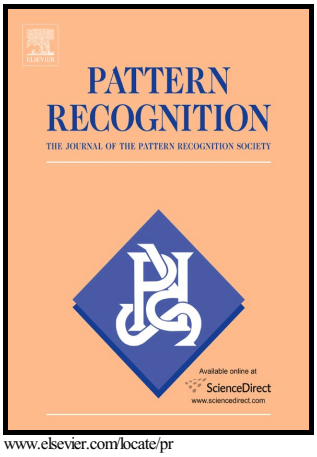
Computation of Heterogeneous Object Co-embeddings from Relational Measurements

DOI:
10.1016/j.patcog.2016.12.004

Document Version
Accepted author manuscript



www.elsevier.com/locate/pr

Computation of Heterogeneous Object Co-embeddings from Relational Measurements

Yu Wu, Tingting Mu, Panos Liatsis, John Y. Goulermas

PII: S0031-3203(16)30390-9
DOI: http://dx.doi.org/10.1016/j.patcog.2016.12.004
Reference: PR5976

To appear in: *Pattern Recognition*

Received date: 24 April 2016
Revised date: 30 September 2016
Accepted date: 4 December 2016

Cite this article as: Yu Wu, Tingting Mu, Panos Liatsis and John Y. Goulermas, Computation of Heterogeneous Object Co-embeddings from Relationa Measurements, *Pattern Recognitior* http://dx.doi.org/10.1016/j.patcog.2016.12.004

Computation of Heterogeneous Object Co-embeddings from Relational Measurements

Yu Wu[a], Tingting Mu[b], Panos Liatsis[c], John Y. Goulermas[a]

[a]*School of Electrical Engineering Electronics and Computer Science, University of Liverpool, Ashton Building, Liverpool, L69 3BX, UK*

[b]*School of Computer Science, University of Manchester, Kilburn Building, Manchester, M13 9PL, UK*

[c]*Department of Electrical Engineering, The Petroleum Institute, Abu Dhabi, Ruwais Building, PO Box 2533, UAE*

Abstract

Dimensionality reduction and data embedding methods generate low dimensional representations of a single type of homogeneous data objects. In this work, we examine the problem of generating co-embeddings or pattern representations from two different types of objects within a joint common space of controlled dimensionality, where the only available information is assumed to be a set of pairwise relations or similarities between instances of the two groups. We propose a new method that models the embedding of each object type symmetrically to the other type, subject to flexible scale constraints and weighting parameters. The embedding generation relies on an efficient optimization despatched using matrix decomposition, that is also extended to support multidimensional co-embeddings. We also propose a scheme of heuristically reducing the parameters of the model, and a simple way of measuring the conformity between the original object relations and the ones re-estimated from the co-embeddings, in order to achieve model selection by identifying the optimal model parameters with a simple search procedure. The capabilities of the proposed method are demonstrated with multiple synthetic and real-world datasets from the text mining domain. The experimental results and comparative analyses indicate that the proposed algorithm outperforms existing methods for co-embedding generation.

Keywords: co-embedding generation, relational information, heterogeneous object analysis, joint space projection.

Email addresses: yu.wu@liverpool.ac.uk (Yu Wu), tingtingmu@me.com (Tingting Mu), pliatsis@pi.ac.ae (Panos Liatsis), j.y.goulermas@liverpool.ac.uk (John Y. Goulermas)

1. Introduction

Methods for the generation of embeddings or pattern representations of data objects in low-dimensional spaces have received significant attention, as they are very important for both unsupervised and supervised machine learning as well as information visualization. Over the years, such methods have continually progressed towards the ability to capture and analyze the structure and latent characteristics of larger and more complex datasets.

Given objects characterized by high-dimensional features, various dimensionality reduction approaches can be employed to learn the low-dimensional representation of these objects. Examples include the classic Principal Components Analysis [1], which is a linear dimensionality reduction and decorrelation technique that maximizes the variance of the projected patterns in the low-dimensional space. Locality Preserving Projections [2] is another linear embedding method, but projects the data to preserve a certain affinity graph constructed from the data pattern similarities. A popular nonlinear alternative is the Locally Linear Embedding (LLE) [3], which recovers global nonlinear structure from local linear neighbor fits. Several variations of these classical projection and embedding approaches have been developed to capture more precisely the structure of the data. Examples include the Multi-Manifold LLE for processing multi-class data [4], versions of discriminant embedding generation [5, 6], and projection methods for processing multimodal data [7]. Moreover, recent advances in deep learning have enabled the learning of low-dimensional representations of objects through mapping functions constructed with neural networks, such as deep semi-supervised embedding [8].

Given objects characterized by distance information, Mutidimensional Scaling [9] can be used to preserve the pairwise distances of the original patterns in the low-dimensional space. When link information is made available between objects, e.g., when representing objects by a knowledge graph, their low-dimensional representations can be learned by the embedding-driven relational learning algorithms that support the processing of link validities [10, 11].

All the above techniques only embed homogeneous (i.e., of a single type) data objects into a low-dimensional space given their higher dimensional feature representations or the relation/distance/link information between them. In many real-world applications, it is important to simultaneously handle heterogeneous types of data, such as genes and symptoms, documents and words or images, review articles from different domains, etc., by mapping them into a single common space.

Various data processing methods have been proposed to address the problem of handling heterogeneous types of data. Examples include methods targeting specific applications, such as biological networks [12], [13], semantic analysis [14], [15] and information retrieval [16], [17]. Heterogeneous data analysis has also been performed by more generic methods. For instance, Correspondence Analysis [18] represents the rows and columns of a data matrix as points in a space of low-dimensionality. Latent Semantic Indexing [19] is a popular information retrieval embedding method, frequently used to embed documents and words

in a common space [20]. Canonical Correlation Analysis [21] attempts to maximize the correlation between two sets of measurements. Similarly, variations of nonmetric Mutidimensional Scaling [14] have been used to place the corresponding reference data as close as possible, so that the patterns are aligned in the common space. More recent methods [22] can learn the joint representation from multiple datasets that lie on multiple manifolds. However, most of these techniques require the availability of pattern information from the different data representations.

The heterogeneous embedding problem considered in this work, only assumes the existence of a relational similarity matrix between two sets of objects of possibly differing cardinality. This is also known as joint embedding or co-embedding [23, 16, 24]. The goal is to generate co-embeddings, where both groups of objects are embedded in a joint space. Various stochastic methods have been previously proposed to achieve this, such as Parametric Embedding [23], Co-occurrence Data Embedding [25], Bayesian Co-occurrence Data Embedding [16], as well as a dynamic embedding model that processes a sequence of co-occurrence data changing over time [26]. These algorithms treat the co-occurrence object pairs as being generated by a Gaussian mixture in the embedding space, and then recover the embedding that maximizes the likelihood of the observed data. An alternative strategy for computing co-embeddings from similarities between heterogeneous objects is Automatic Co-embedding with Adaptive Shaping [24] based on matrix factorization, which generalizes ideas from embedding algorithms such as [19], [18], [27], [28], and controls the factors that generate different shapes and distributions of column and row objects in the common space. There are also methods that are specialized at learning embeddings from a binary relation matrix between two groups of objects. For instance, Maximum-Margin Matrix Factorization [29] attempts to fit a binary target matrix with a low-rank inner product matrix between the embedding vectors of the row and column objects. Another method estimates the data distribution of the row and column objects from binary co-occurrence data using a Deep Embedding Model [30].

In this paper, to generate heterogeneous patterns into a unified embedding space, we propose a new method that models the embedding of each group with respect to the other group using suitable weightings. We only assume availability of the relational similarity information between representatives from each group. The co-embedding generation relies on an efficient joint model optimization based on a matrix decomposition, accompanied by heuristics that permit a drastic reduction of the scaling parameters. The proposed method is compared with state of the art methods using multiple synthetic and real-world datasets.

We organize the rest of this paper as follows. Section 2 briefly reviews some related heterogeneous embedding algorithms. In section 3, we introduce the proposed algorithm, its model, optimization scheme, as well as its parameter identification mechanism. The experimental results and comparative analyses are reported in section 4, while section 5 concludes the work.

2. Related Methods

We are given an $m \times n$ input matrix $\mathbf{R} = [r_{ij}]$, which is assumed to be non-negative and without the existence of rows or columns made entirely of zero entries. These entries represent relations (similarities) between the m (row) objects $\{x_i\}_{i=1}^m$ from group $\mathcal{X}$ and n (column) objects $\{y_j\}_{j=1}^n$ from group $\mathcal{Y}$. Such objects can be heterogeneous and are not assumed to be explicitly representable. The objective is to find a joint embedding of these objects in a common space of dimensionality k, whereby the incurred geometry reflects reasonably well the similarities between the row and column objects. We represent these heterogeneous embeddings through the $m \times k$ embedding matrix $\mathbf{Z}_x$ and the $n \times k$ matrix $\mathbf{Z}_y$, for the row and the column objects, respectively. The embedded patterns are the rows of these matrices, and correspond to the vectors $\boldsymbol{z}_i^{(x)} = [z_{1i}^{(x)}, z_{2i}^{(x)}, \ldots, z_{ki}^{(x)}]^\top$ for objects x_i, and $\boldsymbol{z}_j^{(y)} = [z_{1j}^{(y)}, z_{2j}^{(y)}, \ldots, z_{kj}^{(y)}]^\top$ for objects y_j. In the following subsections, we summarize existing algorithms to generate such heterogeneous co-embeddings.

2.1. Co-Occurrence Data Embedding (CODE)

CODE [25] is based on a statistical model which interprets ijth elements of the input matrix $\mathbf{R}$ as empirical co-occurrence frequencies. By requiring the relation matrix to satisfy either the condition $\sum_{i=1}^m \sum_{j=1}^n r_{ij} = 1$ or $\sum_{j=1}^n r_{ij} = 1$, CODE models the co-occurrence rate $p(x_i, y_j)$ to be proportional to the closeness of embedded points $\boldsymbol{z}_i^{(x)}$ and $\boldsymbol{z}_j^{(y)}$. Through Bayes' theorem, the conditional probability $\hat{p}(y_j|x_i)$ is modelled as

$$\hat{p}(y_j|x_i) \equiv \frac{1}{h(\boldsymbol{z}_i^{(x)})} p(y_j) \exp\left(-\|\boldsymbol{z}_i^{(x)} - \boldsymbol{z}_j^{(y)}\|_2^2\right), \tag{1}$$

where $h(\boldsymbol{z}_i^{(x)})$ is the normalization term, and $p(y_j)$ is the prior probability for object y_j. The degree of the correspondence between input distributions $p(y_j|x_i)$ and embedding $\hat{p}(y_j|x_i)$ is then measured using the log-likelihood function $\sum_{i=1}^m \sum_{j=1}^n p(y_j|x_i) \log \hat{p}(y_j|x_i)$, and finally, the embeddings are obtained by optimizing the underlying problem.

2.2. Bipartite Graph Partitioning (BGP)

BGP [27] models the set of heterogeneous objects, e.g., documents and words when processing a corpus, as a bipartite graph between the two types of objects, so that object co-clustering is converted to a graph partitioning problem. In its model, the $(m+n) \times (m+n)$ adjacency matrix of the graph can be expressed as

$$\mathbf{M} = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{0} \end{bmatrix}. \tag{2}$$

The co-embeddings are then calculated by solving a relaxation to the underlying normalized cut of this graph. Letting $\mathbf{D}_x$ be the $m \times m$ diagonal matrix formed

by the vector of the row sums of $\mathbf{R}$, and $\mathbf{D}_y$ the $n \times n$ diagonal matrix formed similarly by the column sums, the optimal co-embeddings are given as

$$\mathbf{Z}_x = \mathbf{D}_x^{\frac{1}{2}} \mathbf{U}_k, \tag{3}$$

$$\mathbf{Z}_y = \mathbf{D}_y^{\frac{1}{2}} \mathbf{V}_k, \tag{4}$$

where $\mathbf{U}_k$ and $\mathbf{V}_k$ are the matrices containing the left and right singular vectors matrices of $\mathbf{D}_x^{\frac{1}{2}} \mathbf{R} \mathbf{D}_y^{-\frac{1}{2}}$, corresponding to the 2nd to $(k+1)$th largest singular values.

2.3. Correspondence Analysis (CA)

CA [18] regards the input relational matrix $\mathbf{R}$ as a contingency table, such that the Euclidean distances between row (or column) objects in the embedded space are equal to the χ^2 distances between rows (or columns) in the table. If we denote by r_i the ith row sum of $\mathbf{R}$, and by c_j its jth column sum, the χ^2 distance between the ith and the kth rows can be given by

$$d_{ik}^2 = \sum_{j=1}^{n} \frac{1}{c_j} \left(\frac{r_{ij}}{r_i} - \frac{r_{kj}}{r_k} \right). \tag{5}$$

Subsequently, to preserve the row object distances, CA finds the row embedding according to

$$\mathbf{Z}_x = \mathbf{D}_x^{-\frac{1}{2}} \mathbf{U}_k \mathbf{\Theta}_k, \tag{6}$$

and, similarly, for the column objects, the embedding matrix is given as

$$\mathbf{Z}_y = \mathbf{D}_y^{-\frac{1}{2}} \mathbf{V}_k \mathbf{\Theta}_k, \tag{7}$$

where $\mathbf{D}_x$ and $\mathbf{D}_y$ are as defined above. $\mathbf{U}_k$, $\mathbf{V}_k$ and $\mathbf{\Theta}_k$ are the matrices containing the left and right singular vectors, and the corresponding 2nd to $(k+1)$ largest singular values, respectively, of a normalized version of $\mathbf{R}$, such as $\mathbf{D}_x^{-\frac{1}{2}} \mathbf{R} \mathbf{D}_y^{-\frac{1}{2}}$.

2.4. Automatic Co-embedding with Adaptive Shaping (ACAS)

ACAS [24] is a recent matrix factorization method based on exploiting the commonalities amongst the existing models of CA, Latent Semantic Indexing and other methods proposed in [27], [28]. ACAS firstly scales the relational matrix according to

$$\hat{\mathbf{R}} = \mathbf{S}_x^{-\frac{1}{2}} \mathbf{R} \mathbf{S}_y^{-\frac{1}{2}}, \tag{8}$$

where the scaling matrices $\mathbf{S}_x$ and $\mathbf{S}_y$ are generalizations to the row sum diagonal matrix $\mathbf{D}_x$ and column sum diagonal matrix $\mathbf{D}_y$. Specifically, the ith diagonal

element $s_i^{(x)}$ of $\mathbf{S}_x$ and the jth diagonal elements $s_j^{(y)}$ of $\mathbf{S}_y$ are controlled by a model variable p as

$$s_i^{(x)} = \begin{cases} 1, & \text{if } p = 0, \\ \left(\sum_{j=1}^{n} r_{ij}^{p}\right)^{\frac{1}{p}}, & \text{if } p \geq 1, \\ \max(r_{i1}, r_{i2}, \ldots, r_{in}), & \text{if } p = \infty, \end{cases} \tag{9}$$

and

$$s_j^{(y)} = \begin{cases} 1, & \text{if } p = 0, \\ \left(\sum_{i=1}^{m} r_{ij}^{p}\right)^{\frac{1}{p}}, & \text{if } p \geq 1, \\ \max(r_{1j}, r_{2j}, \ldots, r_{mj}), & \text{if } p = \infty. \end{cases} \tag{10}$$

Likewise, the co-embeddings $\mathbf{Z}_x$ and $\mathbf{Z}_y$ are controlled by model variables $\alpha > 0$ and β via

$$\mathbf{Z}_x = \mathbf{S}_x^{-\alpha}\mathbf{U}_k\Theta_k^{\beta}, \tag{11}$$

$$\mathbf{Z}_y = \mathbf{S}_y^{-\alpha}\mathbf{V}_k\Theta_k^{\beta}, \tag{12}$$

where $\mathbf{U}_k$, $\mathbf{V}_k$ and $\mathbf{\Theta}_k$ are as defined before. Using different values for the parameters p, α and β, the method can generate a wide range of embeddings; for example, with $p = 1$, $\alpha = \frac{1}{2}$ and $\beta = 1$, we obtain the CA model, while setting $p = 0$, $\alpha = 0$ and $\beta = 1$ yields the Latent Semantic Indexing model. The optimal model is then obtained by using maximum log-likelihood and a quantized scoring function.

3. The Proposed Framework

3.1. Model Construction

We firstly consider the simpler problem of mapping the pairwise relationships contained in matrix $\mathbf{R}$ to a line. We let $\boldsymbol{z}_x = [z_{x_1}, z_{x_2}, \ldots, z_{x_m}]^\top$ and $\boldsymbol{z}_y = [z_{y_1}, z_{y_2}, \ldots, z_{y_n}]^\top$ be the maps of the m row objects $\{x_i\}_{i=1}^m$ in group $\mathcal{X}$ and the n column objects $\{y_j\}_{j=1}^n$ in $\mathcal{Y}$, respectively. Assuming that the coordinates of the embedding $\boldsymbol{z}_x$ are known, then a reasonably generic criterion for choosing a good map for the points $\boldsymbol{z}_y$ is to minimize a series of cost functions for all objects x_i, each expressed as

$$f_{x_i}(\boldsymbol{z}_y) = (z_{x_i} - z_{y_1})^2 w_{i1} + (z_{x_i} - z_{y_2})^2 w_{i2} + \ldots + (z_{x_i} - z_{y_n})^2 w_{in}. \tag{13}$$

This criterion is similar to embedding methods, such as the Laplacian Eigenmaps [31], where the distances between the embedded points are driven to correspond to those of the original patterns through similarity weights w_{ij}. In Eq.(13), the distances between the embedded $\{y_j\}_{j=1}^n$ and x_i, and the weights w_{ij} should be suitably restricted, by, for example, having $w_{ij} < w_{ik}$ when $(z_{x_i} - z_{y_j})^2 > (z_{x_i} - z_{y_k})^2$. Based on this, we can define normalized weights $w_{ij} = r_{ij} / \sum_{j=1}^{n} r_{ij}$, such that if objects y_j have high similarity to objects x_i, then their embedded counterparts z_{y_j} and z_{x_i} will be proximate.

Applying Eq.(13) to all embedded points $\{z_{x_i}\}_{i=1}^m$, generates m different minimizing functions $\{f_{x_i}(\boldsymbol{z}_y)\}_{i=1}^m$. Since the row sum $\sum_{j=1}^n r_{ij}$ is an indicator of the overall similarity level of object x_i to all objects $\{y_j\}_{j=1}^n$ within $\mathcal{Y}$, it can be taken into account in the optimization through an aggregate cost function

$$\hat{\mathbf{F}}(\boldsymbol{z}_y) = \sum_{i=1}^m \Big(\sum_{j=1}^n r_{ij} \Big)^{\eta_1} f_{x_i}(\boldsymbol{z}_y), \tag{14}$$

where $\eta_1 \geq 0$ is a parameter that controls the row sum weight $\sum_{j=1}^n r_{ij}$ which scales each objective f_{x_i}. The higher this weight is, the more emphasis is given to the minimization of the particular $f_{x_i}(\boldsymbol{z}_y)$, in order to keep the embedded points z_{y_j} close to z_{x_i}. If we then apply the above normalized weights estimated from $\mathbf{R}$ to Eq.(13) and substitute in Eq.(14) we have

$$\begin{aligned}
\hat{\mathbf{F}}(\boldsymbol{z}_y) &= \sum_{i=1}^m \Big(\sum_{j=1}^n r_{ij} \Big)^{\eta_1} \sum_{j=1}^n (z_{x_i} - z_{y_j})^2 w_{ij} \\
&= \sum_{i=1}^m \Big(\sum_{j=1}^n r_{ij} \Big)^{\eta_1} \sum_{j=1}^n (z_{x_i} - z_{y_j})^2 \frac{r_{ij}}{\sum_{j=1}^n r_{ij}} \\
&= \sum_{i=1}^m \sum_{j=1}^n (z_{x_i} - z_{y_j})^2 r_{ij}^{(x)},
\end{aligned} \tag{15}$$

where $r_{ij}^{(x)} = r_{ij} (\sum_{j=1}^n r_{ij})^{\eta_1 - 1}$. This global cost function is, however, subject to knowing the optimal $\{z_{x_i}\}_{i=1}^m$ coordinates in $\boldsymbol{z}_x$.

Reversing the above, and assuming that $\boldsymbol{z}_y$ is given and that we seek to recover $\boldsymbol{z}_x$, we can define a symmetric to $\hat{\mathbf{F}}$ aggregate cost function, as

$$\hat{\mathbf{G}}(\boldsymbol{z}_x) = \sum_{j=1}^n \sum_{i=1}^m (z_{y_j} - z_{x_i})^2 r_{ij}^{(y)}, \tag{16}$$

where $r_{ij}^{(y)} = r_{ij} (\sum_{i=1}^m r_{ij})^{\eta_2 - 1}$ and $\eta_2 \geq 0$. A trivial solution to the above is when all z_{x_i} and z_{y_j} collapse to a single coordinate, and this corresponds to $\hat{\mathbf{F}}(\boldsymbol{z}_y) = \hat{\mathbf{G}}(\boldsymbol{z}_x) = 0$. The exclusion of degenerate solutions during the optimization is discussed in Section 3.2.1.

The minimization problems in Eqs.(15,16) can be expressed in matrix forms as

$$\hat{\mathbf{F}}(\boldsymbol{z}_y) = \boldsymbol{z}_x^\top \mathbf{D}_{r,x} \boldsymbol{z}_x + \boldsymbol{z}_y^\top \mathbf{D}_{c,x} \boldsymbol{z}_y - 2 \boldsymbol{z}_x^\top \mathbf{R}_x \boldsymbol{z}_y, \tag{17}$$

$$\hat{\mathbf{G}}(\boldsymbol{z}_x) = \boldsymbol{z}_x^\top \mathbf{D}_{r,y} \boldsymbol{z}_x + \boldsymbol{z}_y^\top \mathbf{D}_{c,y} \boldsymbol{z}_y - 2 \boldsymbol{z}_x^\top \mathbf{R}_y \boldsymbol{z}_y, \tag{18}$$

where $\mathbf{R}_x = \Big[r_{ij}^{(x)} \Big]$, $\mathbf{R}_y = \Big[r_{ij}^{(y)} \Big]$. $\mathbf{D}_{r,x}$ and $\mathbf{D}_{c,x}$ are the diagonal row and column sum matrices of $\mathbf{R}_x$, respectively, and similarly, $\mathbf{D}_{r,y}$ and $\mathbf{D}_{c,y}$ are the diagonal row and column sum matrices of $\mathbf{R}_y$. After removing the constant terms from Eqs.(17,18) we have the equivalent objective functions

$$\mathbf{F}(\boldsymbol{z}_y) = \boldsymbol{z}_y^\top \mathbf{D}_{c,x} \boldsymbol{z}_y - 2 \boldsymbol{z}_x^\top \mathbf{R}_x \boldsymbol{z}_y, \tag{19}$$

$$\mathbf{G}(\boldsymbol{z}_x) = \boldsymbol{z}_x^\top \mathbf{D}_{r,y} \boldsymbol{z}_x - 2 \boldsymbol{z}_x^\top \mathbf{R}_y \boldsymbol{z}_y. \tag{20}$$

The above can be simplified by setting $\mathbf{D}_r$ and $\mathbf{D}_c$ to be the diagonal row and column sum matrix of $\mathbf{R}$, so that

$$\mathbf{R}_x = \left[r_{ij} \left(\sum_{j=1}^{n} r_{ij} \right)^{\eta_1 - 1} \right] = \mathbf{D}_r^{\eta_1 - 1} \mathbf{R} = \mathbf{D}_r^{\eta_r} \mathbf{R}, \tag{21}$$

$$\mathbf{R}_y = \left[r_{ij} \left(\sum_{i=1}^{m} r_{ij} \right)^{\eta_2 - 1} \right] = \mathbf{R} \mathbf{D}_c^{\eta_2 - 1} = \mathbf{R} \mathbf{D}_c^{\eta_c}, \tag{22}$$

where $\eta_r = \eta_1 - 1$ and $\eta_c = \eta_2 - 1$.

Given a vector $\boldsymbol{z}_x$, the minimization of $\mathbf{F}(\boldsymbol{z}_y)$ produces an embedding $\boldsymbol{z}_y^*$ which best complies with information in $\mathbf{R}$, and similarly, given $\boldsymbol{z}_y$, the minimization of $\mathbf{G}(\boldsymbol{z}_x)$ produces an optimally compliant embedding $\boldsymbol{z}_x^*$. If there exists a pair $(\boldsymbol{z}_x^*, \boldsymbol{z}_y^*)$ that mutually satisfies both optimizations, then it can constitute an acceptable joint co-embedding for the row and column objects.

To avoid the collapse of the solutions $\boldsymbol{z}_x$ and $\boldsymbol{z}_y$, we need to impose the two following scale constraints

$$\boldsymbol{z}_x^\top \mathbf{D}_{r,y} \boldsymbol{z}_x = 1, \tag{23}$$

$$\boldsymbol{z}_y^\top \mathbf{D}_{c,x} \boldsymbol{z}_y = \zeta. \tag{24}$$

The parameter $\zeta \geq 0$ controls the relative scale between the embeddings $\boldsymbol{z}_x$ and $\boldsymbol{z}_y$, as their relative magnitudes need to be taken into account in the geometry of the recovered co-embeddings.

3.2. Co-Embedding Generation

Considering the optimization problem related to variable $\boldsymbol{z}_y$ only, the Lagrangian function for $\mathbf{F}(\boldsymbol{z}_y)$ is defined as

$$\begin{aligned} \mathcal{L}(\boldsymbol{z}_y, \mu_1) &= \boldsymbol{z}_y^\top \mathbf{D}_{c,x} \boldsymbol{z}_y - 2 \boldsymbol{z}_x^\top \mathbf{R}_x \boldsymbol{z}_y - \mu_1 (\boldsymbol{z}_y^\top \mathbf{D}_{c,x} \boldsymbol{z}_y - \zeta) \\ &= (1 - \mu_1) \boldsymbol{z}_y^\top \mathbf{D}_{c,x} \boldsymbol{z}_y - 2 \boldsymbol{z}_x^\top \mathbf{R}_x \boldsymbol{z}_y + \mu_1 \zeta, \end{aligned} \tag{25}$$

where μ_1 is the multiplier for the associated constraint. Differentiating with respect to the embedding $\boldsymbol{z}_y$, gives the following condition for stationarity

$$\frac{\partial \mathcal{L}(\boldsymbol{z}_y, \mu_1)}{\partial \boldsymbol{z}_y} = 2(1 - \mu_1) \mathbf{D}_{c,x} \boldsymbol{z}_y - 2 \mathbf{R}_x^\top \boldsymbol{z}_x = 0. \tag{26}$$

Combining Eqs.(24,26), yields

$$\boldsymbol{z}_y = \pm \alpha(\boldsymbol{z}_x) \mathbf{D}_{c,x}^{-1} \mathbf{R}_x^\top \boldsymbol{z}_x, \tag{27}$$

where we use the shorthand $\alpha(\boldsymbol{z}_x) = \sqrt{\frac{\zeta}{\boldsymbol{z}_x^\top \mathbf{R}_x \mathbf{D}_{c,x}^{-1} \mathbf{R}_x^\top \boldsymbol{z}_x}}$, defined as a function of the given embedding $\boldsymbol{z}_x$ of the row objects. The above expression for $\boldsymbol{z}_y$ provides

the set of possible solutions. Substituting this into Eq.(19), leads to a simpler expression given by

$$\begin{aligned}\mathbf{F}(\boldsymbol{z}_y) &= \zeta \mp 2\alpha(\boldsymbol{z}_x)\boldsymbol{z}_x^\top \mathbf{R}_x \mathbf{D}_{c,x}^{-1}\mathbf{R}_x^\top \boldsymbol{z}_x \\ &= \zeta \mp 2\alpha(\boldsymbol{z}_x)\frac{\zeta}{\alpha(\boldsymbol{z}_x)^2} \;=\; \zeta \mp \frac{2\zeta}{\alpha(\boldsymbol{z}_x)}. \end{aligned} \tag{28}$$

It can therefore be seen that, since $\zeta - \frac{2\zeta}{\alpha(\boldsymbol{z}_x)} < \zeta + \frac{2\zeta}{\alpha(\boldsymbol{z}_x)}$, the minimizing embedding is obtained by the positive branch of Eq.(27) as

$$\boldsymbol{z}_y^* = \underset{\substack{\boldsymbol{z}_y \in \mathcal{R}^n, \\ \boldsymbol{z}_y^\top \mathbf{D}_{c,x} \boldsymbol{z}_y = \zeta}}{\operatorname{argmin}} \; \mathbf{F}(\boldsymbol{z}_y) = \alpha(\boldsymbol{z}_x)\mathbf{D}_{c,x}^{-1}\mathbf{R}_x^\top \boldsymbol{z}_x. \tag{29}$$

We now consider the minimization of $\mathbf{G}(\boldsymbol{z}_x)$, given the embedding $\boldsymbol{z}_y$ for the column objects, under the constraint Eq.(23). The associated Lagrangian is

$$\begin{aligned}\mathcal{L}(\boldsymbol{z}_x, \mu_2) &= \boldsymbol{z}_x^\top \mathbf{D}_{r,y}\boldsymbol{z}_x - 2\boldsymbol{z}_x^\top \mathbf{R}_y \boldsymbol{z}_y - \mu_2(\boldsymbol{z}_x^\top \mathbf{D}_{r,y}\boldsymbol{z}_x - 1) \\ &= (1-\mu_2)\boldsymbol{z}_x^\top \mathbf{D}_{r,y}\boldsymbol{z}_x - 2\boldsymbol{z}_x^\top \mathbf{R}_y \boldsymbol{z}_y + \mu_2, \end{aligned} \tag{30}$$

where μ_2 is the multiplier. Similarly to the previous development, we can find that the minimizing embedding is given as

$$\boldsymbol{z}_x^* = \underset{\substack{\boldsymbol{z}_x \in \mathcal{R}^m, \\ \boldsymbol{z}_x^\top \mathbf{D}_{r,y} \boldsymbol{z}_x = 1}}{\operatorname{argmin}} \; \mathbf{G}(\boldsymbol{z}_x) = \beta(\boldsymbol{z}_y)\mathbf{D}_{r,y}^{-1}\mathbf{R}_y \boldsymbol{z}_y, \tag{31}$$

where $\beta(\boldsymbol{z}_y) = \frac{1}{\sqrt{\boldsymbol{z}_y^\top \mathbf{R}_y^\top \mathbf{D}_{r,y}^{-1}\mathbf{R}_y \boldsymbol{z}_y}}$ is defined to be a function of the given embedding $\boldsymbol{z}_y$ of the column objects.

As mentioned in Section 3.1, a desired co-embedding $(\boldsymbol{z}_x^*, \boldsymbol{z}_y^*)$ should mutually satisfy both optimization problems. Consequently, using Eqs.(29,31), we can explicitly make use of this interdependency to express $\boldsymbol{z}_x^*$ via

$$\boldsymbol{z}_x^* = \beta(\boldsymbol{z}_y^*)\mathbf{D}_{r,y}^{-1}\mathbf{R}_y \; \alpha(\boldsymbol{z}_x^*)\mathbf{D}_{c,x}^{-1}\mathbf{R}_x^\top \boldsymbol{z}_x^* = \alpha(\boldsymbol{z}_x^*)\beta(\boldsymbol{z}_y^*)\mathbf{T}\boldsymbol{z}_x^*, \tag{32}$$

where $\mathbf{T} = \mathbf{D}_{r,y}^{-1}\mathbf{R}_y\mathbf{D}_{c,x}^{-1}\mathbf{R}_x^\top$ is an $m \times m$ matrix defined here to simplify the notation. From Eq.(32), we can see that $\boldsymbol{z}_x^*$ should be an eigenvector of $\mathbf{T}$ with $\frac{1}{\alpha(\boldsymbol{z}_x^*)\beta(\boldsymbol{z}_y^*)}$ being the corresponding eigenvalue. Assuming the eigen-decomposition $\mathbf{T}\boldsymbol{\Psi} = \boldsymbol{\Psi}\boldsymbol{\Lambda}$, with $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_m]$ being the eigenvector matrix and $\boldsymbol{\Lambda} = \operatorname{diag}([\lambda_1, \lambda_2, \ldots, \lambda_m])$ the diagonal matrix of eigenvalues, we can take the sought embedding to be

$$\boldsymbol{z}_x^* = \frac{1}{\sqrt{\boldsymbol{\psi}_q^\top \mathbf{D}_{r,y}\boldsymbol{\psi}_q}}\boldsymbol{\psi}_q, \tag{33}$$

where the solving eigenvector $\boldsymbol{\psi}_q$ (the choice of q is addressed in Section 3.2.1) is scaled accordingly to satisfy the constraint in Eq.(23). Subsequently, the paired

embedding for the column objects can be calculated directly from Eq.(29) as $\boldsymbol{z}_y^* = \alpha(\boldsymbol{z}_x^*)\mathbf{D}_{c,x}^{-1}\mathbf{R}_x^\top \boldsymbol{z}_x^*$.

It has to be noted that the above assumes that $\frac{1}{\alpha(\boldsymbol{z}_x^*)\beta(\boldsymbol{z}_y^*)}$ corresponds to an eigenvalue λ_q of $\mathbf{T}$. This can be verified through the following steps

$$\begin{aligned}\alpha(\boldsymbol{z}_x^*)^2\beta(\boldsymbol{z}_y^*)^2 &= \frac{1}{\left(\frac{1}{\alpha(\boldsymbol{z}_x^*)}\boldsymbol{z}_y^*\right)^\top \mathbf{R}_y^\top \mathbf{D}_{r,y}^{-1}\mathbf{R}_y \left(\frac{1}{\alpha(\boldsymbol{z}_x^*)}\boldsymbol{z}_y^*\right)} \\ &= \frac{1}{\left(\boldsymbol{z}_x^{*\top}\mathbf{R}_x\mathbf{D}_{c,x}^{-1}\right)\mathbf{R}_y^\top\mathbf{D}_{r,y}^{-1}\mathbf{R}_y\left(\mathbf{D}_{c,x}^{-1}\mathbf{R}_x^\top\boldsymbol{z}_x^*\right)} \\ &= \frac{1}{\boldsymbol{z}_x^{*\top}\mathbf{T}^\top\mathbf{D}_{r,y}\mathbf{T}\boldsymbol{z}_x^*} = \frac{1}{\lambda_q^2\boldsymbol{z}_x^{*\top}\mathbf{D}_{r,y}\boldsymbol{z}_x^*} = \frac{1}{\lambda_q^2}, \end{aligned} \tag{34}$$

which rely on Eqs.(29,33).

A final issue concerning the feasibility of the above, is that $\mathbf{T} = \mathbf{D}_{r,y}^{-1}\mathbf{R}_y\mathbf{D}_{c,x}^{-1}\mathbf{R}_x^\top$ must have real and nonnegative eigenvalues λ_q. This can be shown to be the case, because from Eqs.(21,22), we have $\mathbf{R}_x = \mathbf{D}_r^{\eta_r}\mathbf{R}$ and $\mathbf{R}_y = \mathbf{R}\mathbf{D}_c^{\eta_c}$, and hence, the matrix $\mathbf{T}$ can be written as

$$\begin{aligned}\mathbf{T} &= \mathbf{D}_{r,y}^{-1}\mathbf{R}\mathbf{D}_c^{\eta_c}\mathbf{D}_{c,x}^{-1}\mathbf{R}^\top\mathbf{D}_r^{\eta_r} \\ &= \mathbf{D}_r^{-\frac{\eta_r}{2}}\mathbf{D}_{r,y}^{-1}\mathbf{D}_r^{\frac{\eta_r}{2}}\mathbf{R}\mathbf{D}_c^{\frac{\eta_c}{2}}\mathbf{D}_{c,x}^{-1}\mathbf{D}_c^{\frac{\eta_c}{2}}\mathbf{R}^\top\mathbf{D}_r^{\frac{\eta_r}{2}}\mathbf{D}_r^{\frac{\eta_r}{2}} \\ &= \mathbf{D}_r^{-\frac{\eta_r}{2}}\mathbf{D}_{r,y}^{-\frac{1}{2}}\left(\mathbf{D}_{r,y}^{-\frac{1}{2}}\mathbf{D}_r^{\frac{\eta_r}{2}}\mathbf{R}\mathbf{D}_c^{\frac{\eta_c}{2}}\mathbf{D}_{c,x}^{-\frac{1}{2}}\right)\left(\mathbf{D}_{c,x}^{-\frac{1}{2}}\mathbf{D}_c^{\frac{\eta_c}{2}}\mathbf{R}^\top\mathbf{D}_r^{\frac{\eta_r}{2}}\mathbf{D}_{r,y}^{-\frac{1}{2}}\right)\mathbf{D}_{r,y}^{\frac{1}{2}}\mathbf{D}_r^{\frac{\eta_r}{2}} \\ &= \mathbf{P}^{-1}\mathbf{A}^\top\mathbf{A}\mathbf{P}, \end{aligned} \tag{35}$$

where $\mathbf{P} = \mathbf{D}_{r,y}^{\frac{1}{2}}\mathbf{D}_r^{\frac{\eta_r}{2}}$ is a nonsingular diagonal matrix, and $\mathbf{A} = \mathbf{D}_{c,x}^{-\frac{1}{2}}\mathbf{D}_c^{\frac{\eta_c}{2}}\mathbf{R}^\top\mathbf{D}_r^{\frac{\eta_r}{2}}\mathbf{D}_{r,y}^{-\frac{1}{2}}$. Therefore, $\mathbf{T}$ is similar to the positive semidefinite matrix $\mathbf{A}^\top\mathbf{A}$, and consequently, it has the same eigenvalues.

3.2.1. Eigenvector selection

So far we have shown the form of the sought co-embedding $(\boldsymbol{z}_x^*, \boldsymbol{z}_y^*)$ from Eqs.(29,33). Because of the interdependency between the two sets of objects, we must minimize the two objective functions $\mathbf{F}(\boldsymbol{z}_y)$ and $\mathbf{G}(\boldsymbol{z}_x)$ simultaneously. From Eq.(28), we can see that the minimum value of $\mathbf{F}(\boldsymbol{z}_y)$ is $\zeta - \frac{2\zeta}{\alpha(\boldsymbol{z}_x^*)}$. Similarly, for $\mathbf{G}(\boldsymbol{z}_x)$, we can find that its minimum corresponds to $1 - \frac{2}{\beta(\boldsymbol{z}_y^*)}$. These two quantities obtain their smallest values when the denominators $\alpha(\boldsymbol{z}_x^*)$ and $\beta(\boldsymbol{z}_y^*)$ are as small as possible. Since they are both nonnegative, when $\alpha(\boldsymbol{z}_x^*)$ and $\beta(\boldsymbol{z}_y^*)$ achieve their minimum values, their product $\alpha(\boldsymbol{z}_x^*)\beta(\boldsymbol{z}_y^*)$ also is minimized. The latter is equivalent to choosing that eigenvector $\boldsymbol{\psi}_q$ of $\mathbf{T}$ that corresponds to the largest eigenvalue λ_q, in order to compute $\boldsymbol{z}_x^*$ using Eq.(33).

It can be seen that the largest eigenvalue of $\mathbf{T}$ is the unity with an associated eigenvector proportional to $\mathbf{1}_m$ (the m-length vector of ones). Firstly, since $\mathbf{D}_{c,x} = \mathrm{diag}(\mathbf{R}_x^\top\mathbf{1}_m)$ and $\mathbf{D}_{r,y} = \mathrm{diag}(\mathbf{R}_y\mathbf{1}_n)$, we have

$$\mathbf{T}\mathbf{1}_m = \mathbf{D}_{r,y}^{-1}\mathbf{R}_y\mathbf{D}_{c,x}^{-1}\mathbf{R}_x^\top\mathbf{1}_m = \mathbf{D}_{r,y}^{-1}\mathbf{R}_y\mathbf{1}_n = \mathbf{1}_m, \tag{36}$$

which shows that $(1, \mathbf{1}_m)$ is an eigenpair. Further, from the fact that the spectral radius $\rho(\cdot)$ of any square matrix cannot exceed any of the norms for that matrix, we have $\rho(\mathbf{T}) \leq \|\mathbf{T}\|_\infty = 1$. The last equality holds because $\mathbf{T}$ is a row stochastic matrix, that is, it has nonnegative elements, and from Eq.(36) its rows sum to one. Hence, we have $\rho(\mathbf{T}) = 1$ and no other eigenvalue greater than one exists.

However, we cannot select this largest eigenvalue, because its associated eigenvector $\mathbf{1}_m$ will produce via Eq.(33) an embedding $\boldsymbol{z}_x^*$ where all points collapse to a single coordinate. This leads the embedding $\boldsymbol{z}_y^*$ of the other group to also assume a single location. This degenerate solution relates to the case described in Section 3.1, where all embedded patterns coincide to yield the smallest possible aggregate costs, but here the scale constraints are also in force. Consequently, to avoid such solutions, we select the eigenvector $\boldsymbol{\psi}_q$ corresponding to the second largest eigenvalue λ_q. Note, that when $\mathbf{R}$ or a suitable permutation of it contain blocks of disconnected components, then $\mathbf{T}$ has a repeated semisimple eigenvalue of one. In this case, all the associated eigenvectors can be ignored as they map the different groups of the elements of $\boldsymbol{z}_x^*$ to constant coordinates. However, this situation may not correspond to a practically useful relational representation in $\mathbf{R}$ and the different blocks can be processed separately.

3.3. Multidimensional Extension

Although, so far we have focused on the estimation of a unidimensional co-embedding $(\boldsymbol{z}_x, \boldsymbol{z}_y)$, it is more practical for the purposes of visualization or supervised pattern analysis to generate k-dimensional (with $k > 1$) co-embeddings $(\mathbf{Z}_x, \mathbf{Z}_y) \in \mathcal{R}^{m\times k} \times \mathcal{R}^{n\times k}$. In analogy to the previous section, the additional axes can be recovered by processing the remaining eigenvectors of $\mathbf{T}$. Specifically, by assuming decreasing λ_q with increasing index q, and ignoring $\lambda_1 = 1$, we choose k eigenvectors $\boldsymbol{\psi}_{q+1}$ with $q = 1, \ldots, k$. It has to be noted, that since $\mathbf{T}$ is rank deficient and the quantity $\frac{1}{\alpha(\boldsymbol{z}_x^*)\beta(\boldsymbol{z}_y^*)}$ is not defined for zero eigenvalues, we have $k \leq \text{rank}(\mathbf{T}) - 1 \leq \min(m, n) - 1$. However, in practice a small number of the available dimensions is utilized.

In the computed co-embedding, the scale constraints of Eqs.(23,24) need to be maintained for all axes, as

$$\text{diag}(\mathbf{Z}_x^\top \mathbf{D}_{r,y} \mathbf{Z}_x) = \mathbf{1}_k, \tag{37}$$

$$\text{diag}(\mathbf{Z}_y^\top \mathbf{D}_{c,x} \mathbf{Z}_y) = \boldsymbol{\zeta}. \tag{38}$$

The vector $\boldsymbol{\zeta} = [\zeta_1, \zeta_2, \ldots, \zeta_k]^\top$ contains the parameters that control the relative scale between the embedded patterns from the row and column objects at each axis. In order for the k axes to represent different and non-redundant coordinate information, the eigenvectors $\boldsymbol{\psi}_{q+1}$ must be independent. It turns out, that this is the case here due to the problem formulation and without additional constraints in the optimization. Specifically, from Eq.(35), we have $\mathbf{T} = \mathbf{P}^{-1}\mathbf{A}^\top\mathbf{A}\mathbf{P}$, where $\mathbf{P} = \mathbf{D}_{r,y}^{\frac{1}{2}}\mathbf{D}_r^{\frac{\eta_r}{2}}$ is diagonal. Then, the decomposition $\mathbf{T}\boldsymbol{\Psi} = \boldsymbol{\Psi}\boldsymbol{\Lambda}$ can be written as $\mathbf{P}^{-1}\mathbf{A}^\top\mathbf{A}\mathbf{P}\boldsymbol{\Psi} = \boldsymbol{\Psi}\boldsymbol{\Lambda}$ or $\mathbf{A}^\top\mathbf{A}(\mathbf{P}\boldsymbol{\Psi}) = (\mathbf{P}\boldsymbol{\Psi})\boldsymbol{\Lambda}$. This shows that $\mathbf{P}\boldsymbol{\Psi}$

contains the eigenvectors of a symmetric matrix, and therefore, $\mathbf{P\Psi}$ is orthogonal. This is equivalent to $\mathbf{\Psi}^\top\mathbf{P}^2\mathbf{\Psi}$ being diagonal, that is, all the eigenvectors of $\mathbf{T}$ are orthogonal with respect to the scaling matrix $\mathbf{D}_{r,y}\mathbf{D}_r^{\eta_r}$.

Finally, the qth columns of $\mathbf{Z}_x$ and of $\mathbf{Z}_y$ are taken to be

$$\mathbf{Z}_x^{(q)} = \frac{1}{\sqrt{\boldsymbol{\psi}_{q+1}^\top \mathbf{D}_{r,y}\boldsymbol{\psi}_{q+1}}}\boldsymbol{\psi}_{q+1}, \tag{39}$$

$$\mathbf{Z}_y^{(q)} = \alpha(\mathbf{Z}_x^{(q)}, \zeta_q)\, \mathbf{D}_{c,x}^{-1}\mathbf{R}_x^\top \mathbf{Z}_x^{(q)}, \tag{40}$$

where the quantity $\alpha(\cdot)$ is as defined for Eq.(27), but it now depends also on the qth scale parameter and is equal to $\sqrt{\frac{\zeta_q}{\mathbf{Z}_x^{(q)\top}\mathbf{R}_x\mathbf{D}_{c,x}^{-1}\mathbf{R}_x^\top\mathbf{Z}_x^{(q)}}}$.

3.3.1. Parameter reduction heuristics

The optimal selection for the proposed model, depends on the dimensionality k, the k embedding scaling parameters ζ_q, and the two data weighting parameters η_r and η_c (the latter two parameters were introduced in Section 3.1 to scale the objectives, and as they indirectly parameterize matrix $\mathbf{T}$ they cannot vary with each qth dimension). The mechanism and the objective function that drive the model selection will be described in Section 3.4. Here, we show that in the absence of additional information for the relative scales between the row and column embeddings, we can make use of certain heuristics that reduce the number of parameters which need to be identified.

Specifically, we experimentally observed that the identification of multiple ζ_q can be sensitive to the search resolution, and it is more robust to look instead for surrogate parameters $\xi_q > 0$ that can be defined as $\alpha(\boldsymbol{z}_x^*)\sqrt{\lambda_{q+1}}$ or equivalently $\frac{\sqrt{\alpha(\boldsymbol{z}_x^*)}}{\sqrt{\beta(\boldsymbol{z}_y^*)}}$. From the last ratio, it can be intuitively understood that when no scale information about $\mathbf{R}$ is available, ξ_q can be searched within intervals around the value of one. This is because we can assume that the two objective functions $\mathbf{F}(\boldsymbol{z}_y)$ and $\mathbf{G}(\boldsymbol{z}_x)$ are of equal importance and that the two sets of embedded objects are in relative scale by having $\zeta_q \approx 1$, which would make the quantities $\alpha(\boldsymbol{z}_x^*)$ and $\beta(\boldsymbol{z}_y^*)$ to be close to each other. Using this new quantity, and substituting for $\alpha(\boldsymbol{z}_x^*)$, the equation for the column embedded points can be written as $\boldsymbol{z}_y^* = \frac{\xi_q}{\sqrt{\lambda_{q+1}}}\mathbf{D}_{c,x}^{-1}\mathbf{R}_x^\top\boldsymbol{z}_x^*$. Searching for the surrogate parameters ξ_q is easier and more efficient for the model identification.

The above can facilitate a parameter reduction based on the following. We firstly simplify the model by replacing all k parameters ξ_q with a single scalar parameter ξ. Although this speeds up model search significantly, it also removes the flexibility for scale variation between the different axes. To compensate for this, we introduce a second scaling parameter $\gamma \geq 0$ that indirectly adjusts the scale between the different axes through the eigenvalue ratio for a particular iteration. Specifically, the row embedded points $\boldsymbol{z}_x^*$ are adjusted by a factor of $\left(\frac{\lambda_{q+1}}{\lambda_2}\right)^\gamma$. This is useful because eigenvalues correspond to model costs at each step, and with the introduced weighted ratio we can achieve some degree of

scale variation across the coordinates. We have experimentally validated the usefulness and practicality of the proposed parameter reduction heuristics. In summary, the final co-embedding is calculated according to

$$\mathbf{Z}_x^{(q)} = \left(\frac{\lambda_{q+1}}{\lambda_2}\right)^{\gamma} \frac{1}{\sqrt{\boldsymbol{\psi}_{q+1}^{\top}\mathbf{D}_{r,y}\boldsymbol{\psi}_{q+1}}}\boldsymbol{\psi}_{q+1}, \tag{41}$$

$$\mathbf{Z}_y^{(q)} = \frac{\xi}{\sqrt{\lambda_{q+1}}}\mathbf{D}_{c,x}^{-1}\mathbf{R}_x^{\top}\mathbf{Z}_x^{(q)}. \tag{42}$$

3.4. Model Identification

Computing a suitable co-embedding involves the identification of the four model parameters η_1, η_2, ξ and γ. A simple search procedure, such as a grid search, simulated annealing or a genetic algorithm can be used. However, given an input relational matrix $\mathbf{R}$, we need to define a suitable objective function that drives this search. In line with previous work [24], such an objective should compare the original input $\mathbf{R}$ against a re-estimated source based on the generated co-embedding $(\mathbf{Z}_x, \mathbf{Z}_y)$. This can be done, for example, by computing a between-group Euclidean distance matrix $\mathbf{Q} \in \mathcal{R}^{m\times n}$ between the rows of $\mathbf{Z}_x$ and $\mathbf{Z}_y$. Then, a possibility would be to simply minimize the normalized sum of the element-wise products $\frac{\mathrm{tr}(\mathbf{R}\mathbf{Q}^{\top})}{\sqrt{\mathrm{tr}(\mathbf{R}\mathbf{R}^{\top})\,\mathrm{tr}(\mathbf{Q}\mathbf{Q}^{\top})}}$ between $\mathbf{R}$ and $\mathbf{Q}$. However, this procedure was found to be unreliable due to disproportionate error contributions from the different entries. Alternative but more complex schemes based on quantization have previously been used in the evaluation of ACAS [24].

In this work, we evaluate the conformity between $\mathbf{R}$ and $\mathbf{Q}$ using the local structural information of the between-group similarities and dissimilarities they represent. Specifically, we capture the local neighborhoods between the member patterns of the generated co-embedding, by defining a binary matrix $K(\mathbf{R})$. Each of its ijth element is one, if and only if the ith row object is within the first k_r neighbors of the jth column object (that is among the k_r largest entries in the jth column), and at the same time the jth column object is within the first k_c neighbors of the ith row object. The neighborhood is established by using the raw similarities within $\mathbf{R}$, and the parameters k_r and k_c which can be preassigned or set to be a small percentage of the cardinalities m and n, respectively. The quantity $K(\mathbf{Q})$ is similarly defined using the distance information within matrix $\mathbf{Q}$ (the neighbors here are based on the smallest entries in the columns or rows).

This type of mutual neighborhood information can reveal useful structural characteristics that enable the comparison between $\mathbf{R}$ and $\mathbf{Q}$. For example, it can detect co-clustering arrangements between heterogeneous object types, without being sensitive to large error contributions from element-wise comparisons and the discrepant type of information represented by the original $\mathbf{R}$ and the re-estimated $\mathbf{Q}$. Finally, the model parameters η_1, η_2, ξ and γ are identified by minimizing the quantity

$$\Gamma(\mathbf{R}, \mathbf{Q}) = \mathbf{1}_m^{\top}\left(K(\mathbf{R}) - K(\mathbf{Q}) \odot K(\mathbf{R})\right)\mathbf{1}_n, \tag{43}$$

where $\odot$ denotes the Hadamard matrix multiplication, and $\mathbf{Q}$ depends on the co-embedding $(\mathbf{Z}_x, \mathbf{Z}_y)$, which in turn depends on the four search parameters. From the definition of $\Gamma(\cdot)$, it can be seen that it is a sum of the unity errors, defined only at the ijth matrix elements which correspond to object pairs (i, j) that are in mutual local neighborhoods in $\mathbf{R}$ but not $\mathbf{Q}$. The measure assumes that loss of local neighborhood structure from the original similarity matrix breaks down the initial requirement for the co-embedding to preserve local proximity information.

1. **Input**: An $m \times n$ input matrix $\mathbf{R}$ representing similarities between the m row objects and the n column objects, the co-embedding dimension k, and the local neighborhood control parameters k_r, k_c.
2. **Initialization**:
   (a) Set restrictions on the search range of the model parameters, as: $\eta_1, \eta_2 \in [0, 10]$, $\xi \in (0, 3]$, and $\gamma \in [0, 3]$.
   (b) Set some starting values for these model parameters (depending on the search method employed).
3. **Main loop** (repeated as long as step (j) cannot reduce the model cost $\Gamma(\mathbf{R}, \mathbf{Q})$ any further):
   (a) Set $\mathbf{D}_r = \text{diag}(\mathbf{R}\mathbf{1}_n)$ and $\mathbf{D}_c = \text{diag}(\mathbf{R}^\top \mathbf{1}_m)$.
   (b) Calculate $\mathbf{R}_x = \mathbf{D}_r^{\eta_1 - 1}\mathbf{R}$ and $\mathbf{R}_y = \mathbf{R}\mathbf{D}_c^{\eta_2 - 1}$, as in Eqs.(21,22).
   (c) Set $\mathbf{D}_{r,y} = \text{diag}(\mathbf{R}_y \mathbf{1}_n)$ and $\mathbf{D}_{c,x} = \text{diag}(\mathbf{R}_x^\top \mathbf{1}_m)$.
   (d) Construct the matrix $\mathbf{T} = \mathbf{D}_{r,y}^{-1}\mathbf{R}_y\mathbf{D}_{c,x}^{-1}\mathbf{R}_x^\top$.
   (e) Perform an eigen-decomposition of $\mathbf{T}$, where $\mathbf{\Psi} = [\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_m]$ is the eigenvector matrix and $\mathbf{\Lambda} = \text{diag}([\lambda_1, \ldots, \lambda_m])$ contains the eigenvalues in descending order.
   (f) Calculate each qth column (where $q = 1, \ldots, k$) of $\mathbf{Z}_x$ and of $\mathbf{Z}_y$ using Eqs.(41,42).
   (g) Use the resulting co-embedding $(\mathbf{Z}_x, \mathbf{Z}_y) \in \mathcal{R}^{m \times k} \times \mathcal{R}^{n \times k}$, to estimate a between-group Euclidean distance matrix $\mathbf{Q} \in \mathcal{R}^{m \times n}$.
   (h) Calculate the neighborhood indicator structures $K(\mathbf{R})$ and $K(\mathbf{Q})$, as defined in Section 3.4.
   (i) Compute the model cost $\Gamma(\mathbf{R}, \mathbf{Q})$ using Eq.(43).
   (j) If the cost is less than the minimum found so far, store the values of the current model parameters η_1, η_2, ξ and γ and update them to the next search values (depending on the employed search procedure and resolution of the search).
4. **Output**: The optimal co-embedding $(\mathbf{Z}_x, \mathbf{Z}_y)$ and their associated optimal model parameters.

Table 1: Description of the proposed algorithm using a generic type of search for the identification of the optimal co-embedding and the model parameters.

The overall set of operations for the proposed method is summarized in Table 1. It can be seen that the most complex step for the model identification is the eigen-decomposition of matrix $\mathbf{T}$. This is typically of $O(\min(m, n)^3)$, by swapping conveniently the roles of groups $\mathcal{X}$ and $\mathcal{Y}$. The construction of

$\mathbf{T}$ ignoring scaling operations is of $O(\min(m,n)^2\max(m,n))$. The number of decompositions depends on the number of iterations of the adopted search procedure. However, as $\mathbf{T}$ depends only on η_1 and η_2, and not on ξ and γ, a new decomposition is needed only when the former two parameters are updated during the search. For each possible co-embedding, the calculation of $\mathbf{Q}$ is of $O(mnk)$, finding $K(\mathbf{Q})$ of $O(mk_r + nk_c)$, and $\Gamma(\mathbf{R}, \mathbf{Q})$ of $O(mn)$.

4. Experimental Analysis and Results

In this section, we compare the co-embedding generation capabilities of the proposed algorithm with existing state of the art methods, including CA, ACAS, CODE and the recently developed multiple kernel preserving embedding (MKPE) algorithm to preserve similarity between heterogeneous groups of data [32]. We examine both qualitative and quantitative aspects of the comparison, which correspond to effectiveness in data visualization and supervised machine learning. We split the experimentation into three main parts, solving three different data visualization and analysis tasks using a total of 15 datasets:

- The reconstruction of the 2D distribution of data objects given the partial similarities between them (Section 4.1). Eight 2D synthetic datasets[1] with multiple geometric arrangements and clusters with patterns separated into two groups are used.
- The simultaneous learning of the distributional representations of documents and words in the same space, based on the frequency information the words appearing in the documents (Section 4.2). Four document collections containing clinical trials [15], Reuters new articles [15], 20 newsgroup documents[2] and online reviews [33] are used.
- The learning of low-dimensional representations of objects based on link information contained in knowledge graphs (Section 4.3). We use three datasets [34] with the citation networks between the Cora and Citeseer documents, as well as the co-occurrence network between industrial companies.

To identify the optimal model, a set of values for its four model parameters is searched for within the ranges $\eta_1, \eta_2 \in [0, 10]$, $\xi \in (0, 3]$ and $\gamma \in [0, 3]$. The local neighborhood parameters k_r and k_c are both fixed to 5 (in general small values such as $\{5, 10, 15\}$ are the most appropriate; Section 4.4 analyzes their effect on performance). To implement the actual search procedure, we employ a simple genetic algorithm[3], which relies on the $\Gamma(\cdot,\cdot)$ index of Eq.(43) to be its

[1] Some datasets are generated by us and some are downloaded from `cran.r-project.org/web/packages/mlbench`, `cs.joensuu.fi/sipu/datasets`, and `search.r-project.org/library/fpc/html/rFace`.

[2] Downloaded from `qwone.com/~jason/20Newsgroups`.

[3] Using Matlab ver.8.6 internal toolbox implementation.

minimizing objective function, supported by fitness ranking, stochastic uniform parent selection and an elitism operator. It uses a population of 52 real-valued encoded solutions, and terminates when fitness improvement stagnates for 50 generations. Furthermore, we use uniform crossover at a crossover rate of 0.8, and in order to maintain feasibility of the range constraints an adaptive feasible mutation.

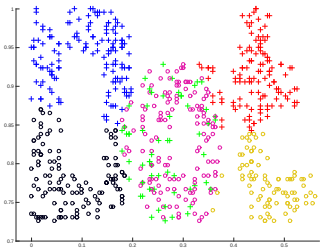
4.1. Reconstruction of Synthetic 2D Data Points

The eight synthetic datasets are shown in Figure 1, where data points are allocated to groups $\mathcal{X}$ or $\mathcal{Y}$ and form various clusters and co-clusters. We calculate the initial input relational measurements in $\mathbf{R}$, according to

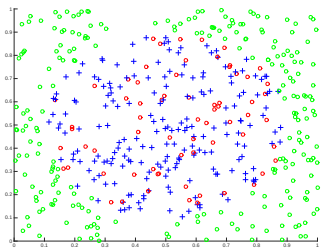
$$r_{ij} = \exp\left(\frac{-mn\|\boldsymbol{x}_i - \boldsymbol{y}_j\|_2^2}{\sum_{i=1}^{m}\sum_{j=1}^{n}\|\boldsymbol{x}_i - \boldsymbol{y}_j\|_2^2}\right), \tag{44}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ are the m and n coordinates of the patterns from groups $\mathcal{X}$ and $\mathcal{Y}$, respectively.

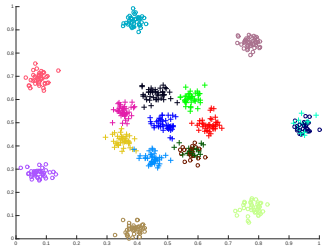
The co-embeddings from all methods and for all datasets are displayed in Figures 2-4. It can be observed that the representations generated by the proposed method resemble more the original arrangements of Figure 1. In most cases, most methods are shown to possess the ability to preserve the principal spatial characteristics, but the existing ones often do not fully capture the proximities of the cluster structures within each individual group $\mathcal{X}$ and $\mathcal{Y}$ or between them. For example, the Rface co-embedding of CODE in Figure 3(b) and that of CA in Figure 3(e) reliably represent the eyes and nose components of the original set in Figure 1(e), but they both completely break down the composition of the mouth structure, which is a co-cluster of points belonging to both $\mathcal{X}$ and $\mathcal{Y}$, and CODE also fails to preserve the relative location of the chin. ACAS in Figure 3(e) totally distorts the relative positions of the different components. For the Target dataset of Figure 1(d), CA, ACAS and CODE in Figure 3(a,d,g) seem to preserve the continuity of the three surrounding clusters separately comprising points from both groups. For the two middle structures forming a single co-cluster structure in Figure 1(d), however, CODE segregates the co-cluster, while ACAS preserves it but compresses one of the structures. The proposed method is shown in Figure 3(m) to reliably reproduce the co-cluster as a uniform mix of points from both groups. Similar observations can be made for the remaining datasets, e.g., for the ring and the compound datasets, CODE, ACAS and CA fail to preserve the co-cluster structures. The proposed algorithm qualitatively shows to preserve the structure, shapes and relative proximities of both within-group clusters and between-group co-clusters. MKPE does not perform well for almost all the datasets, because ideally the algorithm requires both between-group and within-group relations to recover the heterogeneous data embedding. When there is only partial relation information available, e.g., the relation matrix $\mathbf{R}$ between groups $\mathcal{X}$ and $\mathcal{Y}$, it is difficult for MKPE to generate embedding that can satisfactory recover the input relation.



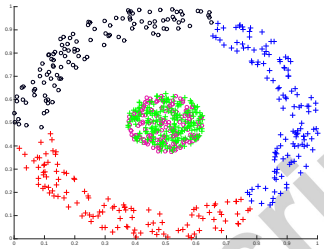
(a) Dots (m=336, n=364)



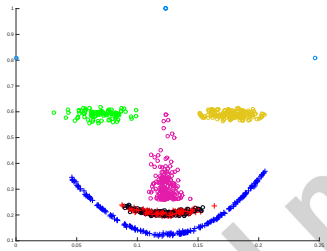
(b) Circles (m=200, n=300)



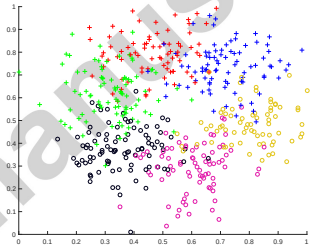
(c) R15 (m=313, n=287)



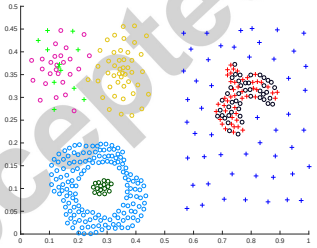
(d) Target (m=496, n=262)



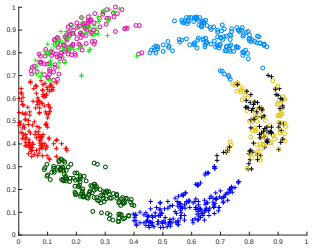
(e) Rface (m=330, n=670)



(f) 2dnormals (m=256, n=244)

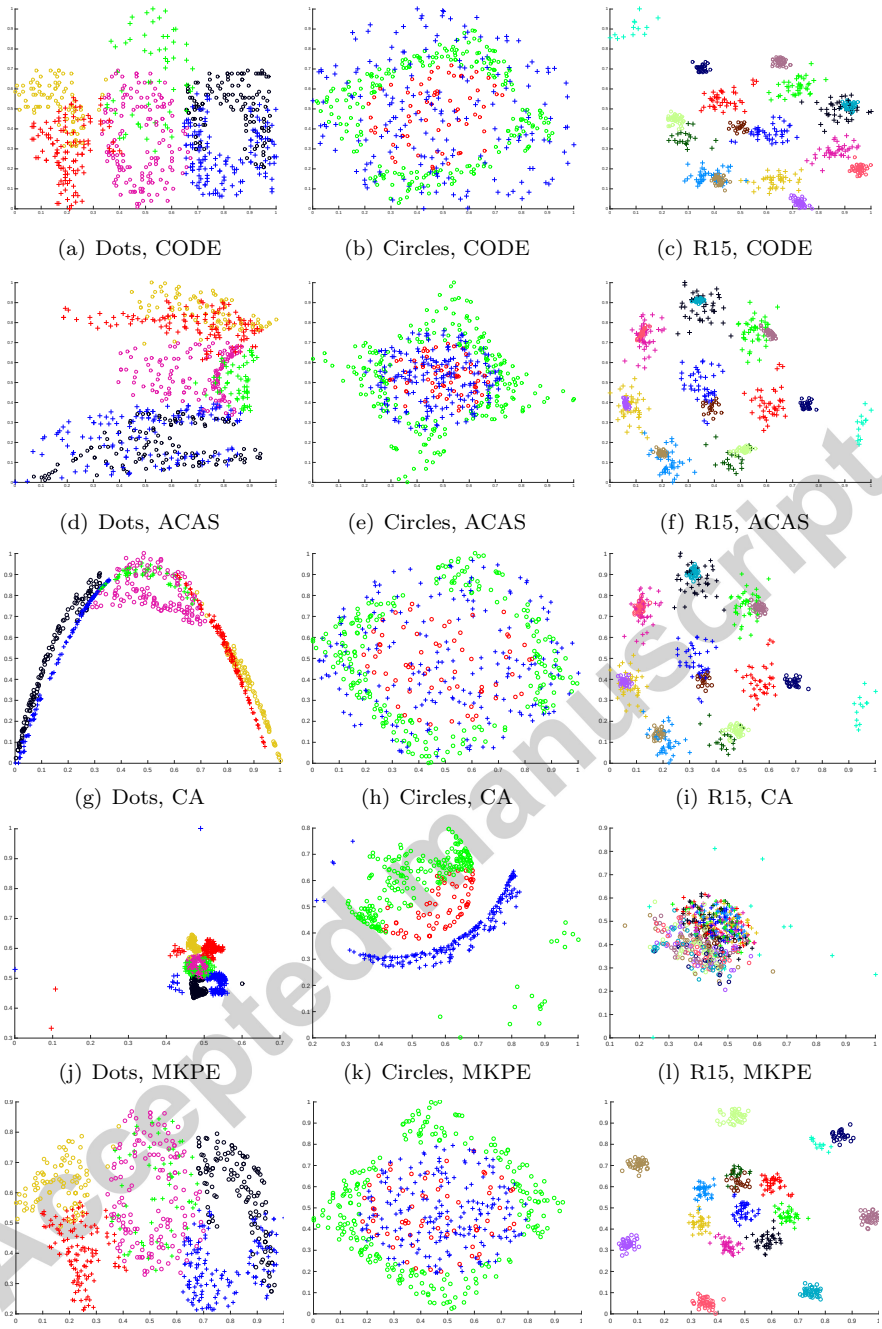


(g) Compound (m=108, n=291)



(h) Ring (m=504, n=496)

Figure 1: Original patterns of the synthetic 2D datasets. Different colors correspond to different clusters and spatial structures. All points with the same color are allocated either to group $\mathcal{X}$ (marked by "∘") or to group $\mathcal{Y}$ (marked by "+"). The cardinalities $m = |\mathcal{X}|$ and $n = |\mathcal{Y}|$ of the groups are shown for each dataset.



(a) Dots, CODE (b) Circles, CODE (c) R15, CODE

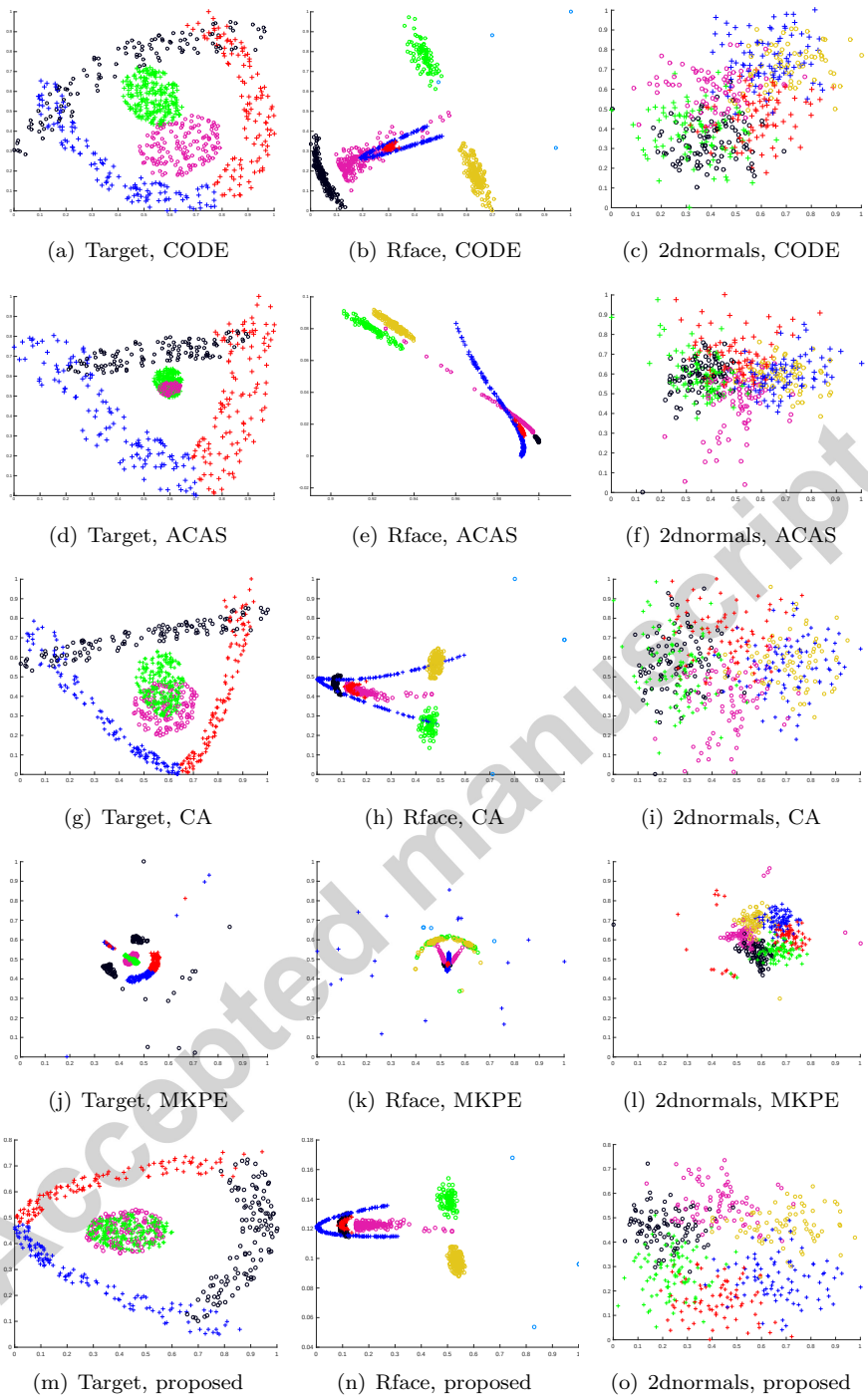
(d) Dots, ACAS (e) Circles, ACAS (f) R15, ACAS

(g) Dots, CA (h) Circles, CA (i) R15, CA

(j) Dots, MKPE (k) Circles, MKPE (l) R15, MKPE

(m) Dots, proposed (n) Circles, proposed (o) R15, proposed

Figure 2: Co-embeddings generated by different algorithms, for the synthetic datasets of Dots, Circles and R15 displayed in Figure 1. Co-embedding axes are scaled within $[0, 1]$.



(a) Target, CODE (b) Rface, CODE (c) 2dnormals, CODE

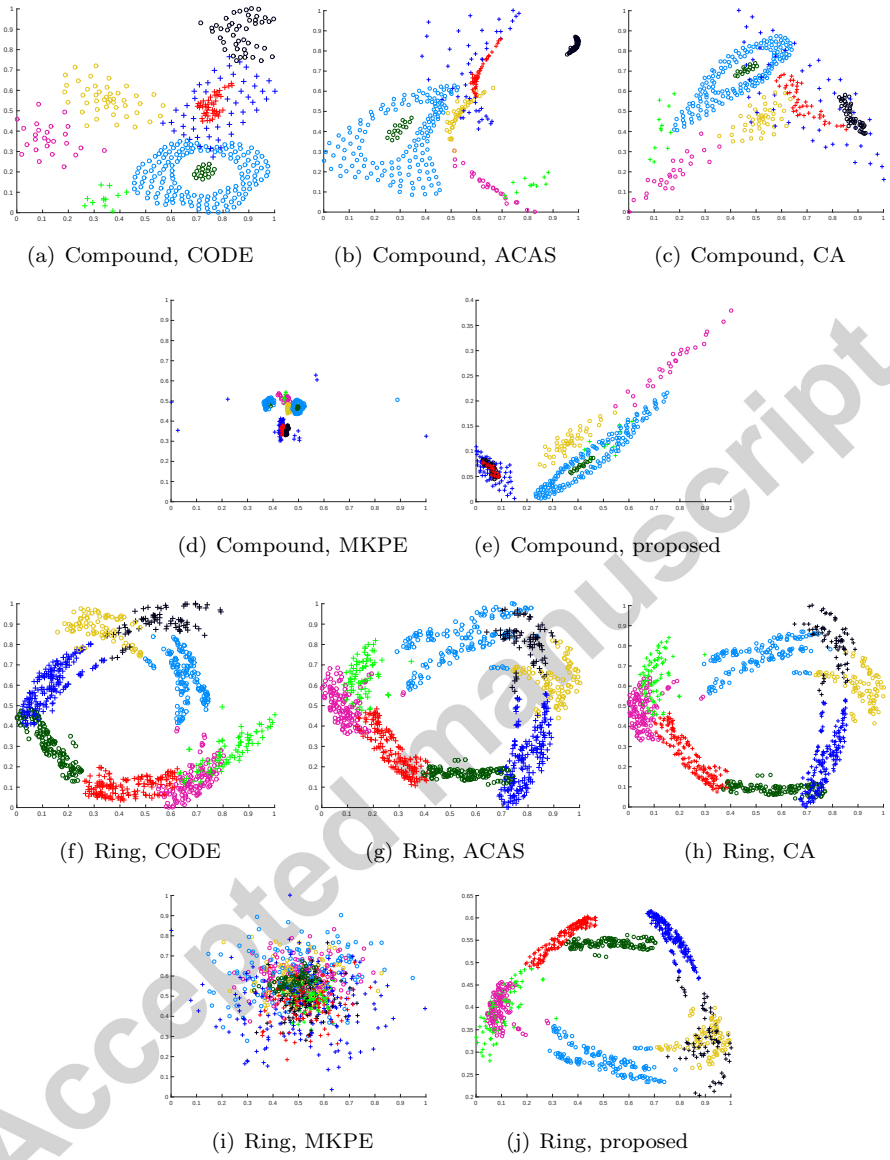
(d) Target, ACAS (e) Rface, ACAS (f) 2dnormals, ACAS

(g) Target, CA (h) Rface, CA (i) 2dnormals, CA

(j) Target, MKPE (k) Rface, MKPE (l) 2dnormals, MKPE

(m) Target, proposed (n) Rface, proposed (o) 2dnormals, proposed

Figure 3: Co-embeddings generated by different algorithms, for the synthetic datasets of Target, Rface and 2dnormals displayed in Figure 1. Co-embedding axes are scaled within $[0, 1]$.



(a) Compound, CODE (b) Compound, ACAS (c) Compound, CA

(d) Compound, MKPE (e) Compound, proposed

(f) Ring, CODE (g) Ring, ACAS (h) Ring, CA

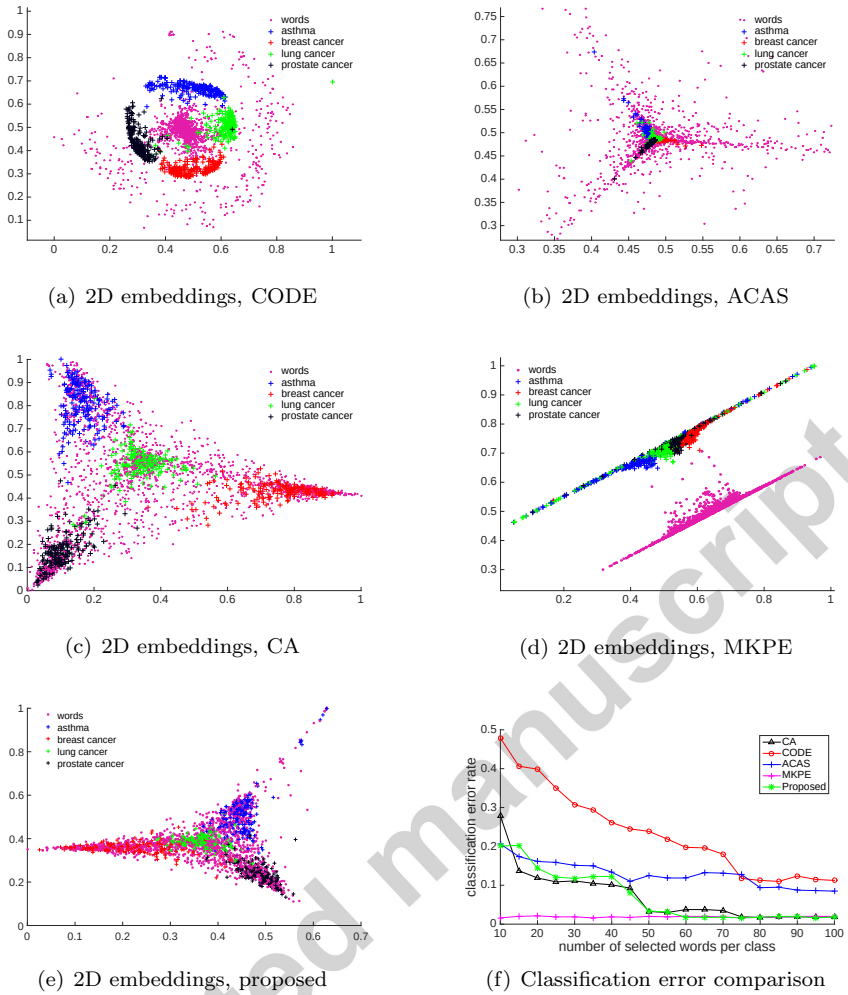
(i) Ring, MKPE (j) Ring, proposed

Figure 4: Co-embeddings generated by different algorithms, for the synthetic datasets compound and ring displayed in Figure 1. Co-embedding axes are scaled within $[0, 1]$.

4.2. Learning Distributional Representations of Documents and Words

Given a collection of documents and a dictionary of unique words, the absence/presence (or frequency) of the words occurring in each document readily provides a source of information on the similarities $\mathbf{R}$ between the documents group $\mathcal{X}$ and the word group $\mathcal{Y}$. Low-dimensional representations can be learned from $\mathbf{R}$ for both documents and words, reflecting the document and word distributions in the same space. The empirical co-occurrence counts from the clinical trials and Reuters news articles are used. In order to quantitatively assess the co-embedding quality, we make use of the class information available in the document collections. After embedding the documents and words in the same space, we first estimate the center for each document class. Then, we calculate the Euclidean distance between the words and each class center in the co-embedding space. Sets of words that are closest to the document class centers are selected. For this particular application, it is reasonable to expect that in a reliable co-embedding map, the words that are more important for the identification of a document class are proximate to the documents that belong to this class. Thus, the selected words are expected to possess higher discriminating power to distinguish between the document classes. Subsequently, we recompose a new document-by-word frequency matrix by only retaining the selected words. This recomposed frequency matrix can constitute the feature matrix input to a classification algorithm, using the document class labels as the target class memberships, and the resulting classification accuracy to represent the co-embedding quality. To implement this, we use a support vector machine (SVM) classifier[3] for all experiments in this section. The classifier uses, for simplicity, a linear kernel, without any data scaling or standardization applied to the predictors. It employs sequential minimal optimization (SMO) for finding the optimal hyperplane, and a regularization parameter for the nonseparable cases set to 1. The multiclass setup is based on an one-against-all coding, and a 10-fold cross-validation is used for model assessment.

Firstly, we visually demonstrate the learned 2D co-embeddings for m=800 clinical trials documents and n=1,780 unique words. Each clinical trial is assigned to one of the four disease classes of asthma, breast cancer, lung cancer and prostate cancer, and each class contains 200 documents. Infrequent words are removed, and only the most informative ones are retained. The word occurrence count is used as the relation measure between a document and a word in $\mathbf{R}$. The resulting co-embeddings of all algorithms are illustrated in Figure 5, where the document objects from the four classes are plotted together with the word objects. It can be seen that the ACAS and CODE algorithms have generated words that can be quite far away from the four classes of documents. For the proposed method and CA, this is far less pronounced, as the document objects appear to be blending with the word objects. It is relatively easy to identify the document-word proximities which correspond to inter-object similarities. As for MKPE, the document and word objects are displayed along two separate and roughly parallel linear arrangements. We also use Figure 5(f) to compare the classification rates that represent the reliability of the proximity between the embedded document and word objects for varying numbers of selected words



(a) 2D embeddings, CODE

(b) 2D embeddings, ACAS

(c) 2D embeddings, CA

(d) 2D embeddings, MKPE

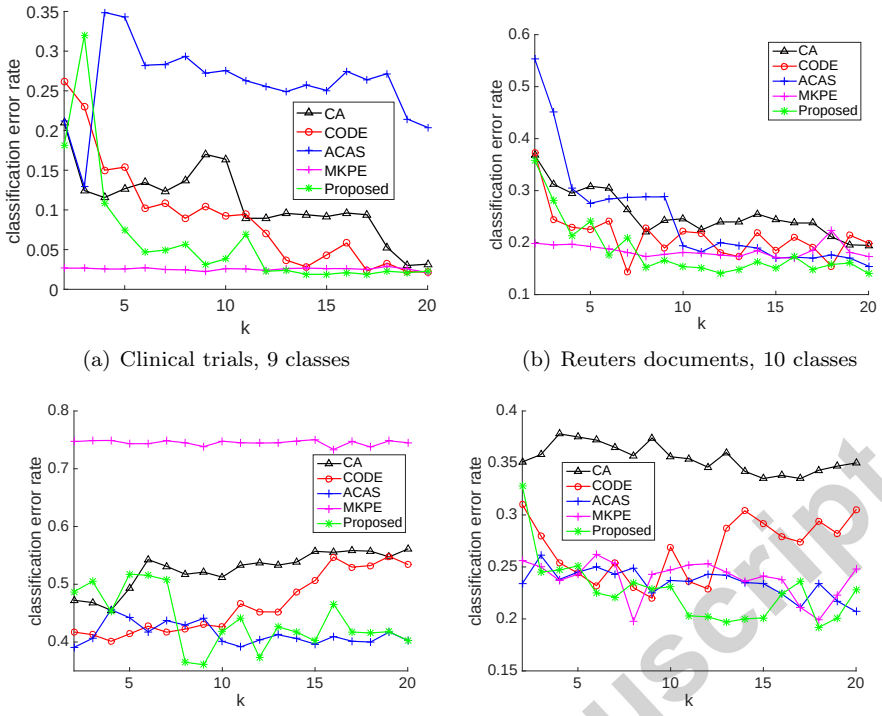
(e) 2D embeddings, proposed

(f) Classification error comparison

Figure 5: 2D demonstration and classification error comparison of co-embeddings generated for 800 clinical trials and 1,780 words belonging to four classes by different methods. Each document (marked by "+") is a member of group $\mathcal{X}$ and belongs to one of the four topics (plotted in different color). Each member of group $\mathcal{Y}$ (marked by "•") is a word object.

(from 10 to 100) per document class. It can be seen that the proposed method and CA show comparable error rates that are lower than CODE and ACAS. MKPE possesses lower error rates in this case, and this indicates that despite being separated along two parallel arrangements, the locations of each document class and its related words are actually compatible. However, although offering a low error rate, such separation is not optimal for visualization purposes.

We further evaluate the algorithms with four more datasets possessing more complex class structures. One is a larger collection of clinical trials consisting of



(a) Clinical trials, 9 classes

(b) Reuters documents, 10 classes

(c) 20Newsgroups documents, 20 classes

(d) Online reviews, 5 classes

Figure 6: Comparison of the classification error rates of different algorithms, for varying the number k of the selected words that are closest to each class center using different document collections.

m=1,800 documents (with 200 documents per topic), containing n=2,300 words after removing the infrequent words and belonging to the nine disease classes of asthma, breast cancer, lung cancer, prostate cancer, cardiovascular, HIV, leukemia, depression and schizophrenia. Another dataset is the Reuters news article collection, containing m=976 documents represented by n=2,185 words, belonging to ten document classes of earn, acq, crude, trade, money-fx, interest, ship, sugar, money-supply and coffee. A third one is the 20Newsgroups data, containing m=3,000 newsgroup documents represented by n=2,000 frequently occurring words, belonging to twenty newsgroups. The fourth dataset is the online review collection containing m=2,000 review documents represented by n=1,500 words after text processing, belonging to the 5 topics of movies, books, dvds, electronics and kitchen. For these datasets, the word occurrence counts in the documents are used as the relation measurements in $\mathbf{R}$. As all datasets contain multiple classes, it is insufficient to learn 2D embeddings to characterize the class structure. We, thus, fix the number of selected words per class to 20 and examine the classification performance while varying the number of used embeddings (k changes from 2 to 20). It can be seen from Figure 6 that the

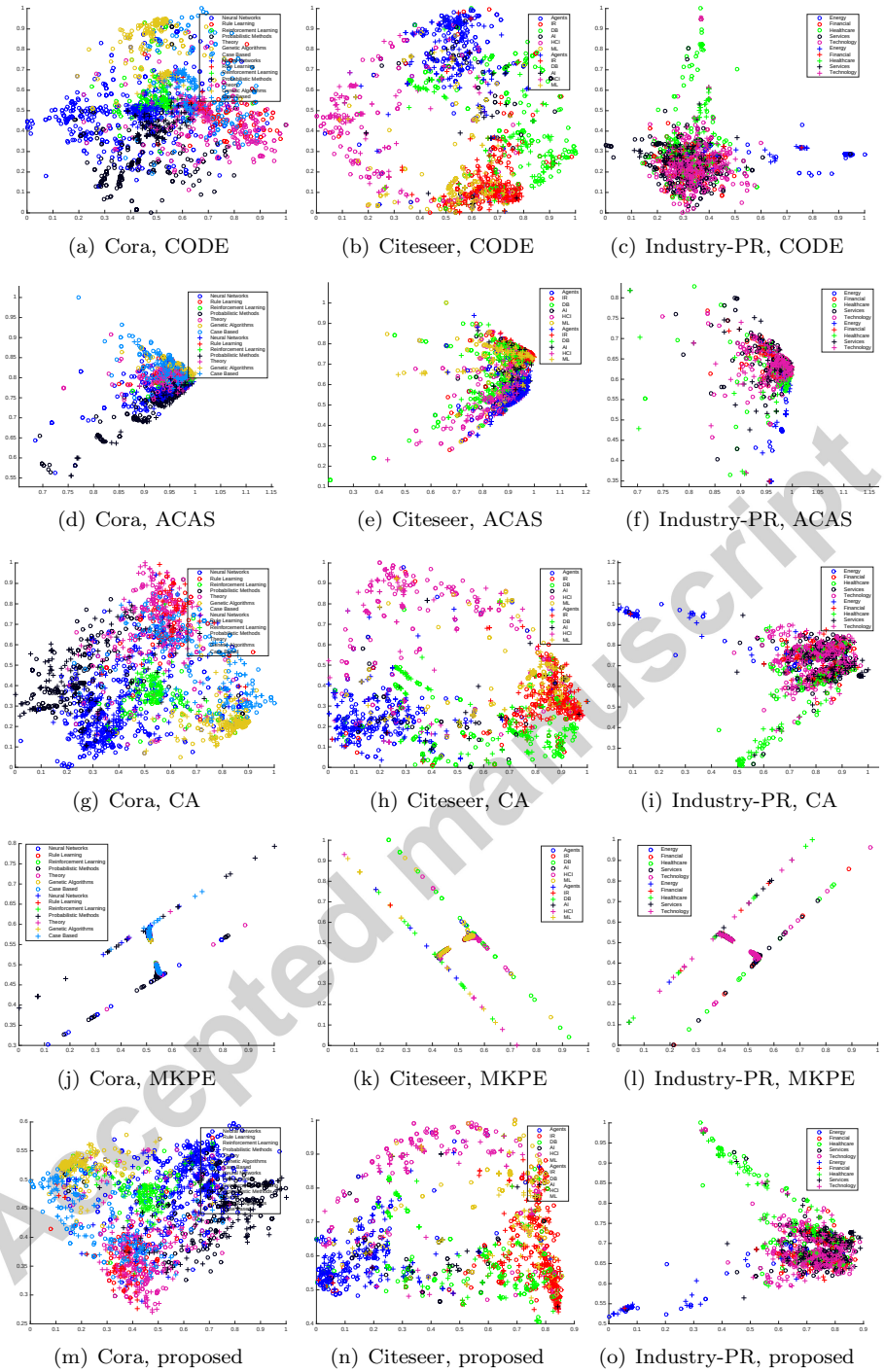
proposed method possesses lower or comparable error rates compared to existing ones. Overall, taking into account all the results demonstrated in Figures 5 and 6, the proposed method exhibits the most consistently good performance in preserving the relation measurements between documents and words.

4.3. Co-embedding Generation from Link Data

In this experiment, we assess the co-embedding algorithms using a different type of data, which provides link information between objects. We use three datasets to experiment with. One is the Cora dataset, which consists of 2,708 academic publications that are classified into one of the seven classes from case based, genetic algorithms, neural networks, probabilistic methods, reinforcement learning, rule learning and theory. The CiteSeer dataset contains 1,540 articles classified into one of six classes of agents from AI, DB, IR, ML and HCI. For both datasets, the citation links between the documents are provided. The third dataset is the Industry-PR, which contains 1,798 companies assigned to one of the 12 classes representing the 12 industry sectors of Yahoo!. Two companies are linked if they are mentioned by the same text documents among the PR Newswire press releases gathered from April 1st, 2003 to September 30th, 2003. For all the datasets, undirected links are studied, representing whether one cites the other in a document pair (for Cora and Citeseer), or whether two companies appear in the same text (for Industry-PR). We analyze objects that are included in the maximally connected subgraph of the given adjacency matrix, constructed from the link information. 40% of the objects from each class are randomly chosen and assigned to group $\mathcal{X}$, while the remaining ones are assigned to group $\mathcal{Y}$. The geodesic distance matrix $\mathbf{D}$ between the objects from the two groups is computed. Then, the Gaussian $e^{-\frac{\mathbf{D}_{ij}}{t}}$, with t denoting the average value of the elements in $\mathbf{D}$, is used to obtain the elements of the similarity matrix $\mathbf{R}$, which is finally used as the input to each of the three co-embedding algorithms.

For the quantitative evaluation, we employ a mean rank score [11] to examine how well the learned embeddings preserve the relation information in $\mathbf{R}$. The goal is to compare the ranks of the closeness between each row object and all the column objects based on $\mathbf{R}$, with the same closeness ranks, but obtained from the Euclidean distances between objects in the co-embedding space. For each row object, the new co-embedding-based ranks of its ten closest column objects searched within $\mathbf{R}$ are averaged. A smaller value of this mean rank indicates better preservation of the learned co-embeddings. The final score is computed by averaging the mean ranks for all the row objects.

In addition to this relation preservation evaluation based on rank, we also examine the compatibility between the generated co-embeddings and the class information associated with the row and column objects. The 10-fold cross-validation classification performance of both row and column objects with respect to their given class labels is computed using a simple one-nearest-neighbor classifier.



(a) Cora, CODE (b) Citeseer, CODE (c) Industry-PR, CODE

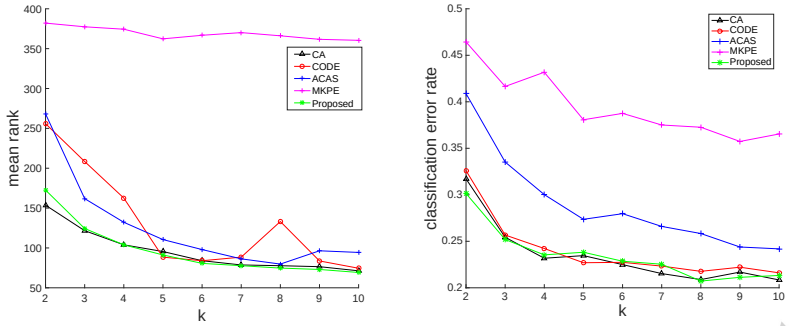
(d) Cora, ACAS (e) Citeseer, ACAS (f) Industry-PR, ACAS

(g) Cora, CA (h) Citeseer, CA (i) Industry-PR, CA

(j) Cora, MKPE (k) Citeseer, MKPE (l) Industry-PR, MKPE

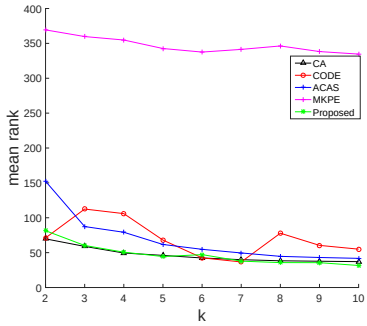
(m) Cora, proposed (n) Citeseer, proposed (o) Industry-PR, proposed

Figure 7: 2D co-embeddings generated by different algorithms, for the Cora and Citeseer datasets. Row objects (marked by "+") and column objects (marked by "∘") are members of different classes plotted in different colors.

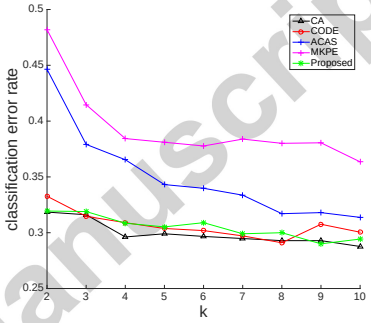


(a) Mean rank, Cora

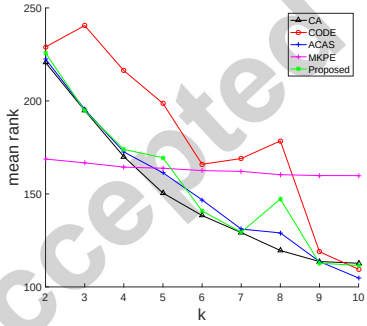
(b) Classification error, Cora



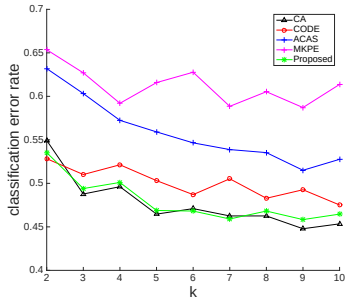
(c) Mean rank, Citeseer



(d) Classification error, Citeseer



(e) Mean rank, Industry-PR



(f) Classification error, Industry-PR

Figure 8: Quantitative comparison of different co-embedding algorithms in terms of the mean rank score and classification error rates using the three link datasets.

Firstly, we visually demonstrate the learned 2D co-embeddings for the three datasets in Figure 7. For all datasets, the proposed algorithm, CODE and CA produce more spread out co-embedding distributions and better class separability than ACAS. For Citeseer, it can be seen from Figures 7(b) and 7(k) that CODE and MKPE fail to preserve the between-group relations, as row and column objects from the same classes are shown to map far from each other. The proposed algorithm, CA and ACAS manage to map the row and column objects from the same class together.

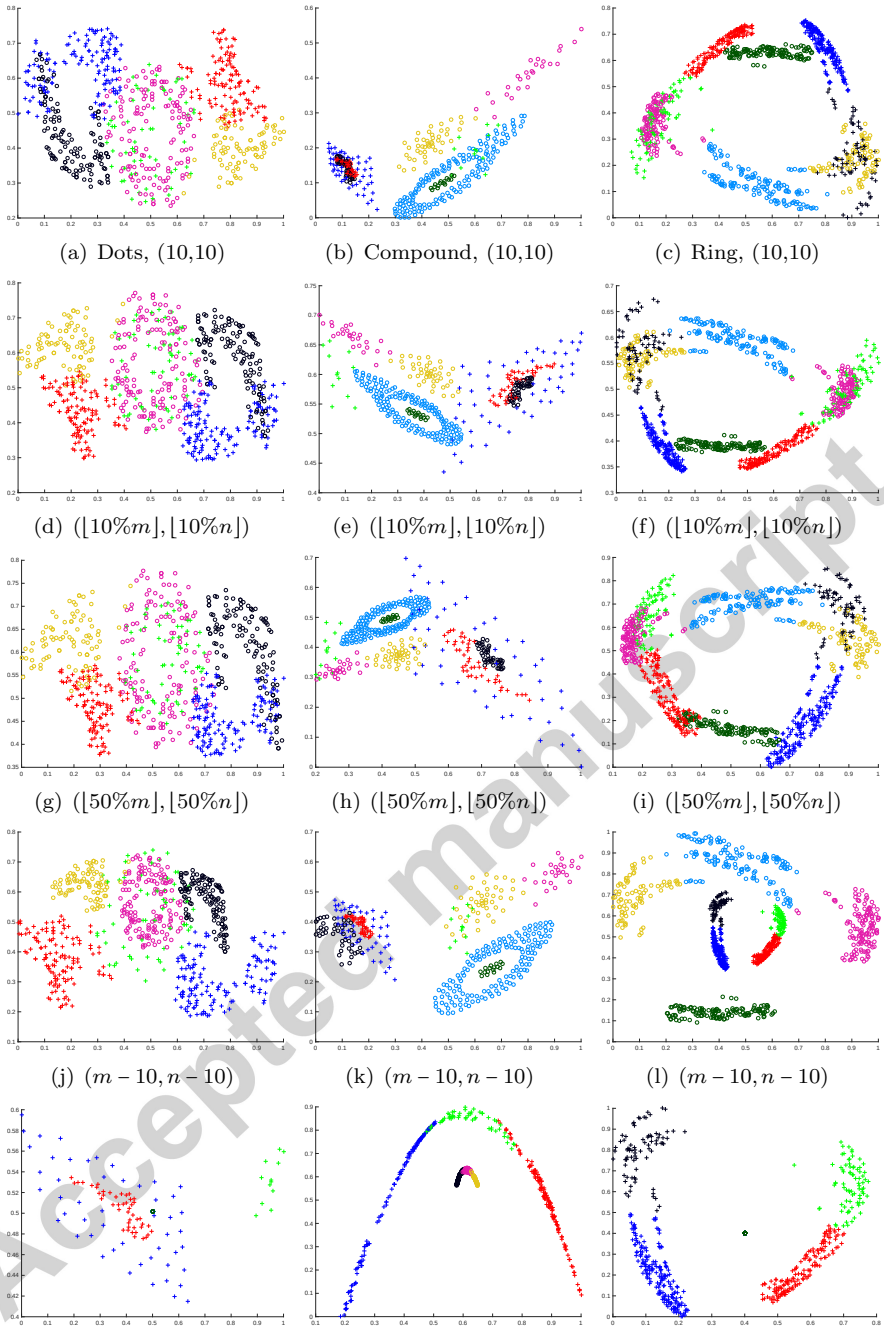
In Figure 8 we compare the three algorithms numerically, in terms of the mean rank scores and classification error rates for varying numbers of embedding dimensions k (from 2 to 10). It can be seen that the proposed algorithm and CA possess comparable performance, and both of them more frequently provide lower mean rank scores and classification errors than the other algorithms. This indicates better preservation of the input relation information in $\mathbf{R}$ and more compatible structure to the associated ground truth class information of the data objects.

4.4. Further Analysis of the Proposed Method

So far, the proposed algorithm provides the best performance for the synthetic datasets, while MKPE performs the worst (see Section 4.1). For document-word representation learning, the proposed algorithm and MKPE yield better quantitative performances, but MKPE provides separate document and word embedding distributions, which are less attractive for visualization purposes (see Section 4.2). With regard to link data evaluation, the proposed algorithm and CA provide better performances in terms of mean rank and classification error (see Section 4.3). Overall, the proposed algorithm is the only one that provides consistently good results for all the examined cases and under most evaluation criteria.

Here, we further compare these algorithms in terms of their used parameters. CA does not involve any parameter in its embedding computation, and CODE and MKPE do not employ parameters when constructing their objective or score functions. Both ACAS and the proposed algorithm employ parameters to control the balance between the local and global structure matching between the learned embedding and the input similarities; that is, the quantization parameter for ACAS, and the neighborhood parameters k_r and k_c for the proposed method. Sometimes, ACAS cannot accurately preserve the desired relation structure even with an exhaustive search over the quantization parameter, especially when processing data with complex geometric distributions (see Figures 2(e) and 3(b) for example). As we will show in the experiments below, although the proposed method employs two parameters, whereas CA, MKPE and CODE employ none, this does not limit its usability because these parameters can be set to small values without any performance sensitivity issues.

We investigate the effect of the neighborhood control parameters k_r and k_c in detail. In previous experiments, we employed small values of k_r and k_c, e.g., $k_r = k_c = 5$, because we observed that it is more reliable to preserve local neighborhood structure than to enforce a global matching that considers both



(a) Dots, (10,10) (b) Compound, (10,10) (c) Ring, (10,10)

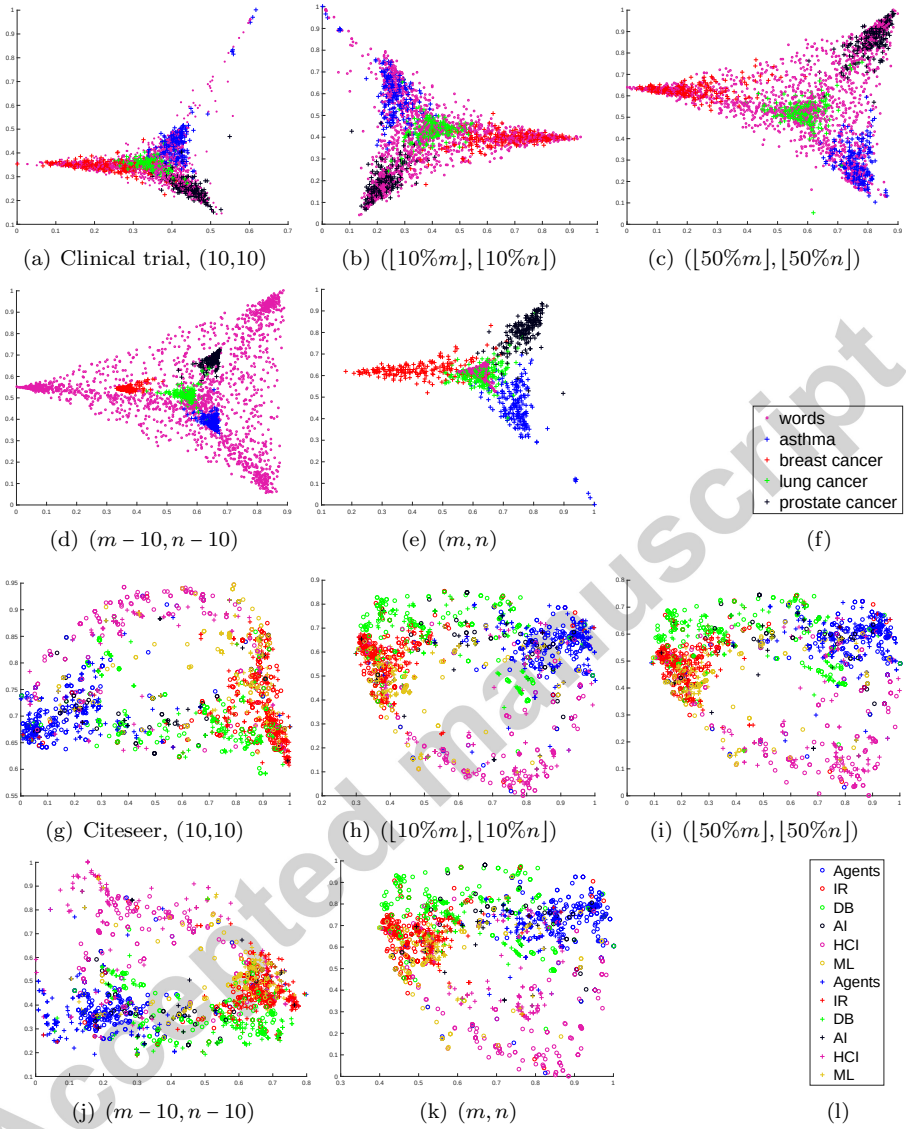
(d) $(\lfloor 10\%m \rfloor, \lfloor 10\%n \rfloor)$ (e) $(\lfloor 10\%m \rfloor, \lfloor 10\%n \rfloor)$ (f) $(\lfloor 10\%m \rfloor, \lfloor 10\%n \rfloor)$

(g) $(\lfloor 50\%m \rfloor, \lfloor 50\%n \rfloor)$ (h) $(\lfloor 50\%m \rfloor, \lfloor 50\%n \rfloor)$ (i) $(\lfloor 50\%m \rfloor, \lfloor 50\%n \rfloor)$

(j) $(m-10, n-10)$ (k) $(m-10, n-10)$ (l) $(m-10, n-10)$

(m) (m, n) (n) (m, n) (o) (m, n)

Figure 9: 2D embeddings generated by the proposed algorithm with varying settings of the neighborhood control parameters k_r and k_c using the three synthetic datasets dots, compound and ring (in each corresponding column).



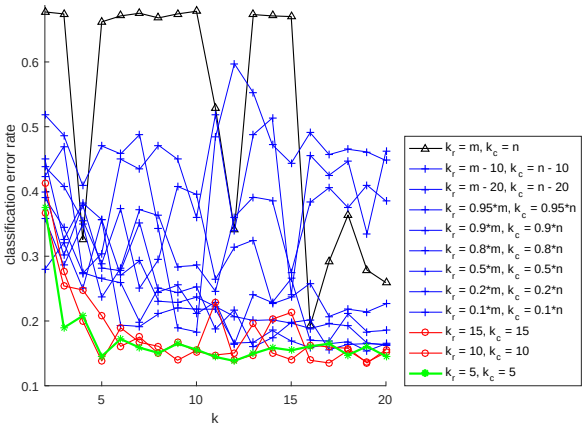
(a) Clinical trial, (10,10) (b) $([10\%m], [10\%n])$ (c) $([50\%m], [50\%n])$

(d) $(m-10, n-10)$ (e) (m, n) (f)

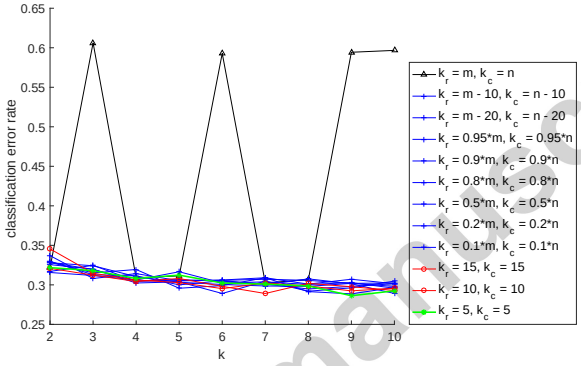
(g) Citeseer, (10,10) (h) $([10\%m], [10\%n])$ (i) $([50\%m], [50\%n])$

(j) $(m-10, n-10)$ (k) (m, n) (l)

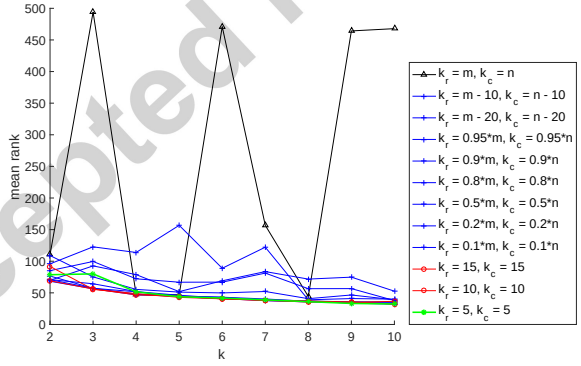
Figure 10: 2D embeddings generated by the proposed algorithm with varying settings of the neighborhood control parameters k_r and k_c using the: (a-e) 4-class clinical trials, and (f-j) Citeseer dataset.



(a) Clinical trials (9-class), classification error



(b) Citeseer, classification error



(c) Citeseer, mean rank

Figure 11: Performance comparison of the proposed algorithm under varying settings of the neighborhood control parameters k_r and k_c. (a) Classification error rates using the whole clinical trial collection. (b,c) Classification error rate and mean rank values using the Citeseer data. The experimented settings of k_r and k_c are shown in the legends. The typical setting of $k_r = k_c = 5$ and similar local settings of $k_r = k_c = 10$ and $k_r = k_c = 15$, as well as the worst setting of $k_r = m$ and $k_c = n$ are also included.

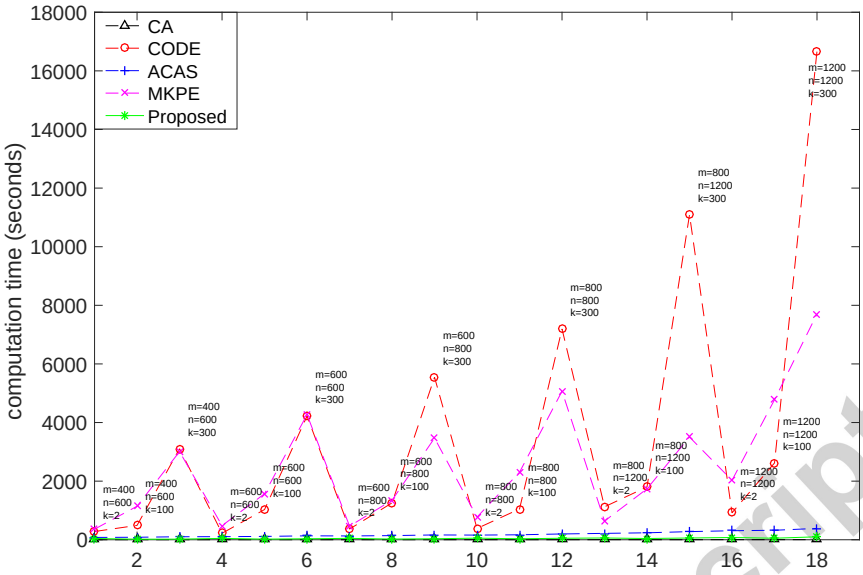


Figure 12: Computational cost comparison of different methods for increasing number of data size (n, m) and embedding dimension (k).

proximate and distant objects. To demonstrate the effect of varying settings of k_r and k_c, in Figures 9 and 10 we illustrate the change of 2D distributions of the learned embeddings using three synthetic datasets (dots, compound, ring), one text (4-class clinical trial) and one link (Citeseer) dataset. For illustration purposes, we select example datasets exhibiting more complex pattern distributions. The experimented settings include $k_r = k_c = 10$, also $k_r = \lfloor pm \rfloor$ and $k_c = \lfloor pn \rfloor$ ($\lfloor \cdot \rfloor$ denotes the floor function) with $p \in \{10\%, 50\%, 100\%\}$ controlling the percentage of the row (or column) objects to be included as neighbors, as well as $k_r = m - 10$ and $k_c = n - 10$ to exemplify the case of large neighborhood ranges that is close to the extreme case of all the objects being considered.

We compare Figures 9 and 10 and those reported in previous sections using the $k_r = k_c = 5$ setting. It can be seen that there is not much difference between embedding distributions obtained with smaller numbers of neighbors e.g., $k_r = k_c = 5$, $k_r = k_c = 10$ and $k_r = \lfloor 10\%m \rfloor$, $k_c = \lfloor 10\%n \rfloor$. For most datasets, the shape of the embedded data patterns starts to show significant distortion when large neighbors, e.g., $k_r = m - 10$ and $k_c = n - 10$, are used; for example Figure 9(l). For the extreme case of $k_r = m$ and $k_c = n$, the resulting embedding distribution collapses for most datasets. For example, in Figure 9(m) some classes almost shrink to single points, and in Figure 10(e), the words shrink to the center of documents.

Using the Citeseer dataset and the whole collection of clinical trials, we compare further parameter settings quantitatively in terms of classification errors

and mean ranks. The performances are displayed in Figure 11. It can be seen that the worst performance is most frequently obtained with $k_r = m$ and $k_c = n$. Also, the figure shows that as long as the neighborhood is kept small, similarly good performances can be obtained.

Finally, we compare the computational time of the proposed and existing methods in Figure 12. It can be seen that CODE and MKPE are more time consuming, particularly for learning embeddings with higher dimensions. The computational times of the remaining methods, that are mainly based on matrix decomposition, do not increase notably when embedding dimensions and data sizes increase. Both CA and our method are slightly more efficient than ACAS. Overall, the proposed method not only offers good performance for preserving relational structure, but also possesses competitive computational requirements.

5. Conclusion

We have proposed a novel method to generate co-embeddings of two different groups of objects within a joint embedding space for use in data visualization tasks, and also unsupervised and supervised machine learning setups. Co-embedding generation algorithms that simultaneously handle heterogeneous groups of data objects, are very important in many application areas, including biological network analysis, co-occurrence data analysis and information retrieval, and act as unique data analysis tools for sources supporting such complex multi-modal information.

The algorithm we have introduced is experimentally demonstrated to be very competitive with the existing state of the art. It holds minimal assumptions with respect to the data, as it does not require explicit knowledge of the initial objects, but instead only a set of similarities between objects from the two groups. Its underlying model is based on a set of weights that allow each embedding group to be defined in terms of the other, and it is shown that this permits the model optimization to be achieved via simple matrix factorization. Using a set of intuitive heuristics, we drastically reduce the number of model parameters needed for the generation of optimal co-embeddings. This is a also supported by a very effective model identification score we propose to search for the optimal parameters of the method.

Acknowledgment

This research has been supported by a PhD studentship jointly funded from the University of Liverpool and the China Scholarships Council. We thank the anonymous reviewers for the very useful comments and suggestions that have improved this manuscript.

References

[1] I. Jolliffe, *Principal Component Analysis*, ser. Springer Series in Statistics. New York: Springer-Verlag, 2002.

[2] X. He and P. Niyogi, "Locality preserving projections," in *NIPS*, vol. 16, 2003, pp. 234–241.

[3] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[4] R. Hettiarachchi and J. F. Peters, "Multi-manifold lle learning in pattern recognition," *Pattern Recognition*, vol. 48, no. 9, pp. 2947–2960, 2015.

[5] W. Zhang, X. Xue, H. Lu, and Y. Guo, "Discriminant neighborhood embedding for classification," *Pattern Recognition*, vol. 39, no. 11, pp. 2240–2243, 2006.

[6] C. Ding and L. Zhang, "Double adjacency graphs-based discriminant neighborhood embedding," *Pattern Recognition*, vol. 48, no. 5, pp. 1734–1742, 2015.

[7] Z. Zhang, M. Zhao, and T. W. S. Chow, "Constrained large margin local projection algorithms and extensions for multimodal dimensionality reduction," *Pattern Recognition*, vol. 45, no. 12, pp. 4466–4493, 2012.

[8] J. Weston, F. Ratle, and R.Collobert, "Deep learning via semi-supervised embedding," in *International Conference on Machine Learning, ICML*, 2008.

[9] I. Borg and P. J. F. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. Springer, Aug. 2005.

[10] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in Neural Information Processing Systems, NIPS*, 2013.

[11] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *In Proceedings of AAAI15*, 2015.

[12] Y. Yamanishi, "Supervised bipartite graph inference," in *Advances in Neural Information Processing Systems*, 2009, pp. 1841–1848.

[13] M. Gönen, "Embedding heterogeneous data by preserving multiple kernels," in *Proceedings of the 21st European Conference on Artificial Intelligence*, 2014.

[14] J. Choo, S. Bohn, G. Nakamura, A. M. White, and H. Park, "Heterogeneous Data Fusion via Space Alignment Using Nonmetric Multidimensional Scaling." in *SDM*. SIAM, 2012, pp. 177–188.

[15] T. Mu, J. Y. Goulermas, I. Korkontzelos, and S. Ananiadou, "Descriptive document clustering via discriminant learning in a co-embedded space of multilevel similarities," *Journal of the Association for Information Science and Technology*, vol. 67, no. 1, pp. 106–133, 2016.

[16] M. Khoshneshin, W. Street, and P. Srinivasan, "Bayesian Embedding of Co-occurrence Data for Query-Based Visualization," in *2011 10th International Conference on Machine Learning and Applications and Workshops (ICMLA)*, vol. 1, Dec. 2011, pp. 74–79.

[17] Y. Maron, E. Bienenstock, and M. James, "Sphere Embedding: An Application to Part-of-Speech Induction," in *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 1567–1575.

[18] M. J. Greenacre, *Theory and Applications of Correspondence Analysis*. Academic Press, 1984.

[19] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *JASIS*, vol. 41, no. 6, pp. 391–407, 1990.

[20] J. R. Bellegarda, "Latent semantic mapping," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 70–80, 2005.

[21] D. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[22] C. Lee, A. Elgammal, and M. Torki, "Learning representations from multiple manifolds," *Pattern Recognition*, vol. 50, pp. 74–87, 2016.

[23] T. Iwata, K. Saito, N. Ueda, S. Stromsten, T. L. Griffiths, and J. B. Tenenbaum, "Parametric embedding for class visualization," *Neural Computation*, vol. 19, no. 9, p. 25362556, 2007.

[24] T. Mu and J. Goulermas, "Automatic generation of co-embeddings from relational data with adaptive shaping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 10, pp. 2340–2356, Oct. 2013.

[25] A. Globerson, G. Chechik, F. Pereira, and N. Tishby, "Euclidean embedding of co-occurrence data," *Journal of Machine Learning Research*, vol. 8, no. 10, pp. 2265–2295, 2007.

[26] P. Sarkar, S. M. Siddiqi, and G. J. Gordon, "A latent space approach to dynamic embedding of co-occurrence data," in *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS 2007)*, 2007.

[27] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001, pp. 269–274.

[28] M. Rege, M. Dong, and F. Fotouhi, "Bipartite isoperimetric graph partitioning for data co-clustering," *Data Mining and Knowledge Discovery*, vol. 16, no. 3, pp. 276–312, 2008.

[29] N. Srebro, J. D. M. Rennie, and T. S. Jaakola, "Maximum-margin matrix factorization," in *Advances in Neural Information Processing Systems 17*. MIT Press, 2005, pp. 1329–1336.

[30] Y. Shen, R. Jin, J. Chen, X. He, J. Gao, and L. Deng, "A deep embedding model for co-occurrence learning," in *IEEE International Conference on Data Mining Workshop, ICDMW 2015, Atlantic City, NJ, USA, November 14-17, 2015*, 2015, pp. 631–638.

[31] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Computation*, vol. 15, pp. 1373–1396, 2002.

[32] M. Gönen, "Embedding Heterogeneous Data by Preserving Multiple Kernels," in *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014)*, 2014.

[33] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the Association of Computational Linguistics (ACL)*, 2004.

[34] S. A. Macskassy and F. Provost, "Classification in networked data: A toolkit and a univariate case study," *The Journal of Machine Learning Research*, vol. 8, pp. 935–983, 2007.