method is also useful for domain adaptation (i.e., transfer learning) tasks such as cross-domain information retrieval.

5 CONCLUSIONS

In this paper, we introduce a novel embedding algorithm, called multiple kernel preserving embedding, for heterogeneous data. Our method allows us to map objects from different domains into a unified embedding space by preserving both cross-domain interactions and within-domain similarities, which are approximated with Gaussian kernels. Using these nonlinear kernels in the embedding space transfers local neighborhood information from the provided interactions and similarities. We also extend our formulation for out-of-sample embedding using parametric projection rules. Experimental results on two unrelated tasks, namely, modeling biological interaction networks and cross-domain information retrieval, show wide applicability of our model.

ACKNOWLEDGEMENTS

This study was financially supported by the Integrative Cancer Biology Program of the National Cancer Institute (grant no 1U54CA149237).

REFERENCES

[1] M. M. Borenstein, A. M. Borenstein, F. Michel, and N. Paragios, 'Data fusion through cross-modality metric learning using similarity-sensitive hashing', in *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition*, (2010).

[2] J. Choo, S. Bohn, G. Nakamura, A. M. White, and H. Park, 'Heterogeneous data fusion via space alignment using nonmetric multidimensional scaling', in *Proceedings of the 12th SIAM International Conference on Data Mining*, (2012).

[3] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling*, Chapman & Hall/CRC, London, 2000.

[4] M. D. Dyer, T. M. Murali, and B. W. Sobral, 'Computational prediction of host–pathogen protein–protein interactions', *Bioinformatics*, **23**, i159–i166, (2007).

[5] A. Gaulton, L. J. Bellis, A. P. Bento, J. Chambers, M. Davies, et al., 'ChEMBL: A large-scale bioactivity database for drug discovery', *Nucleic Acids Research*, **40**, D1100–D1107, (2012).

[6] A. Globerson, G. Chechik, F. Pereira, and N. Tishby, 'Euclidean embedding of co-occurrence data', *Journal of Machine Learning Research*, **8**, 2265–2295, (2007).

[7] M. Gönen, 'Predicting drug–target interactions from chemical and genomic kernels using Bayesian matrix factorization', *Bioinformatics*, **28**, 2304–2310, (2012).

[8] M. Gönen and E. Alpaydın, 'Multiple kernel learning algorithms', *Journal of Machine Learning Research*, **12**, 2211–2268, (2011).

[9] Y. Guo, J. Gao, and P. W. Kwan, 'Twin kernel embedding', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**, 1490–1495, (2008).

[10] M. Hattori, Y. Okuno, S. Goto, and M. Kanehisa, 'Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways', *Journal of the American Chemical Society*, **125**, 11853–11865, (2003).

[11] H. Hotelling, 'Relations between two sets of variants', *Biometrika*, **28**, 321–377, (1936).

[12] M. Kanehisa, S. Goto, Y. Sato, M. Furumichi, and M. Tanabe, 'KEGG for integration and interpretation of large-scale molecular data sets', *Nucleic Acids Research*, **40**, D109–D114, (2012).

[13] H. Kashima, Y. Yamanishi, T. Kato, M. Sugiyama, and K. Tsuda, 'Simultaneous inference of biological networks of multiple species from genome-wide data and evolutionary information: A semi-supervised approach', *Bioinformatics*, **25**, 2962–2968, (2009).

[14] C. Knox, V. Law, T. Jewison, P. Liu, S. Ly, et al., 'DrugBank 3.0: A comprehensive resource for 'omics' research on drugs', *Nucleic Acids Research*, **39**, D1035–D1041, (2011).

[15] J. Kruskal, 'Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis', *Psychometrika*, **29**, 1–27, (1964).

[16] S. Kumar and R. Udupa, 'Learning hash functions for cross-view similarity search', in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, (2011).

[17] J. H. Manton, 'Optimization algorithms exploiting unitary constraints', *IEEE Transactions on Signal Processing*, **50**, 635–650, (2002).

[18] N. Quadrianto and C. H. Lampert, 'Learning multi-view neighborhood preserving projections', in *Proceedings of the 28th International Conference on Machine Learning*, (2011).

[19] N. Rasiwasia, J. C. Pereira, E. Coviello, and G. Doyle, 'A new approach to cross-modal multimedia retrieval', in *Proceedings of the 18th International Conference on Multimedia*, (2010).

[20] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, 'Adapting visual category models to new domains', in *Proceedings of the 11th European Conference on Computer Vision*, (2010).

[21] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, MA, 2002.

[22] *Kernel Methods in Computational Biology*, eds., B. Schölkopf, K. Tsuda, and J.-P. Vert, MIT Press, Cambridge, MA, 2004.

[23] X. Shi, Q. Liu, W. Fan, P. S. Yu, and R. Zhu, 'Transfer learning on heterogenous feature spaces via spectral transformation', in *Proceedings of the 10th IEEE International Conference on Data Mining*, (2010).

[24] T. F. Smith and M. S. Waterman, 'Identification of common molecular subsequences', *Journal of Molecular Biology*, **147**, 195–197, (1981).

[25] Y. Yamanishi, M. Araki, A. Gutteridge, W. Honda, and M. Kanesiha, 'Prediction of drug–target interaction networks from the integration of chemical and genomic spaces', *Bioinformatics*, **24**, i232–i240, (2008).

[26] Y. Yamanishi, M. Kotera, M. Kanesiha, and S. Goto, 'Drug–target interaction prediction from chemical, genomic and pharmacological data in an integrated framework', *Bioinformatics*, **26**, i246–i254, (2010).

[27] X. Zhai, Y. Peng, and J. Xiao, 'Effective heterogeneous similarity measure with nearest neighbors for cross-media retrieval', in *Proceedings of the 18th International Conference on Advances in Multimedia Modeling*, (2012).

[28] D. Zhang, F. Wang, and L. Si, 'Composite hashing with multiple information sources', in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, (2011).

[29] Y. Zhen and D.-Y. Yeung, 'A probabilistic model for multimodal hash function learning', in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2012).