THE UNIVERSITY *of* LIVERPOOL

Embedding Approaches for Relational Data

Thesis submitted in accordance with the
requirements of the University of Liverpool
for the degree of Doctor of Philosophy

in

Electrical Engineering and Electronics

by

Yu Wu, B.Sc.(Eng.)

January 2018

Embedding Approaches for Relational Data

by

Yu Wu

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisors Dr. Tingting Mu and Dr. Yannis Goulermas, for their patient, encouragement, and professional instructions throughout this research. I could never progress toward the ability to reach to this stage without their consistent and illuminating support.

I would also like to extend my appreciation to my annual progress committee members: Dr. Xu Zhu and Prof. Simon Maskell, for their invaluable advice during my research. And my thanks also go to the Department of Electrical Engineering and Electronics at the University of Liverpool, for providing the research facilities that made it possible for me to carry out this research.

I would like to thank my colleagues, Austin Brockmeier, Alexandros Kostopoulos, Andrew Jones, Yanbin Hao, Xingjian Gao and Jinmeng Wu. For their wonderful camaraderie I enjoyed and many helpful discussions that challenged me to think deeper and wider about my research.

I offer my regards and blessings to all my friends in Liverpool for the experiences and pleasures we have had together. Special thanks to Peng Yin, Darcy, Ling Wang, Hang Dong, Benhong Xiao, Puiyi Cheung, Chaojie Liu, Ruiqi Zhang, Zhouxiang Fei and Zhihao Tian for all their wonderful camaraderie and suggestions during my time at the University of Liverpool. I am especially grateful to the English Corner members, Jo, Anna, Simon and Mike, for their support and advice in improving English.

Last but not least, I would like to extend my deepest gratitude to my family, especially to my parents Wanyong Wu and Xiaohong Zeng. Their unconditional love, belief, and support always encourage me and give me strength.

Abstract

Embedding methods for searching latent representations of the data are very important tools for unsupervised and supervised machine learning as well as information visualisation. Over the years, such methods have continually progressed towards the ability to capture and analyse the structure and latent characteristics of larger and more complex data. In this thesis, we examine the problem of developing efficient and reliable embedding methods for revealing, understanding, and exploiting the different aspects of the relational data. We split our work into three pieces, where each deals with a different relational data structure.

In the first part, we are handling with the weighted bipartite relational structure. Based on the relational measurements between two groups of heterogeneous objects, our goal is to generate low dimensional representations of these two different types of objects in a unified common space. We propose a novel 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 optimisation despatched using matrix decomposition. And we have also proposed a simple way of measuring the conformity between the original object relations and the ones re-estimated from the embeddings, in order to achieve model selection by identifying the optimal model parameters with a simple search procedure. We show that our proposed method achieves consistently better or on-par results on multiple synthetic datasets and real world ones from the text mining domain when compared with existing embedding generation approaches.

In the second part of this thesis, we focus on the multi-relational data, where objects are interlinked by various relation types. Embedding approaches are very popular in this field, they typically encode objects and relation types with hidden

representations and use the operations between them to compute the positive scalars corresponding to the linkages' likelihood score. In this work, we aim at further improving the existing embedding techniques by taking into account the multiple facets of the different patterns and behaviours of each relation type. To the best of our knowledge, this is the first latent representation model which considers relational representations to be dependent on the objects they relate in this field. The multimodality of the relation type over different objects is effectively formulated as a projection matrix over the space spanned by the object vectors. Two large benchmark knowledge bases are used to evaluate the performance with respect to the link prediction task. And a new test data partition scheme is proposed to offer a better understanding of the behaviour of a link prediction model.

In the last part of this thesis, a much more complex relational structure is considered. In particular, we aim at developing novel embedding methods for jointly modelling the linkage structure and objects' attributes. Traditionally, link prediction task is carried out on either the linkage structure or the objects' attributes, which does not aware of their semantic connections and is insufficient for handling the complex link prediction task. Thus, our goal in this work is to build a reliable model that can fuse both sources of information to improve the link prediction problem. The key idea of our approach is to encode both the linkage validities and the nodes neighbourhood information into embedding-based conditional probabilities. Another important aspect of our proposed algorithm is that we utilise a margin-based contrastive training process for encoding the linkage structure, which relies on a more appropriate assumption and dramatically reduces the number of training links. In the experiments, our proposed method indeed improves the link prediction performance on three citation/hyperlink datasets, when compared with those methods relying on only the nodes' attributes or the linkage structure, and it also achieves much better performances compared with the state-of-arts.

Declaration

The author hereby declares that this thesis is a record of work carried out in the Department of Electrical Engineering and Electronics at the University of Liverpool during the period from October 2013 to September 2017. The thesis is original in content except where otherwise indicated.

Contents

List of Figures

List of Tables

Abbreviations

SRL Statistical Relational Learning. 3, 4

MCMC Markov Chain Monte Carlo. 4

LLE Locality Linear Embedding. 12, 24, 27, 36

KG Knowledge Graph. 14, 16, 85–89, 101, 110, 136

LE Laplacian Eigenmaps. 17, 21, 22, 36, 50

SNE Stochastic Neighbour Embedding. 17, 27, 28, 36

PCA Principal Component Analysis. 18–20, 36

CCA Canonical Correlation Analysis. 29–32, 36, 38

RBM Restricted Boltzmann Machine. 33–36

CA Correspondence Analysis. 38, 46–48, 60, 64, 69, 76, 77, 82

CODE Co-occurrence Data Embedding. 38, 40–42, 60, 64, 69, 76, 77, 82

ACAS Automatic Co-embedding with Adaptive Shaping. 38, 48–50, 60, 64, 69, 76, 77, 82, 136

PSD Positive Semi-Definite. 41, 42, 120

BGP Bipartite Graph Partitioning. 42–45, 49

MKPE Multiple Kernel Preserving Embedding. 60, 64, 66, 69, 76, 77, 82

TransPES Translating on Pairwise Entity Space. 88, 105, 110

SE Structured Embedding. 89, 90

LFM Latent Factor Model. 89, 92

SME Semantic Matching Energy. 89, 92

LDA Latent Dirichlet Allocation. 115, 136, 138

MMSB Mixed Membership Stochastic Block. 115, 117

RTM Relational Topic Model. 117, 118, 129, 131

CESNA Communities from Edge Structure and Node Attributes. 118, 129, 131

SPML Structure Preserving Metric Learning. 120, 121, 129, 131

SPE Structure Preserving Embedding. 120, 121

SDP Semi-Definite Programming. 121

Chapter 1

Introduction

In many information domains, an object is usually characterised by a continuous or discrete feature vector of attributes, e.g., a scientific paper can be represented by its contextual words, references and key words; genes and gene products are often characterised at multiple levels including mRNA expression levels, protein abundance levels, cellular location and other factors; an image is characterised by its pixel intensity of colour channels or its associated text descriptions. With the attribute representations (or propositional representations [1]), traditional machine learning algorithms are concerned with learning a mapping from the input feature vectors to an output of interest, which may correspond to class labels, regression target values, clustering identifiers or intrinsic latent representations. Although many such attributes-oriented algorithms can generalise well (e.g., predict, learn the concepts) to new data from the problem domain, the rich information regarding the relationships between objects or attributes is ignored, which should be useful for uncovering, understanding, and exploiting the intrinsic properties of the data. For example, in natural language processing, the references for scientific papers are created with different types of motivations (e.g., relevant works, empirical findings, background reading), which are usually relevant to the central theme of the citing paper; in biology, the activation of a subset of genes or the protein domain interactions in a cellular compartment may correspond to a common functional process (e.g., cellular differentiation processes, protein synthesis processes, protein-protein interactions); in image processing and analysis, the visual contents in a scene are usually semantically interacted, and the salient interactions in the image are interpreted by its associated textual descriptions.

Indeed, learning to reason on the relationships is of vital importance in the physical and natural world. In physics, any of the four fundamental forces – gravitational, electromagnetic, strong, and weak – are considered as the ways that individual particles interact with each other [2]. It turns out that all physical aspects of the universe can be fully explained and linked together by these basic interactions. In a natural world, the organisms and their physical environment are linked together through processes of energy transfer and nutrient cycles. Understanding the vital connections between plants and animals and the world around them can provide us information about the benefits of ecosystems, manage our natural resources, and protect human health [3]. Moreover, the relational data can be very helpful for understanding and handling human generated big data, allowing one to infer and exploit latent properties, classes and structures about objects. Taking the email network as an example, which embodies the relations of sending and replying to messages, we are interested in exploiting the relational aspects of this data to uncover various latent properties (e.g., roles, communities and preferences) of the instances.[1] It can be noted that people who frequently receive messages containing relational types of assistant requests like photocopying, hotel bookings and meeting room arrangements are identified to have the latent role "administrative assistant". As another example, real world knowledge can be stored in a relational database, e.g., Semantic Web, Linked-Cloud and knowledge bases, and an inference engine can reason about the existing knowledge in the data to deduce new facts or highlight inconsistencies. Recently, the content of relational data is growing rapidly with the development of various areas such as web mining, bioinformatics, social network analysis and marketing.

Developing effective and reliable methods to deal with such unreliable, complex, and large-scale relational structure has been considered as one of the greatest challenges for today's machine learning. The first difficulty is that most relations are inherently vague and ambiguous. For example, in a movie rating database, viewers rate a movie via a score, of which the quantity is not precise at all for measuring

[1] We use the terms "instances", "entities" and "objects" interchangeably throughout this thesis.

their "degree" of inclination, but rather intuitive. And even if different people like the same movie, the reasons behind their motivations are usually very different: i.e., Jo may like Top Gun because she loves 80s action movies, while Felix likes Top Gun because he likes movies with Kenny Loggins soundtracks. So the fact that both viewers watched and rated the movie highly does not necessarily mean they will value the same set of other movies with high probabilities. On the other hand, most relational data are incomplete, noisy and including false information, a problem that is being aggravated due to the increasing usage of automatic information extraction techniques. Thus, a relational learning model also needs to be considered for improving the quality of a relational database, such as predicting unknown relationships, correcting existing relations, and detecting duplicate objects.

A worth mentioning field addressing the learning and inference from the uncertain and complex relational data is called *Statistical Relational Learning (SRL)*, which relies on a variety of statistical models that target relational learning tasks. Early SRL focuses on relational graphical models [1], which can be divided into two categories: **a)** logic-based (i.e., rule-based) models and **b)** frame-based (i.e., object-oriented) models. The logic-based models build upon combining the traditional *inductive logic programming* [4] methods for representing knowledge in the first-logic setting and graphical models for supporting probabilistic reasoning, i.e., Bayesian logic programs [5], stochastic logic programs [6] and Markov logic networks [7]. Frame-based models extend the traditional graphical models, such as Bayesian networks and Markov networks, avoiding their underlying i.i.d. assumption by incorporating the relational database models. For example, probability relational models [8, 9] define a probability model based on a relational database. The directed acyclic probability entity relationship model [10] adapts Bayesian analysis to entity-relationship database representation [11]. The relational Markov network [12] and relational dependency networks [13] are the relational extensions to the Markov networks. In these relational graphical models, objects and attributes are encoded by random variables and the statistical dependencies between these variables are either built from prior knowledge or inferred from the data automatically. To enable effective training, a variety of approximate inference methods has been utilised, such as variational inference, loopy belief propagation and Markov Chain Monte Carlo (MCMC) methods [14–16]. However, these models still remain highly expensive to train since they require a large number of statistical dependencies to build the dependency structure to link together the objects. If the required dependency structure is unknown, it has to be inferred automatically from the data, which is often very complex and time-consuming [17–19]. Hence, it is impractical to apply these relational graphical models to large-scale relational learning problems.

For effectively handling the relational structure, recent developments in SRL have devoted to latent variable models [20–22]. Unlike the relational graphical model, the statistical dependencies in these models are not solely explained using variables that have been observed in the data. Latent variable models assume that there are hidden causes for the observable data and model the probability of a particular relationship via either random generation processes or simple operations on some latent variables, i.e., variables that are not directly observed but are rather inferred from other observed variables. Simultaneously, the dependency structure is defined through only a small number of latent variables (e.g., group memberships, latent roles). As a result, such models do not suffer a loss of expressiveness and avoid the time-consuming structure learning that is necessary for the functioning of the previous models.

Embedding methods are latent variable models in which each entity is represented as a point in the latent space (e.g., Euclidean space), and a relation is modelled as the mathematical operation between these entity vectors. In multi-relational database [23, 24], they further assign each type of relationships to an operation that is characterised by vectors, matrices or tensors. These approaches seek a balance between the expressiveness and the complexity of their models and have been successfully applied to a range of relational learning problems, especially for the very large scale multi-relational data. Moreover, it has been shown that some latent vectors learned by these models coincidentally relate to the semantic relationships between entities. For an example, in text embedding [25], if you take the embedding vector of Paris, subtract the embedding vector of France, and add the embedding vector of Germany, the resulting embedding vector will be close to the embedding vector of Berlin.

In the following content of this chapter, we first provide a formal definition of relational data with its categorisations, properties and relevant relational learning tasks. Then we state our motivations and summarise the main contributions of this thesis. At last, we briefly describe how our thesis is organised.

1.1 Relational Data

1.1.1 Representation

All sorts of real world systems can be represented in a relational structured format in which the instances are linked (i.e., related) to each other. One could represent relational structure as networks (also referred to as graphs), whose nodes represent the objects and edges correspond to the connections between objects. Though the network model is insufficient for representing all relational structures (e.g., ternary relationships, n-ary relationships), it provides us a natural view of data by isolating entities and relationships. For example, the Internet is a big worldwide communication network where the nodes are computers and the edges are physical (or wireless) connections between the computers. The World Wide Web is a network where the webpages are nodes and hyperlinks are edges. Other examples include social networks of acquaintances, publication networks linked by citations, transportation networks with the flow of vehicles and metabolic networks of metabolic pathways. To illustrate the subtle details, we follow the mathematical definition of relational data in this chapter, based on the entity-relationship model [11].

A relational data set contains a set of entities and relationships. An entity is a "thing" which can be distinguished from other "things". It can exist physically or logically, i.e., a specific object, event or concept is an example of an entity. We should distinguish between an entity and an entity set, where an entity set is a category that the entities belong to. In other words, an entity is an instance of a given entity set. Let E_i denote the ith entity set. In general, the entity sets are not mutually disjoint. For instance, an entity in the entity set ”male-person” must also belong to the entity set ”person”. In this respect, entity set ”male-person” is a subset of the entity set ”person”.

Consider associations among entities. An n-ary relationship set $\mathcal{R}$, is a mathematical relation among n entities, each taken from an entity set:

$$\{(e_1, e_2, \ldots, e_n) \mid e_1 \in E_1 \wedge \ldots \wedge e_n \in E_n\} \tag{1.1.1}$$

and we refer to a single n-tuple $(e_1, e_2, \ldots, e_n)$ as a relationship[2] between the entities $e_1, e_2, \ldots, e_n$. Note that a relation can exist between the same set of entities. For example, a "marriage" is a relationship between two entities in the same entity set ”person”. And there may exist heterogeneous relations among the same set of entities. For example, besides the "friendship" relation, there may also exist an "officemate" relation between employees.

For an n-ary relation $\mathcal{R}$, its characteristic mapping function is defined accordingly, as

$$\phi_{\mathcal{R}} \,:\, E_1 \times E_2 \times \ldots \times E_n \mapsto \{0, 1\} \tag{1.1.2}$$

where × denote the Cartesian product of sets. Following this definitions, the boolean-valued characteristic function $\phi_{\mathcal{R}}$ gives true if and only if a particular relationship exists. In other cases, $\phi_{\mathcal{R}}$ can map each relationship to a real number, indicating the strength of the associated linkage. There are also observations or measurements information about the relationship or entity, which are usually expressed by a feature vector of attributes. For example, a person can be described by different attributes, such as inch, colour and height. Optionally, the relevant category information of entities can also be provided as attributes in a relational data.

In this thesis, we are only interested in binary relations, which only occur between two entity sets. Therefore, the characteristic mapping values of any particular relation for all possible entity pairs can be represented by a matrix (we referred to it as a characteristic matrix). For example, the ”friendship” relation between employees can be represented by a matrix, where the ijth entry indicates whether employee i is a friend of employee j or not. We denote the characteristic matrix for the kth relation $\mathcal{R}_k$ as $\mathbf{A}_k$, in the context of a single relation $\mathcal{R}$, we overload it by a simple matrix $\mathbf{A}$.

In general, a relational data can be very complex, e.g., including multiple relation sets and entity sets, and each element of these sets may be described by different

[2] We distinguish "a relationship" and "a relation" in this thesis. A relationship refers to an n-ary link between instances, while a relation refers to a set of relationships between the entity sets.

attribute sets. Simultaneously handling all sorts of different information in a relational data is a very challenge task since there may have multiple levels of uncertainty about the data, i.e., uncertainty about the number of attributes, attribute's type, and the identity of an object as well as relationship membership, attribute and type. Moreover, collecting a complete description of every entity and relationship in the data is generally impractical. It is thus necessary and important to study only the partially available information within a relational database. Therefore, relational data is divided into simpler data formats, each of which is covered by different subfields of relational learning. We address some important categories of the relational data as following:

- **Undirected graph**: An undirected relation is symmetric in that its characteristic matrix satisfies $\mathbf{A}(i,j) = \mathbf{A}(j,i)$ for any entity pair i and j. For example, a "marriage" relation is an undirected relation as if person i is married to person j, person j is also married to person i. Similarity or dissimilarity information can be viewed as undirected relations, given objects characterised by such information, Multidimensional Scaling [26] and its variants attempt to model such information as distances among points in a geometric space.

- **Directed graph**: A direct relation is asymmetric and it typically has an asymmetric characteristic matrix $\mathbf{A}$ for describing directional relationships in a data set. Examples include paper citation relations, hyperlink relations between webpages and sending/replying relations in an email network. The directed network data is usually served as the base information for more complex relational data, it is thus very important to uncover the interdependency information between observations in this simple network. For this type of data, a mixed-membership model [22] is able to capture the multiple roles that objects exhibit in interactions with others in a friendship network and a protein interaction data.

- **Bipartite graph**: The relational measurements in a bipartite network are between two groups of heterogeneous objects. It can be either undirected or directed — such as co-occurrence rates of articles and words in text data or item ratings given by users in a recommendation system. Commonly co-occurrence data learning approaches are topic modelling [27] and joint embeddings [28]. Collaborative filtering [29, 30] is a field for processing the preference (rating) data that is collected from many users. In general, the propositional data can also be generalised to bipartite relations if we treat the attributes as another set of entities.

- **Multi-relational data**: Relational data typically consists of several types of relations among entities. Handling multi-relations is essential for identifying valid, novel, useful, and understandable patterns from large datasets. Recently, a large number of works regarding multi-relational data learning have been proposed (see [31] for a review), based on learning the semantic embeddings of the structured text.

- **Document Network**: By document network, we refer to the kind of relational data where the objects are described by a single type of relations associated with objects' feature representations. Specifically, the attributes for representing objects are fixed and defined in the same homogeneous set. As such, the feature vectors for all the objects have the same dimensionality. The fusion of these two sources of information has proven to enhance the models' ability for classification [32] and link prediction along with improved latent representations [33]. A small example of the document network data is provided in Figure. 1.1.

There are other types of relational data with more components and more complex dependency structure. For examples, [34, 35] included the relations' associated text to better identify the group memberships in a network. In [36, 37], the object labels are utilised to derive a combined classification of the network data. Also, different dependency patterns have been explored such as collaborations in co-authorship networks, which are jointly modelled with the co-occurrence terms in a text body to refine the discovery of abstract "topics" [38].

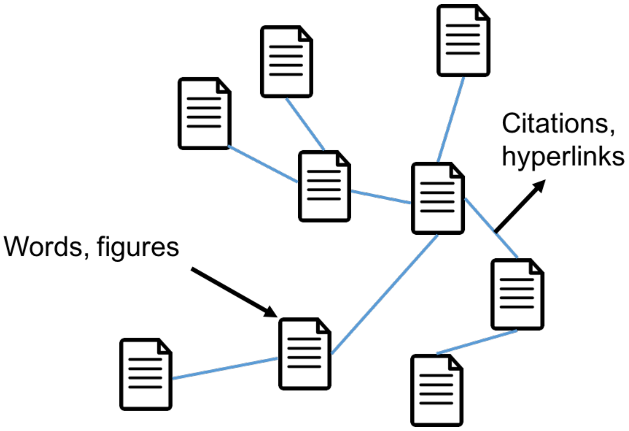


Figure 1.1: In a document network, each document contains its contents (such as figures, tables and text contents) as well as its linkages (or citations) to other documents.

.

1.1.2 Properties

Unlike the propositional data representations, the objects in a relational structure are inter-linked to each other. Without prior knowledge, such very complex interdependence structures are very difficult to exploit. However, two primary patterns are shown to prevail and structure the ties of many network data, namely, homophily and stochastic equivalence, which should be useful for developing relational learning algorithms. We describe these two patterns and illustrate how they can be exploited by relational learning methods as below.

- **Homophily**: The principle of homophily indicates that the relationships between similar objects are stronger than the relationships among dissimilar objects, which is also well-known as in the proverb "birds of a feather flock together". For example, people tend to make friends with regards to similar interests, ages, and analogous many other characteristics. It has been discovered in a large-scale of literature from the analysis of social networks [39]. Homophily provides the predictive patterns for relational data, such as predicting the religion of a person from the religions of his or her friends in a social network. In designing relational learning algorithms, the homophily is captured by either the similarity of observable/hidden object representations or the connectivity patterns of nodes in a network.

- **Stochastic Equivalence**: Sometimes, the individual object may exhibit its relationships to other objects that are characterised from the *category memberships* they belong to. All objects of the same group have similar relationships to objects of other groups. This property is referred to as stochastic equivalence in network analysis, where objects' relationships are explained via relationships between their associated group memberships. Analysis of stochastic equivalence in relational data can be very helpful for predicting unknown relationships between entities for which the memberships of these entities are known. In relational learning models, this feature is either explicitly captured by assigning latent classes or roles to entities in relational data or implicitly conveyed through the clusters/locations of the entities' latent representations.

1.1.3 Relational Learning Tasks

Various tasks can be brought for relational data learning, which are listed as follows:

- **Constructing Latent Representation**: Inferring the latent representations from the observed relational structure is a fundamentally important aspect in many disciplines, including economics, medicine, bioinformatics, natural language processing, management and social sciences. It serves to reduce the dimensionality of the data and can help us to understand the observed data by revealing its underlying concepts, i.e., sometimes latent variables correspond to aspects of physical reality which could in principle be measured or correspond to abstract concepts, like categories, functionalities and hidden states. In particular, embedding objects, relations, or both in a two or three dimensional geometric space allows us to gain a quick and intuitive summary of the data.

- **Link Prediction**: Link prediction concerns about learning the boolean-valued characteristic function $\phi_{\mathfrak{R}}$ for determining the existence of certain relations. It require the relational learning algorithms to understand the observed data based on entity attributes, relationships, and other information so that they can be used to predict the likelihood of novel associations between entities. Moreover, it has been shown in [33] that their model can even predict the existence of the words given only the links to existing articles of an new one. Alternatively, the link prediction problem has been transformed into an entity ranking task by many relational learning methods [40, 41], which ranks the objects to a query object according to their relatedness. Link prediction is central to almost all types of relational data.

- **Collective Classification/ Clustering**: Classifying or labelling for objects is a very important step in many application domains. These tasks are typically carried out on each object independently in standard classification/clustering setting, without considering the underlying relations between objects. While relations undoubtedly provide valuable information for classification or clustering, i.e., homophily in race and ethnicity naturally creates strong clustering patterns in our social network. Thus, relational learning approaches have an advantage over traditional approaches since they can improve the classification accuracy or clustering quality by including the relevant relational structure information. Numerous approaches [12, 36, 42–44] have brought clustering and classification into a relational setting, and have been shown to improve the learning results significantly.

- **Object Identification**: object identification is the problem of assigning an object instance with a unique, unchanging identity. It is also known as record linkage [45], entity resolution [46], instance matching [47] and data deduplication [48] in other naming conventions. Object identification clearly has many applications. For example, in the context of word sense disambiguation, there are eight senses of ”bass” in the large lexical database WordNet [49]. Given this word in a text, object identification is find the correct choice of meaning for it in that particular context. In the database domain, object identification is required when sharing data and resources within and across organisations, as it may be the case due to their differences in record shape, storage location, or curator style or preference. It is undeniable that entities are distinguishable via their patterns of connections for the reason that identical entities are expected to exhibit the same pattern of relations. Thus, object identification is made collectively by integrating various information in the relational learning setting, which has been researched in various data domains [50–53].

1.2 Motivation and Main Contributions

Methods for the generation of embeddings or pattern representations of data objects in low-dimensional spaces have been widely studied in conventional machine learning. Given the objects characterised by the feature vectors of attributes, numerous dimensionality reduction approaches [54–57] can be employed to learn the low-dimensional representation of these objects. Over the years, variants of these classical methods have been developed towards the ability to capture and analyse the structure and latent characteristics of larger and more complex datasets. For example, the classical dimensionality reduction method Locality Linear Embedding (LLE) [58] has been extended by [59] to process multiclass data, versions of discriminant embedding generation [60, 61], and projection methods are developed for processing multimodal data [62].

Embedding approaches have also been used for analysing various types of relational data, such as Multidimensional Scaling [26] for preserving pairwise distances of the original patterns in the low-dimensional space, joint embedding methods [63, 64] for heterogeneous bipartite data and many tensor factorisation [65–67] or energy-based learning models [23, 24, 68, 69] for encoding the multi-relational data into embedded points. Indeed, learning the representations of objects and relations in a low dimensional space gives us an easy and flexible way for implementing various relational learning tasks. In link prediction, link validities can be converted into mathematical operations between these latent vector representations; classification or clustering is simply conducted on the embedded points based on their pairwise neighbourhood relationships; the likelihood that two entities are identical is derived from whether they are close enough in the latent embedding space.

Hence, embedding approaches provide an alternative way to solve the relational learning tasks. One promising advantage for embedding methods compared with aforementioned statistical models (e.g., relational graphical models, probabilistic latent variable models) is that the optimisation of such methods can be delivered efficiently and explicitly through either matrix decomposition or gradient-based methods. In comparison, exact learning and inference are computationally intractable in those statistical probability models for big data that they have to utilise various approximate inference methods for achieving a reasonable computation time, which comes at a cost in terms of the model stability and accuracy. In practical, relational data may correspond to massive volumes of relations that it is prohibitive to apply those expensive inference models for processing them. Therefore, it is necessary to develop novel and efficient embedding algorithms for processing large-scale relational datasets.

In this thesis, we take the embedding approaches for handling three different relational data types — bipartite graph, multi-relational data and document network (see Section 1.1.1). Accordingly, three new models are developed for processing each data type and these models are briefly introduced as follows:

- **Heterogeneous Object Co-Embeddings**: In Chapter 3, 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 (it is thus a weighted bipartite graph). 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 optimisation 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.

- **Knowledge Graph Embedding**: In Chapter 4, we consider to model the Knowledge Graph (KG) [70], which is a particular type of multi-relational data. A KG stores information in a graph structured format, such as a directed graph whose nodes (entities) represent the objects and edges (links) correspond to the relation types between objects. It has become a very important resource to support many AI related applications, i.e., word-sense disambiguation [71] [72], search engine [73] [74], question answering [75]. Given that most KGs are noisy and far from being complete, KG analysis and completion is required to establish the likely truth of new facts and correct unlikely ones based on the existing data within the KG. An effective way for tackling this is through translation techniques which encode entities and links with hidden representations in embedding spaces. We aim at improving the existing translation techniques by taking into account the multiple facets of the different patterns and behaviours of each relation type. By considering relational representations to be dependent on the entities they relate, the multi-modality of the relation type over different entities is automatically and effectively formulated as a projection matrix over the space spanned by the entity vectors. A new test data partition scheme is also proposed to offer a better understanding of the behaviour of a link prediction model.

- **Link Prediction in Document Network Data**: In Chapter 5, we deal with the very imbalanced document network data. In the proposed approach, we encode both the linkage validities and the nodes neighbourhood information into embedding-based conditional probabilities. And the conformity between the document network and the embedding-based conditional probabilities are trained by combining two objective functions, one is a contrastive margin-based criterion for aligning the conditional probability distribution with the network structure, the other is a Kullback-Leibler divergence measuring the mismatch between the attribute representation distribution and the embedding data distribution. By combining the information of a linkage network and nodes' attributes, our proposed method not only improves the link prediction performance over methods using purely the nodes' attribute data or the linkage network data, but also gives good predictive performance when only the nodes' attribute data is provided.

1.3 Thesis Outline and Related Publications

We organise the thesis as follows:

CHAPTER 2 deals with the preliminaries by introducing various generic embedding approaches for dimensionality reduction. The chosen methods studied in this chapter represent the basic concepts and methodologies in this area. And in fact, most existing works in relational data learning are largely based on recognising some of these ideas and create novel extensions to them.

CHAPTER 3 presents a co-embeddings generation model for the analysis of arbitrary relational information between heterogeneous objects (e.g., co-occurrence rates between documents and terms). This model is highly efficient due to simple matrix decomposition and a small set of parameters. The capabilities of the proposed model are demonstrated for use in various machine learning tasks, and are compared to existing algorithms with multiple synthetic and real-world datasets from the text mining domain.

CHAPTER 4 introduces an embedding model specifically for multi-relational data analysis in the text domain. It is based on the hypothesis that the relationships of the same type in a multi-relational graph should possess distinct but related representations when associate with different node pairs. To reflect this hypothesis, we encode every relationship as a translation vector between entities, and relate every relationship vector to their associated relation label via simple projection operators in the modelling. Our model requires only a minimal parameterisation and provides a better model interpretability. Performance comparison with the state-of-art relational learning algorithms and in-depth analysis of the algorithm It superiority is demonstrated in various performance comparisons and deep analysis on two large KGs.

CHAPTER 5 studies the document network data. We propose an embedding-based method that encodes both the linkage network and the nodes' attribute representations into conditional probabilities. This model utilises a pairwise margin-based criterion that better respect the linkage structure and is consistently shown to achieve the best link prediction performance among all the comparing methods.

CHAPTER 6 concludes the whole thesis. We underline the contribution of this thesis, recapture the key ideas, and propose several potential directions for future research.

The publications produced from this research work are listed as follows:

- Wu, Y., Mu, T., Liatsis, P. and Goulermas, J.Y., 2017. Computation of heterogeneous object co-embeddings from relational measurements. *Pattern Recognition*, 65, pp.146-163.

- Wu, Y., Mu, T. and Goulermas, J.Y., 2017. Translating on pairwise entity space for knowledge graph embedding. *Neurocomputing*, 260, pp. 411-419.

Chapter 2

Generic Embedding Approaches

Embedding approaches give each datapoint a location in a lower dimensional space while preserving as much of the significant structure of the original high-dimensional data as possible. In this chapter, we explore the generic embedding models since they are building blocks for developing more complicated methods or to process more complex data structure. Almost all the relational learning algorithms are implicitly developed based on these generic embedding methods. For example, the methods present in Chapter 3 and Chapter 5 in this thesis are partly dependent on the ideas of the two generic embedding methods in this chapter, i.e., Laplacian Eigenmaps (LE) [56] and Stochastic Neighbour Embedding (SNE) [76]. Hence, it is of vital importance to investigate these generic methods in great detail.

Notations: From Section 2.1 to Section 2.4, we are given a set of datapoints (samples) $\{\boldsymbol{x}_i\}_{i=1}^n$ of dimension d, where $\boldsymbol{x}_i = [x_{i1}, \ldots, x_{id}]^\top$, the goal of these dimensionality reduction methods is to generate a set of embeddings $\{\boldsymbol{z}_i\}_{i=1}^n$ of dimension k ($k \ll d$), where $\boldsymbol{z}_i = [z_{i1}, z_{i2}, \ldots, z_{ik}]^\top$ so that the embedding datapoints in matrix $\mathbf{Z} = [z_{ij}]$ capture as much intrinsic structure of the original datapoint matrix $\mathbf{X} = [x_{ij}]$ as possible. For those methods in Section 2.5 and Section 2.6, different notations are used.

2.1 Principal Component Analysis

2.1.1 Model Construction

Principal Component Analysis (PCA) [77] is a linear method that seeks a projection from high dimensional data onto a lower dimensional space, such that the variance of the projected data is maximised. To begin with, we shall consider the simpler case of projecting the high dimensional data onto a line first. Let $\boldsymbol{w}$ be the unit vector in the direction of this line. Each data point $\boldsymbol{x}_i$ is projected onto this line with the location given by the scalar $\boldsymbol{w}^\top \boldsymbol{x}_i$. Note that the mean of the projected data points is calculated by

$$\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{w}^\top \boldsymbol{x}_i = \boldsymbol{w}^\top \bar{\boldsymbol{x}}, \tag{2.1.1}$$

where the d dimensional vector $\bar{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$ is the sample set mean.

So the variance of the projected data is given by

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{w}^\top \bar{\boldsymbol{x}})^2 &= \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{w}^\top (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top \boldsymbol{w} \\ &= \boldsymbol{w}^\top \left[\frac{1}{n}(\mathbf{X} - \frac{1}{n}\mathbf{1}_{n\times n}\mathbf{X})^\top (\mathbf{X} - \frac{1}{n}\mathbf{1}_{n\times n}\mathbf{X})\right] \boldsymbol{w} \\ &= \boldsymbol{w}^\top \mathbf{S} \boldsymbol{w}, \end{aligned} \tag{2.1.2}$$

where S is the empirical sample covariance matrix defined by

$$\mathbf{S} = \frac{1}{n}(\mathbf{X} - \frac{1}{n}\mathbf{1}_{n\times n}\mathbf{X})^\top (\mathbf{X} - \frac{1}{n}\mathbf{1}_{n\times n}\mathbf{X}). \tag{2.1.3}$$

Thus, the projected variance along the direction on $\boldsymbol{w}$ is given by $\boldsymbol{w}^\top \mathbf{S} \boldsymbol{w}$. To maximise it with regards to the projection vector $\boldsymbol{w}$, we are aware of the normalisation constraint $\boldsymbol{w}^\top \boldsymbol{w} = 1$. So the Lagrangian function for this constrained optimisation problem is defined as

$$\mathcal{L}(\boldsymbol{w}, \lambda_1) = \boldsymbol{w}^\top \mathbf{S} \boldsymbol{w} + \lambda_1(\boldsymbol{w}^\top \boldsymbol{w} - 1) \tag{2.1.4}$$

By setting the derivative with respect to $\boldsymbol{w}$ equal to zero, we have

$$\mathbf{S}\boldsymbol{w} = \lambda_1 \boldsymbol{w} \tag{2.1.5}$$

Thus, all the stationary points $(\lambda_1, \boldsymbol{w})$ for this optimisation problem are eigenvalue and eigenvector pairs of the covariance matrix S. The associated maximal quantity of the variance is

$$\boldsymbol{w}^\top \mathbf{S}\boldsymbol{w} = \boldsymbol{w}^\top \lambda_1 \boldsymbol{w} = \lambda_1, \tag{2.1.6}$$

so the variance will be maximised if $\boldsymbol{w}$ equals to the eigenvector corresponding to the largest eigenvalue λ_1, of the covariance matrix $\mathbf{S}$. This eigenvector is known as the first principal component.

2.1.2 Multidimensional Extension

In the above section, PCA only considers projecting data onto a one-dimensional space while preserves as much data variance as possible. Now, suppose the data sample is projected onto a k dimensional subspace, and assume that the columns of $\mathbf{W}_k = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k]$ forms an orthogonal basis for this subspace. This indicates that

$$\mathbf{W}_k^\top \mathbf{W}_k = \mathbf{I}_k, \tag{2.1.7}$$

where $\mathbf{I}_k$ is the identity matrix of size k.

Similar to the above section, PCA maximises a "projected variance" onto this space which is defined as the sum of variances in each axis. As seen from Eq. (2.1.2), it is given by

$$\sum_{i=1}^{k} \boldsymbol{w}_i^\top \mathbf{S}\boldsymbol{w}_i = \mathrm{Tr}(\mathbf{W}_k^\top \mathbf{S}\mathbf{W}_k). \tag{2.1.8}$$

Thus, the k dimensional linear space is found by solving the following optimisation problem

$$\underset{\substack{\mathbf{W}_k^\top \mathbf{W}_k = \mathbf{I}_k, \\ \mathbf{W}_k \in \mathbb{R}^{n \times k}}}{\mathrm{argmax}} \ \mathrm{Tr}(\mathbf{W}_k^\top \mathbf{S}\mathbf{W}_k) \tag{2.1.9}$$

Notice that when $k = n$, we have $\mathrm{Tr}(\mathbf{W}_k^\top \mathbf{S}\mathbf{W}_k) = \mathrm{Tr}(\mathbf{S}\mathbf{W}_k\mathbf{W}_k^\top) = \mathrm{Tr}(\mathbf{S})$, it indicates that the total variance of the data is invariant with respect to any other orthogonal basis.

2.1.3 Eigenvector Solution

The trace optimisation problem in Eq. (2.1.9) is a well-known result of linear algebra that will be exploited repeatedly in this thesis. An overview of a variety of such problems is discussed and solved in [78]. We give an effective proof to derive the eigenvalue solution for this problem in this section.

Firstly, the empirical variance matrix $\mathbf{S}$ can be decomposed as

$$\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top, \tag{2.1.10}$$

where $\mathbf{\Lambda} = \mathrm{Diag}(\lambda_1, \ldots, \lambda_n)$ is a diagonal matrix whose entries are the eigenvalues of $\mathbf{S}$ and $\mathbf{U}$ is an orthogonal matrix containing the associated eigenvectors. Here, we assume λ_q decreases with increasing subscript q.

Let $\mathbf{P} = \mathbf{U}^\top \mathbf{W}_k$, it is easy to see that the columns of $\mathbf{P}$ still forms an orthogonal basis for a k dimensional subspace. We can equivalently rewrite the optimisation problem Eq. (2.1.9) with respect to $\mathbf{P}$ as

$$\underset{\substack{\mathbf{P}^\top\mathbf{P}=\mathbf{I}_k,\\ \mathbf{P}\in\mathbb{R}^{n\times k}}}{\mathrm{argmax}}\, \mathrm{Tr}(\mathbf{P}^\top\mathbf{\Lambda}\mathbf{P}). \tag{2.1.11}$$

And for this trace quantity, we can further prove that

$$\begin{aligned}\mathrm{Tr}(\mathbf{P}^\top\mathbf{\Lambda}\mathbf{P}) &= \sum_{i=1}^{n} \lambda_i (\sum_{j=1}^{k} p_{ij}^2) \\ &\leq \lambda_1 + \ldots + \lambda_k.\end{aligned} \tag{2.1.12}$$

This inequality is an immediate consequence of $\sum_{i=1}^{n}(\sum_{j=1}^{k} p_{ij}^2) = \mathrm{Tr}(\mathbf{P}^\top\mathbf{P}) = \mathrm{Tr}(\mathbf{I}_k) = k$ and $\sum_{j=1}^{k} p_{ij}^2 = \|\mathbf{P}^\top \boldsymbol{e}_i\|_2^2 \leq \|\boldsymbol{e}_i\|_2^2 = 1$ ($\boldsymbol{e}_i$ is the standard basis vector with 1 in the ith position), where the quantity $\|\mathbf{P}^\top \boldsymbol{e}_i\|_2^2$ has to be smaller than the squared length of $\boldsymbol{e}_i$ since it is the squared length of the projection of $\boldsymbol{e}_i$ onto the column space of $\mathbf{P}$. The equality sign holds if and only if the columns of $\mathbf{P}$ and the set of standard basis vector $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_k\}$ span the same subspace. Hence, $\mathbf{W}_k = \mathbf{U}\mathbf{P}$ is an orthonormal basis of the eigenspace associated with the first k eigenvalues.

In fact, PCA aligns each principal component with the eigenvectors. Let $\boldsymbol{u}_1$, $\boldsymbol{u}_{\mathbf{2}}, \ldots, \boldsymbol{u}_k$ be the eigenvectors corresponding to the k largest eigenvalues $\lambda_1, \ldots, \lambda_k$

of S. The qth principal component is computed as the qth eigenvector $\boldsymbol{u}_q$. Consequently, if we denote $\mathbf{U}_k = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k]$, then the associated k dimensional embedding is given by

$$\mathbf{Z} = \mathbf{X}\mathbf{U}_k. \tag{2.1.13}$$

2.2 Laplacian Eigenmaps

2.2.1 Model Construction

LE [56] is a locality neighbourhood preserving method that constructs the lower dimensional representation based on the data neighbourhood structure. It first builds a weighted undirected graph $G = (V, E)$ with edges connecting nearby points to each other. The weighted adjacency matrix $\mathbf{W} = [w_{ij}]$ of this graph should reflect the neighbourhood relationships between data points, which is chosen either as the Heat kernel with parameter t

$$w_{ij} = \begin{cases} e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{4t}} & \text{if } \|\boldsymbol{x}_i - \boldsymbol{x}_j\| < \xi \\ 0 & \text{otherwise} \end{cases}, \tag{2.2.1}$$

or simply

$$w_{ij} = \begin{cases} 1 & \text{if } \|\boldsymbol{x}_i - \boldsymbol{x}_j\| < \xi \\ 0 & \text{otherwise} \end{cases}. \tag{2.2.2}$$

To keep the neighbouring data points staying as close together as possible, LE chooses to minimise the following objective function

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2 w_{ij}, \tag{2.2.3}$$

under appropriate constraints. Intuitively, this objective function incurs a heavy penalty for mapping close data points i, j far apart since closer points refers to larger weight value w_{ij}. Thus, minimising such an objective function tends to keep the mapped points close when their high dimensional counterparts are close.

Let $\mathbf{D}$ be the degree matrix of the graph G with its diagonal entries defined as the column (or row, since $\mathbf{W}$ is symmetric) sums of $\mathbf{W}$. And the Laplacian matrix is $\mathbf{L} = \mathbf{D} - \mathbf{W}$, which is positive semi-definite. It turns out that Eq. (2.2.3) can be rewritten as

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2 w_{ij} &= \sum_{i=1}^{n}\sum_{j=1}^{n}(\boldsymbol{z}_i - \boldsymbol{z}_j)^\top(\boldsymbol{z}_i - \boldsymbol{z}_j)w_{ij} \\ &= \sum_{i}^{n}\boldsymbol{z}_i^\top\boldsymbol{z}_i(\sum_{j}^{n} w_{ij}) + \sum_{j}^{n}\boldsymbol{z}_j^\top\boldsymbol{z}_j(\sum_{i=1}^{n} w_{ij}) - 2\sum_{i=1}^{n}\sum_{j=1}^{n}\boldsymbol{z}_i^\top\boldsymbol{z}_j w_{ij} \\ &= 2\mathrm{Tr}(\mathbf{Z}^\top\mathbf{D}\mathbf{Z}) - 2\mathrm{Tr}(\mathbf{Z}^\top\mathbf{W}\mathbf{Z}) \\ &= 2\mathrm{Tr}(\mathbf{Z}^\top\mathbf{L}\mathbf{Z}). \qquad (2.2.4)\end{aligned}$$

We now minimised Eq. (2.2.4) with respect to the data embedding matrix $\mathbf{Z}$. Clearly, this has to a constrained minimisation to remove the arbitrary scaling factor for the mapped points in $\mathbf{Z}$. LE requires $\mathbf{Z}$ to be $\mathbf{D}$-orthogonal

$$\mathbf{Z}^\top\mathbf{D}\mathbf{Z} = \mathbf{I}. \qquad (2.2.5)$$

Also, LE removes the arbitrary translation factor by centring the embedded points with weights in $\mathbf{D}$

$$\mathbf{Z}^\top\mathbf{D}\mathbf{1} = 0, \qquad (2.2.6)$$

where $\mathbf{1}$ is n-length the vector of ones.

Finally, the minimisation problem reduces to finding

$$\underset{\substack{\mathbf{Z}^\top\mathbf{D}\mathbf{Z}=\mathbf{I},\\ \mathbf{Z}^\top\mathbf{D}\mathbf{1}=0,\ \mathbf{Z}\in\mathbb{R}^{n\times k}}}{\mathrm{argmin}} \quad \mathrm{Tr}(\mathbf{Z}^\top\mathbf{L}\mathbf{Z}). \qquad (2.2.7)$$

This optimisation problem can be solved in a similar manner to that of Eq. (2.1.9).

2.2.2 Solving the Constrained Optimisation Problem

We define the normalised graph Laplacian matrix $\tilde{L} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}$, it is easy to show that this real symmetric matrix is positive semi-definite. We denote its eigendecomposition as $\tilde{\mathbf{L}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ with $\boldsymbol{\Lambda} = \mathrm{Diag}(\lambda_1, \dots, \lambda_n)$ the diagonal matrix whose entries are the eigenvalues in ascending order and $\mathbf{U}$ is an orthogonal matrix containing the associated eigenvectors. Here, we assume λ_q *increases* with increasing subscript q. At first, we show that $\mathbf{D}^{\frac{1}{2}}\mathbf{1}$ is an eigenvector with eigenvalue 0 since

$$\begin{aligned}\tilde{\mathbf{L}}(\mathbf{D}^{\frac{1}{2}}\mathbf{1}) &= \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}^{\frac{1}{2}}\mathbf{1}) \\ &= \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{1} \\ &= \mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{1} \\ &= 0. \end{aligned} \tag{2.2.8}$$

And it is the only eigenvector for eigenvalue 0 if we assume a connected graph G [56].

Therefore, If we substitute $\mathbf{P} = \mathbf{U}^\top \mathbf{D}^{\frac{1}{2}}\mathbf{Z}$ into the Eq. (2.2.7), the condition of $\mathbf{Z}^\top\mathbf{D}\mathbf{1} = 0$ is converted as

$$\mathbf{P}^\top\mathbf{U}^\top\mathbf{D}^{\frac{1}{2}}\mathbf{1} = \mathbf{P}^\top\boldsymbol{e}_1 = 0, \tag{2.2.9}$$

where $\boldsymbol{e}_1$ is a unit vector with 1 in the first position and we have made use of the fact that $\lambda_1 = 0$ and all eigenvectors are orthogonal to derive this. Thus, to satisfy the condition $\mathbf{Z}^\top\mathbf{D}\mathbf{1} = 0$, we simply put the first row of $\mathbf{P}$ to be all zeros.

And Eq. (2.2.7) turns out to be

$$\underset{\substack{\mathbf{P}^\top\mathbf{P}=\mathbf{I},\\ \mathbf{P}^\top\boldsymbol{e}_1=0,\ \mathbf{P}\in\mathbb{R}^{n\times k}}}{\operatorname{argmin}} \quad \mathrm{Tr}(\mathbf{P}^\top\mathbf{\Lambda}\mathbf{P}) \tag{2.2.10}$$

Again, we have $\sum_{i=2}^{n}(\sum_{j=1}^{k} p_{ij}^2) = \mathrm{Tr}(\mathbf{P}^\top\mathbf{P}) = \mathrm{Tr}(\mathbf{I}_k) = k$, $\sum_{j=1}^{k} p_{ij}^2 = \|\mathbf{P}^\top\boldsymbol{e}_i\|_2^2 \le \|\boldsymbol{e}_i\|_2^2 = 1$, $\lambda_1 = 0$, the trace $\mathrm{Tr}(\mathbf{P}^\top\mathbf{\Lambda}\mathbf{P})$ can be rewritten as

$$\begin{aligned}\mathrm{Tr}(\mathbf{P}^\top\mathbf{\Lambda}\mathbf{P}) &= \sum_{i=2}^{n}\lambda_i(\sum_{j=1}^{k} p_{ij}^2) \\ &\ge \lambda_2 + \ldots + \lambda_{k+1}. \end{aligned} \tag{2.2.11}$$

The equality sign holds if and only if the columns of $\mathbf{P}$ and the set of standard basis vectors $\{\boldsymbol{e}_2, \ldots, \boldsymbol{e}_{k+1}\}$ span the same subspace. Hence, $\mathbf{D}^{\frac{1}{2}}\mathbf{Z} = \mathbf{U}\mathbf{P}$ is an orthonormal basis of the eigenspace associated with the 2nd to the $(k+1)$th smallest eigenvalues.

Finally, let $\mathbf{U}_k$ be the eigenvector matrix correspond to the 2nd to the $(k+1)$th smallest eigenvalues of $\tilde{\mathbf{L}}$, LE finds the k dimensional embedding as

$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}}\mathbf{U}_k. \tag{2.2.12}$$

2.3 Locality Linear Embedding

2.3.1 Model Construction

LLE [79] also assumes the data are sampled from some smooth underlying manifold, it then constructs the nonlinear neighbourhood preserving mapping by approximately reconstructing the locally linear geometry of the manifold in the lower dimensional space. Provided n points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ are well-sampled such that each data point and its neighbours are lying on or close to a locally linear patch of the manifold. In the formulation of LLE, the geometry of these patches are characterised by minimising the following cost function

$$\mathcal{E}(\mathbf{W}) = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \sum_{j=1}^{n} w_{ij}\boldsymbol{x}_j\|_2^2, \tag{2.3.1}$$

where the weight matrix $\mathbf{W} = [w_{ij}]$ is the linear coefficients that reconstruct each data point from its neighbours. Two constraints are imposed on $\mathbf{W}$: first, each data point $\boldsymbol{x}_i$ is reconstructed only from its neighbours, $w_{ij} = 0$ is enforced if $\boldsymbol{x}_j$ does not belong to this set; second, the rows of $\mathbf{W}$ must sum to one: $\sum_{j=1} w_{ij} = 1$. The second restriction ensures that the optimal weights $\mathbf{W}$ to the cost function $\mathcal{E}(\mathbf{W})$ is invariant to rotations, rescalings, and translations of the data. Consequently, the weights in $\mathbf{W}$ can reflect the intrinsic geometric properties of the manifold rather than properties that depend on a particular frame of reference.

After obtaining the optimal $\mathbf{W}$ that characterises the local geometry in the original data space, LLE expects to use it to reconstruct the embedded points in a lower dimensional space. In the same way to the previous cost function, the following objective function is chosen to minimise

$$\Phi(\mathbf{Z}) = \sum_{i=1}^{n} \|\boldsymbol{z}_i - \sum_{j=1}^{n} w_{ij}\boldsymbol{z}_j\|_2^2. \tag{2.3.2}$$

While opposed to optimising weights w_{ij} in Eq. (2.3.1), here the weights w_{ij} are fixed for finding the embedded points in $\mathbf{Z}$.

2.3.2 Computing the Weight Matrix

We first solve the constrained least squares problem for minimising $\mathcal{E}(\mathbf{W})$. In fact, we can minimise each term in $\mathcal{E}(\mathbf{W})$ separately for calculating the optimal weight matrix $\mathbf{W}$ row by row.

Therefore, we only need to consider a single patch with a data point $\boldsymbol{x}$, its K nearest neighbours $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_K$ and the reconstruction weights $\boldsymbol{w} = [w_j]$ that sum to one. The cost for this patch is written as

$$\begin{aligned} \xi &= \|\boldsymbol{x} - \sum_{j=1}^{K} w_j \boldsymbol{\eta}_j\|_2^2 \\ &= \|\sum_{j=1}^{K} w_j (\boldsymbol{x} - \boldsymbol{\eta}_j)\|_2^2 \\ &= \sum_{j=1}^{K} \sum_{l=1}^{K} w_j w_l (\boldsymbol{x} - \boldsymbol{\eta}_j)^\top (\boldsymbol{x} - \boldsymbol{\eta}_l) \\ &= \boldsymbol{w}^\top \mathbf{C} \boldsymbol{w}, \end{aligned} \tag{2.3.3}$$

where we have introduced the local covariance matrix $\mathbf{C} = [c_{jl}]$,

$$\mathbf{C} = \begin{bmatrix} (\boldsymbol{x} - \boldsymbol{\eta}_1)^\top \\ \vdots \\ (\boldsymbol{x} - \boldsymbol{\eta}_1)^\top \end{bmatrix} \Big[(\boldsymbol{x} - \boldsymbol{\eta}_1), \ldots, (\boldsymbol{x} - \boldsymbol{\eta}_1) \Big]. \tag{2.3.4}$$

From its construction, we see that this covariance matrix is symmetric and positive semidefinite.

To enforce the constraint $\mathbf{1}^\top \boldsymbol{w} = 1$ ($\mathbf{1}$ is the vector with 1 in each position), the Lagrangian function is defined as

$$\mathcal{L}(\boldsymbol{w}, \mu) = \boldsymbol{w}^\top \mathbf{C} \boldsymbol{w} - \mu(\mathbf{1}^\top \boldsymbol{w} - 1), \tag{2.3.5}$$

where μ is the associated Lagrange multiplier.

Differential $\mathcal{L}(\boldsymbol{w}, \mu)$ with respect to μ, we have

$$\mathbf{C}\boldsymbol{w} = \frac{\mu}{2}\mathbf{1} \tag{2.3.6}$$

Thus, $\boldsymbol{w}$ can be computed through

$$\boldsymbol{w} = \frac{\mu}{2}(\mathbf{C} + \alpha\mathbf{I})^{-1}\mathbf{1}, \tag{2.3.7}$$

where $\alpha > 0$ is a regularisation term to avoid invertibility and singularity problems. The Lagrange multiplier $\mu = 2/\left[\mathbf{1}^\top(\mathbf{C} + \alpha\mathbf{I})^{-1}\mathbf{1}\right]$ is picked to properly normalise $\boldsymbol{w}$.

2.3.3 Computing the Embedding Coordinates

The embedding coordinates need to be found by minimising $\Phi(\mathbf{Z})$ in Eq. (2.3.2), which reduces to finding

$$\begin{aligned}
\Phi(\mathbf{Z}) &= \sum_{i=1}^{n} \|\boldsymbol{z}_i - \sum_{j=1}^{n} w_{ij}\boldsymbol{z}_j\|_2^2 \\
&= \sum_{i=1}^{n} \boldsymbol{z}_i^\top \boldsymbol{z}_i - \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\boldsymbol{z}_i^\top \boldsymbol{z}_j + \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\boldsymbol{z}_j^\top \boldsymbol{z}_i + \sum_{i=1}^{n}(\sum_{j=1}^{n} w_{ij}\boldsymbol{z}_j)^\top(\sum_{l=1}^{n} w_{il}\boldsymbol{z}_l) \\
&= \mathrm{Tr}\left(\mathbf{Z}^\top(\mathbf{I} - \mathbf{W} - \mathbf{W}^\top + \mathbf{W}^\top\mathbf{W})\mathbf{Z}\right). \qquad (2.3.8)
\end{aligned}$$

Define the $n \times n$ matrix $\mathbf{M} = \mathbf{I} - \mathbf{W} - \mathbf{W}^\top + \mathbf{W}^\top\mathbf{W}$. Notice that the above calculation also shows that $\mathbf{M}$ is positive semidefinite. And

$$\Phi(\mathbf{Z}) = \mathrm{Tr}(\mathbf{Z}^\top\mathbf{M}\mathbf{Z}). \tag{2.3.9}$$

This looks very familiar since we have already dealt similar trace optimisation problem of this in Section 2.1 and Section 2.2. Proceeding as we did with LE, we add a constraint on the embedded points to remove an arbitrary scale factor

$$\mathbf{Z}^\top\mathbf{Z} = \mathbf{I}, \tag{2.3.10}$$

and removes the degree of arbitrary translation by requiring the datapoints to be centred on the origin

$$\mathbf{Z}^\top\mathbf{1} = 0, \tag{2.3.11}$$

where $\mathbf{1}$ is defined similarly as a vector of all ones.

Using the same optimisation technique for solving the minimisation problem of Eq. (2.2.7), the embedded points are given by the eigenvector matrix $\mathbf{U}_k$ correspond to the $2nd$ and $(k+1)$th smallest eigenvalues of M. Formally, LLE finds the k dimensional embedding points as

$$\mathbf{Z} = \mathbf{U}_k. \tag{2.3.12}$$

2.4 Stochastic Neighbour Embedding

2.4.1 Model Construction

SNE [76] is a stochastic dimensionality reduction technique that tries to place the objects in a low dimensional space so as to optimally preserve the potential neighbours of the objects. It starts by encoding the high dimensional data structure into conditional or joint probabilities. Specifically, the conditional probability, $p_{j|i}$, that $\boldsymbol{x}_i$ would pick $\boldsymbol{x}_j$ as its neighbour, is defined as

$$p_{j|i} = \frac{\exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2/2\sigma_i^2\right)}{\sum_{l\neq i}\exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_l\|_2^2/2\sigma_i^2\right)}, \quad p_{i|i} = 0, \tag{2.4.1}$$

where a reasonable σ_i should enforce $p_{j|i}$ to be relatively high for nearby datapoints and low for widely separated datapoints. We will present the method for determining σ_i in the later section.

For the low dimensional mapping, it is possible to compute a similar conditional probability, $q_{j|i}$, as

$$q_{j|i} = \frac{\exp\left(-\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2/2\sigma^2\right)}{\sum_{l\neq i}\exp\left(-\|\boldsymbol{z}_i - \boldsymbol{z}_l\|_2^2/2\sigma^2\right)}. \tag{2.4.2}$$

For simplicity, SNE does not employ different values of σ for every datapoint in its low dimensional map and it is set to $\frac{1}{\sqrt{2}}$[1].

The aim of SNE is to find the low-dimensional embeddings $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ to match the corresponding distributions $q_{j|i}$ and $p_{j|i}$ as well as possible. This is carried out through minimising a sum of Kullback-Leibler divergence between the conditional distributions $P_i = \{p_{j|i}\}$ and $Q_i = \{q_{j|i}\}$ over all the datapoints:

$$C = \sum_{i=1}^{n} \mathrm{KL}(P_i\|Q_i) = \sum_{i=1}^{n}\sum_{j=1}^{n} p_{j|i}\log\frac{p_{j|i}}{q_{j|i}}. \tag{2.4.3}$$

[1] Setting σ to another value will only rescale the final map.

Notice that this cost function focuses on retaining the local structure of the data: it incurs a large cost for using distant mapped points to represent nearby datapoints (i.e., for using a small $q_{j|i}$ to model a large $p_{j|i}$); whereas the cost for using nearby mapped points to represent distant datapoints is almost infinitesimal.

Differential C with respect to the embedding point $\boldsymbol{z}_i$ has a surprisingly interesting form

$$\frac{\partial C}{\partial \boldsymbol{z}_i} = 2\sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(\boldsymbol{z}_i - \boldsymbol{z}_j), \tag{2.4.4}$$

where the mismatch $(p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})$ between the pairwise similarities of the datapoints and mapped points can be interpreted as creating forces that repel or attract $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$. Therefore, the minimisation problem of Eq. (2.4.3) can be solved via gradient-based optimisation methods.

2.4.2 Setting the Model Parameters

The remaining problem is to select the parameter σ_i to derive a proper probability distribution P_i. Since any probability distribution P_i needs to be informative to drive the learning process, it is natural to employ entropy as a measure of the randomness of this information. Let the Shannon entropy $H(P_i)$ to be measured in bits

$$H(P_i) = -\sum_{j=1}^{n} p_{j|i} \log_2(p_{j|i}). \tag{2.4.5}$$

SNE searches for the value of σ_i that makes the entropy of each distribution P_i equals to $\log_2(K)$, where K is a hyperparameter set by hand.

Next, σ_i need to be unique for any presetting $K > 0$. We prove this by noticing that the entropy of the distribution P_i increases monotonically with the parameter σ_i. Consider a distribution $P = \{p_j\}$ controlled by t as a simplified version to that distribution in Eq. (2.4.1)

$$p_j = \frac{\exp(-a_j/t)}{\sum_k \exp(-a_k/t)}, \tag{2.4.6}$$

where the positive scalars $\{a_1, a_2, \dots a_n\}$ are surrogate variables for the Euclidean distances in Eqs.(2.4.1) and t represents σ_i^2, is non-negative.

Differential the Shannon entropy $H(P)$ with respect to t, we have

$$\begin{aligned}
\frac{\partial H}{\partial t} &= -\frac{\partial\Big[\sum_j (p_j \log_2 p_j)\Big]}{\partial t} \\
&= -\sum_j \frac{\partial (p_j \log_2 p_j)}{\partial p_j}\frac{\partial p_j}{\partial t} \\
&= -\sum_j \frac{\partial p_j}{\partial t} log_2 p_j + \ln 2 \frac{\partial \sum_j p_j}{\partial t} \qquad (\frac{\partial \sum_j p_j}{t} = \frac{\partial 1}{t} = 0) \\
&= -t^{-2} \sum_j \big[a_j p_j - (\sum_k a_k p_k) p_j\big] log_2 p_j \\
&= -t^{-2} \sum_j \big[a_j p_j - (\sum_k a_k p_k) p_j\big] \big[log_2[\exp(-a_j/t)] - log_2(\sum_k \exp(-a_k/t))\big] \\
&= -t^{-2} \sum_j \big[a_j p_j - (\sum_k a_k p_k) p_j\big] log_2[\exp(-a_j/t)] \\
&= t^{-3} \ln 2 \big[\sum_j a_j^2 p_j - (\sum_j a_j p_j)^2\big] \\
&= t^{-3} \ln 2 \big[\sum_j a_j^2 p_j \sum p_j - (\sum_j a_j p_j)^2\big] \geq 0. \qquad \text{(Cauchy-Schwarz inequality)}
\end{aligned}$$

Thus, since we have $\frac{\partial H(P)}{\sigma^2} \geq 0$, the entropy $H(P)$ must increase monotonously with σ.

2.5 Canonical Correlation Analysis

2.5.1 Theoretical Foundations

Canonical Correlation Analysis (CCA) [80] [81] extracts the lower dimensional representations from two sets of correlated multidimensional variables, e.g., two views of the same semantic object, such that the strong relationships between these two sets of multidimensional variables are revealed. In particular, it projects the two sets of variables onto a common semantic space, and maximises the correlations their transformed data co-ordinates.

Formally, the input to CCA is a set of high dimensional datapoints $S = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$ of two random variables $\{\boldsymbol{x}, \boldsymbol{y}\}$, where each pair of measurements $\boldsymbol{x}_i$ (of dimension d_1) and $\boldsymbol{y}_i$ (of dimension d_2) are supposed to have a strong association.

We represent the $\boldsymbol{x}$ measurements $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ as rows in a $n \times d_1$ matrix $\mathbf{X}$ and the $\boldsymbol{y}$ measurements in $n \times d_2$ matrix $\mathbf{Y}$.

CCA defines two directions $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$ for projecting the measurements onto the same axis, with the new coordinates given as $\boldsymbol{z}_x = \mathbf{X}\boldsymbol{w}_x$ and $\boldsymbol{z}_y = \mathbf{Y}\boldsymbol{w}_y$, which can be viewed as sample measurements of the two transformed random variables $x' = \boldsymbol{x}^\top \boldsymbol{w}_x$ and $y' = \boldsymbol{y}^\top \boldsymbol{w}_y$. The correlation coefficient between the two new random variables are computed as

$$\begin{aligned}\rho &= \mathrm{corr}(x', y') \\ &= \frac{\mathrm{E}\left[(x' - \mathrm{E}(x')(y' - \mathrm{E}(y')\right]}{\sigma_{x'}\sigma_{y'}} \\ &= \frac{\mathrm{E}\left[(\boldsymbol{x}^\top \boldsymbol{w}_x - \bar{\boldsymbol{x}}^\top \boldsymbol{w}_x)(\boldsymbol{y}^\top \boldsymbol{w}_y - \bar{\boldsymbol{y}}^\top \boldsymbol{w}_y))\right]}{\sqrt{\mathrm{E}\left[(\boldsymbol{x}^\top \boldsymbol{w}_x - \bar{\boldsymbol{x}}^\top \boldsymbol{w}_x)^2\right]\mathrm{E}\left[(\boldsymbol{y}^\top \boldsymbol{w}_y - \bar{\boldsymbol{y}}^\top \boldsymbol{w}_y)^2\right]}},\end{aligned} \tag{2.5.1}$$

where E is the expected value operator, $\sigma_{x'}$ and $\sigma_{y'}$ are standard deviation of random variables x' and y'. $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{y}}$ are the expected value of random variables $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively.

CCA aims to find the directions $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$ that maximises the the sample correlation of the embedded measurements, which is written as

$$\begin{aligned}\bar{\rho}(\boldsymbol{w}_x, \boldsymbol{w}_y) &= \frac{\hat{\mathrm{E}}\left[\boldsymbol{w}_x^\top(\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{y} - \bar{\boldsymbol{y}})^\top \boldsymbol{w}_y\right]}{\sqrt{\hat{\mathrm{E}}\left[\boldsymbol{w}_x^\top(\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})^\top \boldsymbol{w}_x\right]\hat{\mathrm{E}}\left[\boldsymbol{w}_y^\top(\boldsymbol{y} - \bar{\boldsymbol{y}})(\boldsymbol{y} - \bar{\boldsymbol{y}})^\top \boldsymbol{w}_y\right]}} \\ &= \frac{\boldsymbol{w}_x^\top \hat{\mathrm{E}}\left[(\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{y} - \bar{\boldsymbol{y}})^\top\right]\boldsymbol{w}_y}{\sqrt{\boldsymbol{w}_x^\top \hat{\mathrm{E}}\left[(\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})^\top\right]\boldsymbol{w}_x \boldsymbol{w}_y^\top \hat{\mathrm{E}}\left[(\boldsymbol{y} - \bar{\boldsymbol{y}})(\boldsymbol{y} - \bar{\boldsymbol{y}})^\top\right]\boldsymbol{w}_y}},\end{aligned}$$

where $\hat{\mathrm{E}}$ is the empirical expectation operator.

Note that $\hat{\mathrm{E}}\left[(\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{y} - \bar{\boldsymbol{y}})^\top\right]$, $\hat{\mathrm{E}}\left[(\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{y} - \bar{\boldsymbol{y}})^\top\right]$ and $\hat{\mathrm{E}}\left[(\boldsymbol{y} - \bar{\boldsymbol{y}})(\boldsymbol{y} - \bar{\boldsymbol{y}})^\top\right]$ are sample covariance matrices, their analytic forms can be calculated similarly as in Eq. (2.1.3). Following the convention of [81], we denote these matrices as $\mathbf{C}_{\boldsymbol{xy}}$, $\mathbf{C}_{\boldsymbol{xx}}$ and $\mathbf{C}_{\boldsymbol{yy}}$, respectively.

Hence, the optimisation problem for CCA can be rewritten as

$$\underset{\boldsymbol{w}_x, \boldsymbol{w}_y}{\mathrm{argmax}}\, \bar{\rho}(\boldsymbol{w}_x, \boldsymbol{w}_y) = \frac{\boldsymbol{w}_x^\top \mathbf{C}_{\boldsymbol{xy}} \boldsymbol{w}_y}{\sqrt{\boldsymbol{w}_x^\top \mathbf{C}_{\boldsymbol{xx}} \boldsymbol{w}_x \boldsymbol{w}_y^\top \mathbf{C}_{\boldsymbol{yy}} \boldsymbol{w}_y}},$$

To make the optimisation problem well-posed, CCA imposes two constraints as

$$\boldsymbol{w}_x^\top \mathbf{C}_{\boldsymbol{xx}} \boldsymbol{w}_x = 1, \tag{2.5.2}$$

$$\boldsymbol{w}_y^\top \mathbf{C}_{\boldsymbol{yy}} \boldsymbol{w}_y = 1. \tag{2.5.3}$$

It can be noticed that $\mathbf{C}_{\boldsymbol{xx}}$ and $\mathbf{C}_{\boldsymbol{yy}}$ are real, symmetric and positive semidefinite matrix.

2.5.2 Multidimensional Extension

From the above definition of CCA, only one maximally correlated axis is recovered. The additional axes can be computed to be that which maximises the correlation amongst all possible directions non-redundant to those already considered. Therefore, CCA formulates the problem of finding the k maximally correlated axes as follows,

$$\begin{array}{ll} \mathrm{argmax}_{\{\mathbf{W}_x,\mathbf{W}_y\}} & \mathrm{Tr}\left(\mathbf{W}_x^\top \mathbf{C}_{\boldsymbol{xy}} \mathbf{W}_y\right) \\ s.t. & \mathbf{W}_x^\top \mathbf{C}_{\boldsymbol{xx}} \mathbf{W}_x = \mathbf{I}_k, \quad \mathbf{W}_y^\top \mathbf{C}_{\boldsymbol{yy}} \mathbf{W}_y = \mathbf{I}_k, \\ \text{and} & \mathrm{Off}\left(\mathbf{W}_x^\top \mathbf{C}_{\boldsymbol{xy}} \mathbf{W}_y\right) = 0. \end{array} \tag{2.5.4}$$

Where $\mathrm{Off}\,(\cdot) = 0$ means all off-diagonal elements of the associated matrix are zeros.

Again, this is a trace optimisation problem that can be dealt in a similar manner as those in Section 2.1 and Section 2.2. Let matrix $\mathbf{C}_{\boldsymbol{xx}}^{-\frac{1}{2}} \mathbf{C}_{\boldsymbol{xy}} \mathbf{C}_{\boldsymbol{yy}}^{-\frac{1}{2}}$ to be decomposed as $\mathbf{U\Sigma V}^\top$, where columns of $\mathbf{U}$ and $\mathbf{V}$ contain the left and right singular vectors, the diagonal entries $\{\sigma_i\}_{i=1}^{\min(d_1,d_2)}$ of $\mathbf{\Sigma}$ are the non-negative real singular values. Substitute $\mathbf{P} = \mathbf{U}^\top \mathbf{C}_{\boldsymbol{xx}}^{\frac{1}{2}} \mathbf{W}_x$ and $\mathbf{Q} = \mathbf{V}^\top \mathbf{C}_{\boldsymbol{yy}}^{\frac{1}{2}} \mathbf{W}_y$ into the above optimisation problem, it follows that

$$\begin{array}{ll} \mathrm{argmin}_{\{\mathbf{P},\mathbf{Q}\}} & \mathrm{Tr}\left(\mathbf{P}^\top \mathbf{\Sigma Q}\right) \\ \text{s.t.} & \mathbf{P}^\top \mathbf{P} = \mathbf{I}_k, \quad \mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_k, \\ \text{and} & \mathrm{Off}\left(\mathbf{P}^\top \mathbf{\Sigma Q}\right) = 0. \end{array} \tag{2.5.5}$$

Since we have

$$\begin{aligned} \mathrm{Tr}\left(\mathbf{P}^\top \mathbf{\Sigma Q}\right) &= \mathrm{Tr}\left(\mathbf{P}^\top \mathbf{\Sigma Q}\right) \\ &= \sum_{i=1}^{\min(d_1,d_2)} \sigma_i \sum_{j=1}^{k} p_{ij} q_{ij} \end{aligned} \tag{2.5.6}$$

We show that

$$\begin{aligned}\sum_{j=1}^{k} p_{ij}q_{ij} &\leq \sqrt{(\sum_{j=1}^{k} p_{ij}^2)(\sum_{l=1}^{k} q_{il}^2)} && \text{(Cauchy-Schwarz inequality)}\\ &= \sqrt{\|\mathbf{P}^\top \boldsymbol{e}_i\|_2^2 \|\mathbf{Q}^\top \boldsymbol{e}_i\|_2^2}\\ &\leq \sqrt{\|\boldsymbol{e}_i\|_2^2\|\boldsymbol{e}_i\|_2^2} = 1 \end{aligned} \tag{2.5.7}$$

and

$$\begin{aligned}\sum_{i}^{\min(d_1,d_2)} \sum_{j=1}^{k} p_{ij}q_{ij} &\leq \sum_{i=1}^{\min(d_1,d_2)} \sqrt{(\sum_{j=1}^{k} p_{ij}^2)(\sum_{l=1}^{k} q_{il}^2)} && \text{(Cauchy-Schwarz inequality)}\\ &\leq \sqrt{(\sum_{i=1}^{\min(d_1,d_2)} \sum_{j=1}^{k} p_{ij}^2)(\sum_{i=1}^{\min(d_1,d_2)} \sum_{l=1}^{k} q_{il}^2)}\\ & && \text{(Cauchy-Schwarz inequality)}\\ &\leq \sqrt{(\sum_{i=1}^{d_1} \sum_{j=1}^{k} p_{ij}^2)(\sum_{i=1}^{d_2} \sum_{l=1}^{k} q_{il}^2)}\\ &= \sqrt{\operatorname{Tr}\left(\mathbf{P}^\top\mathbf{P}\right)\operatorname{Tr}\left(\mathbf{Q}^\top\mathbf{Q}\right)} = k \end{aligned} \tag{2.5.8}$$

Thus, we conclude that

$$\operatorname{Tr}\left(\mathbf{W}_x^\top \mathbf{C}_{\boldsymbol{xy}} \mathbf{W}_y\right) = \operatorname{Tr}\left(\mathbf{P}^\top \boldsymbol{\Sigma} \mathbf{Q}\right) \leq \sum_{i=1}^{k} \sigma_i, \tag{2.5.9}$$

and CCA chooses the projection matrices $\mathbf{W}_x$ and $\mathbf{W}_y$ as

$$\mathbf{W}_x = \mathbf{C}_{\boldsymbol{xx}}^{-\frac{1}{2}} \mathbf{U}_k, \tag{2.5.10}$$

$$\mathbf{W}_y = \mathbf{C}_{\boldsymbol{yy}}^{-\frac{1}{2}} \mathbf{V}_k, \tag{2.5.11}$$

where the columns of $\mathbf{U}_k$ and $\mathbf{V}_k$ are the left and right singular vectors corresponding to the k largest singular values of the normalised covariance matrix $\mathbf{C}_{\boldsymbol{xx}}^{-\frac{1}{2}} \mathbf{C}_{\boldsymbol{xy}} \mathbf{C}_{\boldsymbol{yy}}^{-\frac{1}{2}}$. Setting $k = 1$, the directions $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$ in the previous section are derived.

2.6 Restricted Boltzmann Machine

2.6.1 Theoretical Foundations

Restricted Boltzmann Machine (RBM) is an undirected graphical model [82] associated with a bipartite conditional independence structure as shown in Figure 2.1. The nodes of this bipartite are divided into two disjoint sets – so called visible and hidden units – the visible units correspond to the observable components (e.g., attributes, features) of the data, whereas the hidden units are never observed and are expected to extract relevant hidden features from the data. Because of its specific dependence structure (only have visible-hidden connections), the inferences on RBMs are much easier and faster than general undirected graphical models.

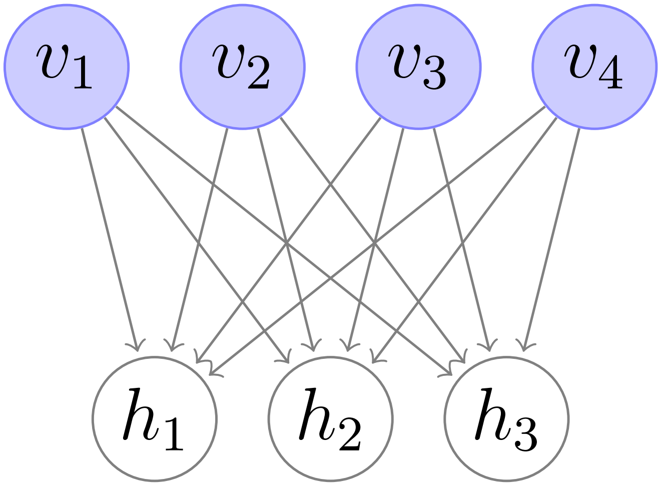


Figure 2.1: A graphical depiction of an RBM with 4 visible and 3 hidden units.

Let d visible random variables to be $\{v_1, \ldots, v_d\}$ and k hidden variables denoted as $\{h_1, \ldots, h_k\}$. And suppose the observed training data are given in set $\mathcal{D} = \{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(n)}\}$. Here, We focus on the commonly studied cases of using binary units where v_j and $h_i \in \{0, 1\}$. The joint probability distribution under this model is expressed as exponentials

$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} \exp\{-\mathrm{E}(\boldsymbol{v}, \boldsymbol{h})\}, \tag{2.6.1}$$

with the linear energy function $\mathrm{E}(\boldsymbol{v}, \boldsymbol{h})$ to be defined as:

$$\mathrm{E}(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{b}^\top \boldsymbol{v} - \boldsymbol{c}^\top \boldsymbol{h} - \boldsymbol{h}^\top \mathbf{W} \boldsymbol{v}, \tag{2.6.2}$$

where $\mathbf{W}$ represents the weights connecting hidden and visible units and $\boldsymbol{b}$, $\boldsymbol{c}$ are the offsets for the visible and hidden nodes respectively. Z is the partition function that is used to normalise the probability distribution $p(\boldsymbol{v}, \boldsymbol{h})$.

In RBMs, visible and hidden units are conditional independent given one-another since there are no visible-visible and hidden-hidden connections. Using this property, we can write the decomposition of the conditional probability as:

$$p(\boldsymbol{h}|\boldsymbol{v}) = \prod_{i}^{k} p(h_i|\boldsymbol{v}), \tag{2.6.3}$$

$$p(\boldsymbol{v}|\boldsymbol{h}) = \prod_{j}^{d} p(v_j|\boldsymbol{h}). \tag{2.6.4}$$

Also, it is easy to verify that

$$P(h_i = 1|\boldsymbol{v}) = \sigma(c_i + \mathbf{W}_{i\cdot}\boldsymbol{v}), \tag{2.6.5}$$

$$P(v_j = 1|\boldsymbol{h}) = \sigma(b_j + \boldsymbol{h}^\top \mathbf{W}_{\cdot j}), \tag{2.6.6}$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid activation function, $\mathbf{W}_{i\cdot}$ and $\mathbf{W}_{\cdot j}$ are the ith row and jth column of the weight matrix $\mathbf{W}$, respectively.

From Eqs. (2.6.3-2.6.6), we see that computing $p(\boldsymbol{h}|\boldsymbol{v})$ and $p(\boldsymbol{v}|\boldsymbol{h})$ given the free parameters $\mathbf{W}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ are surprisingly simple. Consequently, the samples in the joint probability distribution $p(\boldsymbol{v}, \boldsymbol{h})$ of an RMB is easy to obtain: *block Gibbs sampling* can be performed by sampling the new state $\boldsymbol{h}$ for the hidden units based on $p(\boldsymbol{h}|\boldsymbol{v})$ and sampling new state $\boldsymbol{v}$ for the visible units based on $p(\boldsymbol{v}|\boldsymbol{h})$. This is beneficial for approximating the negative log-likelihood gradient as we will see in the following.

2.6.2 Approximations of Negative Log-likelihood Gradient

For training an RBM model, we are interested in its marginal distribution of observed samples

$$\begin{aligned} p(\boldsymbol{v}) &= \frac{1}{Z}\sum_{\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z}\sum_{\boldsymbol{h}} e^{-\mathrm{E}(\boldsymbol{v},\boldsymbol{h})} \\ &= \frac{1}{Z} e^{-\mathcal{F}(\boldsymbol{v})}, \end{aligned} \tag{2.6.7}$$

where the free energy $\mathcal{F}(\boldsymbol{v})$ is defined as

$$\mathcal{F}(\boldsymbol{v}) = -\log \sum_{\boldsymbol{h}} e^{-\mathrm{E}(\boldsymbol{v},\boldsymbol{h})}, \tag{2.6.8}$$

and the partition function Z can be expressed as $Z = \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-\mathrm{E}(\boldsymbol{v},\boldsymbol{h})} = \sum_{\boldsymbol{v}} e^{-\mathcal{F}(\boldsymbol{v})}$.

As for the RBM, it can be simplifies to

$$\mathcal{F}(\boldsymbol{v}) = -\boldsymbol{b}^\top \boldsymbol{v} - \sum_i \log(1 + e^{(c_i + \mathbf{W}_i.\boldsymbol{v})}). \tag{2.6.9}$$

An RBM model is learnt by minimising the empirical negative log-likelihood of the training data, which is expressed as

$$\begin{aligned}\ell(\mathbf{W}, \boldsymbol{b}, \boldsymbol{c}, \mathcal{D}) &= -\frac{1}{n} \sum_{\boldsymbol{v}^{(i)} \in \mathcal{D}} \log p(\boldsymbol{v}^{(i)}) \\ &= -\frac{1}{n} \sum_{\boldsymbol{v}^{(i)} \in \mathcal{D}} \left[\mathcal{F}(\boldsymbol{v}^{(i)}) + \log(\sum_{\boldsymbol{v}} e^{-\mathcal{F}(\boldsymbol{v})}) \right] \end{aligned} \tag{2.6.10}$$

Then, the data negative log-likelihood gradient with respect to any of the free parameters $\mathbf{W}$, $\boldsymbol{b}$ or $\boldsymbol{c}$ has the form

$$-\frac{\partial \log p(\boldsymbol{v}^{(i)})}{\partial \theta} = \frac{\partial \mathcal{F}(\boldsymbol{v}^{(i)})}{\partial \theta} - \sum_{\hat{\boldsymbol{v}}} p(\hat{\boldsymbol{v}}) \frac{\partial \mathcal{F}(\hat{\boldsymbol{v}})}{\partial \theta}, \tag{2.6.11}$$

where θ refers to either $\mathbf{W}$, $\boldsymbol{b}$ or $\boldsymbol{c}$.

It is usually difficult to determine this gradient analytically, as it has to compute the term $E_p[\frac{\partial \mathcal{F}(\boldsymbol{v})}{\partial \theta}]$, which is an expectation over all possible configurations of the multivariate random variable $\boldsymbol{v}$ under the marginal distribution $p(\boldsymbol{v})$ formed by the model. But we can approximate it as the expectation over a fixed number of model samples, which are denoted as $\mathcal{N}$. And the gradient can then be approximated as:

$$-\frac{\partial \log p(\boldsymbol{v}^{(i)})}{\partial \theta} = \frac{\partial \mathcal{F}(\boldsymbol{v}^{(i)})}{\partial \theta} - \frac{1}{|\mathcal{N}|} \sum_{\hat{\boldsymbol{v}} \in \mathcal{N}} \frac{\partial \mathcal{F}(\hat{\boldsymbol{v}})}{\partial \theta} \tag{2.6.12}$$

where the elements $\hat{\boldsymbol{v}}$ can be sampled according to the block Gibbs sampling procedure mentioned in the above section. With the gradient practical to compute, RBM models are learnt by gradient based optimisation methods, such as stochastic gradient descent.

2.7 Conclusion

In this chapter, we have scrutinised the a number of machine learning algorithms in the field of dimensionality reduction. These generic algorithms lay the foundations for developing new embedding approaches and their key ideas are repeatedly used and developed in the whole field of machine learning. We summarise their main ideas here: **a)** PCA uses an orthogonal transformation to convert a set of observation variables into a set of variables called principle components. In this procedure, PCA aims to maintain as much of the *variability* in the data as possible. **b)** LE, LLE and SNE are all *locality* preserving methods. They differ from one another in their ways of encoding the locality information, i.e., making the use of neighbourhood graphs, local linear structures or probability distributions. **c)** CCA explores relationships between two sets of multivariate variables, and maximises the *correlation* between pairs of transformed versions of these variables. **d)** An RBM is a instance of *generative stochastic models*. It defines a parametric probability distribution over its set of inputs and the learning is carried out by minimising the mismatch between the input and the probability distribution.

We have come across the trace optimisation problem on numerous occasions (such as in Eqs. (2.1.9, 2.2.7, 2.3.9, 2.5.4)). This problem is important to almost all the matrix factorisation methods, and due to its convexity the analytic form of solution can be given directly. In this chapter, we have closely studied various forms of this trace optimisation problem and clarifying its mathematic formulation and solution will certainly be helpful to new algorithm designs (e.g., properly imposing constraints).

From next chapters we will present our consecutive works in order. It should be noted that our proposed methods in Chapters 3 and 5 rely heavily on the formulations of LE and SNE, respectively. It is encouraged to compare them with the LE and SNE demonstrated in this chapter.

Chapter 3

Heterogeneous Object Co-embeddings from Relational Measurements

3.1 Introduction

In Chapter 2, we have reviewed some of the conventional embedding methods that build the low dimensional mappings from the high dimensional feature representations. Such methods are valuable tools for data preprocessing, data analysis and information visualisation. However, these techniques only embed homogeneous (i.e., of a single type) data objects into a low-dimensional space given their higher dimensional feature representations. While in many real-world applications, data may come from heterogeneous sources, such as genes and symptoms, documents and words or images, review articles from different domains. It could therefore be useful to simultaneously handle heterogeneous types of data, 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 [83, 84], semantic analysis [64, 85] and information retrieval [86, 87]. Heterogeneous data analysis has also been performed by more generic methods. For instance, Correspondence Analysis (CA) [88] represents the rows and columns of a data matrix as points in a space of low-dimensionality. Latent Semantic Indexing [89] is a popular information retrieval embedding method, frequently used to embed documents and words in a common space [90]. CCA [91] attempts to maximise the correlation between two sets of measurements. Similarly, variations of nonmetric Mutidimensional Scaling [64] 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 [92] 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 (correspond to the bipartite relations in Section 1.1.1) between two sets of objects of possibly differing cardinality. This is also known as joint embedding or co-embedding [28, 86, 93]. 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 [93], Co-occurrence Data Embedding (CODE) [63], Bayesian Co-occurrence Data Embedding [86], as well as a dynamic embedding model that processes a sequence of co-occurrence data changing over time [94]. 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 maximises 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 (ACAS) [28] based on matrix factorisation, which generalises ideas from embedding algorithms such as [88,89,95,96], 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 specialised at learning embeddings from a binary relation matrix between two groups of objects. For instance, Maximum-Margin Matrix Factorization [97] 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 [98].

In this work, 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 the availability of the relational similarity information between representatives from each group. The co-embedding generation relies on an efficient joint model optimisation 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 organise the rest of this chapter as follows. Section 3.2 briefly reviews some related heterogeneous embedding algorithms. In Section 3.3, we introduce the proposed algorithm, its model, optimisation scheme, as well as its parameter identification mechanism. The experimental results and comparative analyses are reported in Section 3.4, while Section 3.5 concludes the work.

3.2 Related Methods

In the following subsections, we summarise existing algorithms to generate the heterogeneous co-embeddings from a relational similarity matrix. Before that, we introduce the mathematical definition and the objective of the problem.

We are given an $m \times n$ input matrix $\mathbf{R} = [r_{ij}]$, which is assumed to be nonnegative 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

$\boldsymbol{z}_j^{(y)} = [z_{1j}^{(y)}, z_{2j}^{(y)}, \dots, z_{kj}^{(y)}]^\top$ for objects y_j.

3.2.1 Co-occurrence Data Embedding

CODE [63] is based on a statistical model which interprets ijth element of the input matrix $\mathbf{R}$ as empirical joint probabilities. That is, the set of object pairs $\{x_i, y_j\}$ are assumed to be drawn from a discrete joint distribution $p(X, Y)$ of random variable $X \in \{x_1, \dots, x_m\}$ and $Y \in \{y_1, \dots, y_n\}$. And the empirical joint distribution $\bar{p}(X, Y)$ is obtained by dividing the observed relation matrix by its grand total $r_{ij}/(\sum_{i=1}^m \sum_{j=1}^n r_{ij})$. Then CODE relates the unknown joint distribution $p(X, Y)$ to the underlying low dimensional points $\{\boldsymbol{z}_i^{(x)}\}_{i=1}^m$ and $\{\boldsymbol{z}_j^{(y)}\}_j^n$ in such a way that a pair of objects that are embedded as two nearby points in the map have a higher statistical interaction than a pair that is embedded as two distant points. Formally, the squared Euclidean distance between the embedding points $\boldsymbol{z}_i^{(x)}$ and $\boldsymbol{z}_j^{(y)}$ is denoted by

$$d_{i,j}^2 = \|\boldsymbol{z}_i^{(x)} - \boldsymbol{z}_j^{(y)}\|_2^2 = \sum_{q=1}^k (z_{qi}^{(x)} - z_{qj}^{(y)})^2. \tag{3.2.1}$$

To model the statistical relationships between objects, CODE transforms the squared Euclidean distances into probabilities via the exponential form $\mathrm{e}^{-d_{i,j}^2}$. It either construct the joint distribution symmetrically via

$$p(x_i, y_j) \equiv \frac{1}{h(\boldsymbol{z}_i^{(x)}, \boldsymbol{z}_j^{(y)})} \bar{p}(x_i)\bar{p}(y_j)\mathrm{e}^{-d_{i,j}^2}, \tag{3.2.2}$$

or asymmetrically as

$$p(x_i, y_j) = \bar{p}(x_i)p(y_j|x_i) \equiv \frac{1}{h(\boldsymbol{z}_i^{(x)})} \bar{p}(x_i)\bar{p}(y_j)\mathrm{e}^{-d_{i,j}^2}, \tag{3.2.3}$$

where $h(\boldsymbol{z}_i^{(x)}, \boldsymbol{z}_j^{(y)}) = \sum_{i=1}^m \sum_{j=1}^n \bar{p}(x_i)\bar{p}(y_j)\mathrm{e}^{-d_{i,j}^2}$ and $h(\boldsymbol{z}_i^{(x)}) = \sum_{j=1}^n \bar{p}(y_j)\mathrm{e}^{-d_{i,j}^2}$ are the normalisation terms, $\bar{p}(x_i) = \sum_{j=1}^n \bar{p}(x_i, y_j)$ and $\bar{p}(y_j) = \sum_{i=1}^m \bar{p}(x_i, y_j)$ are the empirical margins. In fact, these models are constructed by requiring the ratio $\bar{r}_p = \frac{p(x_i, y_j)}{\bar{p}(x_i)\bar{p}(y_j)}$ to be proportional to $\mathrm{e}^{-d_{i,j}^2}$. They have been shown to outperform other models which do not address both margins $\bar{p}(X)$ and $\bar{p}(Y)$.

The degree of the correspondence between input distributions $\bar{p}(X, Y)$ and parametric distributions $p(X, Y)$ is then measured using the negative log-likelihood function

$$\ell(\mathbf{Z}_x, \mathbf{Z}_y) = -\sum_{i=1}^{m}\sum_{j=1}^{n} \bar{p}(x_i, y_j) \log p(x_i, y_j) \tag{3.2.4}$$

The negative log-likelihood in this case is a non-concave function, one can simply find the local optimum by using a standard conjugate gradient descend algorithm. However, since the joint probabilities $p(X, Y)$ depends only on the squared Euclidean distances between points in $\mathcal{X}$ and in $\mathcal{Y}$, i.e. , considering the asymmetrical model in Eq. (3.2.3), the negative log-likelihood can be rewritten as

$$\begin{aligned}\ell(\mathbf{Z}_x, \mathbf{Z}_y) &= -\sum_{i=1}^{m}\sum_{j=1}^{n} \bar{p}(x_i, y_j)(-d_{ij}^2 - \log h(\boldsymbol{z}_i^{(x)}) + \log \bar{p}(x_i) + \log \bar{p}(y_j)) \\ &= \sum_{i=1}^{m}\sum_{j=1}^{n} \bar{p}(x_i, y_j) d_{ij}^2 + \sum_{i=1}^{m} \log h(\boldsymbol{z}_i^{(x)}) + \text{const} \\ &= \sum_{i=1}^{m}\sum_{j=1}^{n} \bar{p}(x_i, y_j) d_{ij}^2 + \sum_{i=1}^{m} \log \sum_{j=1}^{n} \bar{p}(y_j) \mathrm{e}^{-d_{i,j}^2} + \text{const}\end{aligned} \tag{3.2.5}$$

where $\text{const} = -\sum_{i=1}^{m} \log \bar{p}(x_i) + \sum_{j=1}^{n} \log \bar{p}(y_j)$ is a constant term that does not depend on model parameters $\mathbf{Z}_x$ and $\mathbf{Z}_y$.

This allows CODE to reformulate the problem as constrained convex optimisation over the cone of Positive Semi-Definite (PSD) matrices, i.e., the symmetric PSD Gram matrix $\mathbf{G} \equiv \begin{bmatrix} \mathbf{Z}_x \\ \mathbf{Z}_y \end{bmatrix} \begin{bmatrix} \mathbf{Z}_x^\top & \mathbf{Z}_y^\top \end{bmatrix}$. By the fact that the embedding distances are linear functions of the elements of $\mathbf{G}$ as $d_{ij}^2 = g_{i,i} + g_{m+j,m+j} - 2g_{i,j+m}$, we can write the negative log-likelihood $\ell(\mathbf{Z}_x, \mathbf{Z}_y)$ as function of $\mathbf{G}$ only. Such as

$$\begin{aligned} f(\mathbf{G}) = &\sum_{i=1}^{m}\sum_{j=1}^{n} \bar{p}(x_i, y_j)(g_{i,i} + g_{m+j,m+j} - 2g_{i,j+m}) \\ &+ \sum_{i=1}^{m} \log \sum_{j=1}^{n} \bar{p}(y_j) \exp\left(-g_{i,i} - g_{m+j,m+j} + 2g_{i,j+m}\right), \end{aligned} \tag{3.2.6}$$

where the constant additive terms are ignored.

Thus, CODE reduces the problem to finding the positive semidefinite matrix $\mathbf{G}$

$$\begin{array}{ll} \text{argmin}_{\mathbf{G}} & f(\mathbf{G}) + \lambda \text{Tr}(\mathbf{G}) \\ \text{s.t.} & \mathbf{G} \succeq 0. \end{array} \tag{3.2.7}$$

where the regularisation term $\lambda\mathrm{Tr}(\mathbf{G})$ with some positive constant λ is added so as to learn a sparse set of eigenvalues of $\mathbf{G}$. As a result, the low embedding points can be obtained by factorisation of the PSD matrix $\mathbf{G} = \mathbf{Z}^\top\mathbf{Z}$.

Note that in this formulation, the minimising function $f(\mathbf{G}) + \lambda\mathrm{Tr}(\mathbf{G})$ is convex. Since the first term in $f(\mathbf{G})$ and the regularisation term $\lambda\mathrm{Tr}(\mathbf{G})$ are linear functions of elements in $\mathbf{G}$; the second term in $\mathbf{G}$ is a sum of convex terms $\log\sum\exp$ of an affine expression in the PSD matrix $\mathbf{G}$ [99]. This constrained convex optimisation problem is solved via *projected gradient* algorithm with the use of an Armijo rule [100] to select proper step size in CODE method.

3.2.2 Bipartite Graph Partitioning

Bipartite Graph Partitioning (BGP) [95] consider the problem of simultaneous or co-clustering of a set of heterogeneous objects, e.g., documents and words when processing a corpus. It converts the dual clustering problem — clustering documents based upon their word distributions and clustering word by their co-occurrence in documents — to finding minimum cut vertex partitions in a bipartite graph between documents and words. Given k disjoint row clusters $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k$, the association that a column object y_i with the column cluster $\mathcal{Y}_m$ is measured by the sum of edge-weights to all row objects in the cluster $\mathcal{X}_m$. And it belongs to cluster $\mathcal{Y}_m$ if and only if its association to cluster $\mathcal{X}_m$ is greater than any other row clusters

$$\mathcal{Y}_m = \left\{ y_i : \sum_{j\in\mathcal{X}_m} r_{ij} \geq \sum_{j\in\mathcal{X}_l} r_{ij},\ \forall\, l = 1, \ldots, k \right\}. \tag{3.2.8}$$

Symmetrically, the cluster assignment of row object x_i is determined by its association to column clusters $\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_k$

$$\mathcal{X}_m = \left\{ x_j : \sum_{i\in\mathcal{Y}_m} r_{ij} \geq \sum_{i\in\mathcal{Y}_l} r_{ij},\ \forall\, l = 1, \ldots, k \right\}. \tag{3.2.9}$$

From this viewpoint, the "best" row and column clustering would correspond to a graph partitioning such that the crossing edges between partitions have minimum weight. To build such a bipartite graph, BGP sets the $(m + n) \times (m + n)$ adjacency matrix of the graph as

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{0} \end{bmatrix}, \tag{3.2.10}$$

where the vertices are ordered such that the first m vertices index the row objects in $\mathcal{X}$ group while the last n index the column objects in $\mathcal{Y}$ group.

Thus, for any k partitioning $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_k$ of the bipartite graph vertices in $\mathcal{V} = \mathcal{X} \cup \mathcal{Y}$, the minimum cut problem is defined as

$$\underset{\mathcal{V}_1,\ldots,\mathcal{V}_k}{\operatorname{argmin}} \operatorname{cut}(\mathcal{V}_1, \ldots, \mathcal{V}_k) = \underset{\mathcal{V}_1,\ldots,\mathcal{V}_k}{\operatorname{argmin}} \sum_{i<j} \sum_{p\in\mathcal{V}_i, q\in\mathcal{V}_j} a_{pq}, \tag{3.2.11}$$

where a_{pq} is the edge weight between vertices p, q in the bipartite graph.

Finding a globally optimal solution to such a graph partition problem is NP-complete, therefore BGP seeks an effective heuristic method that is introduced as *spectral graph partitioning* [101]. We introduce the spectral graph partition heuristic and show their connections to a real approximation solution using singular value decomposition as follows.

For any bipartitioning of $\mathcal{V}$ into $\mathcal{V}_1$ and $\mathcal{V}_2$, let $\boldsymbol{g}$ be the associated *partition vector* given by

$$g_i = \begin{cases} +1, & i \in \mathcal{V}_1, \\ -1, & i \in \mathcal{V}_2. \end{cases} \tag{3.2.12}$$

then, the edge-weight between vertices $\mathcal{V}_1$ and $\mathcal{V}_2$ can be expressed as

$$\begin{aligned} \operatorname{cut}(\mathcal{V}_1, \mathcal{V}_2) &= \sum_{i\in\mathcal{V}_1, j\in\mathcal{V}_2} a_{ij} \\ &= \frac{1}{4} \sum_{i,j} a_{ij}(g_i - g_j)^2 \\ &= \frac{1}{4} \boldsymbol{g}^\top \mathbf{L} \boldsymbol{g} \end{aligned} \tag{3.2.13}$$

where $\mathbf{L}$ is the *Laplacian matrix* with the ijth elements to be

$$l_{ij} = \begin{cases} \sum_k a_{ik}, & i = j \\ -a_{ij}, & i \neq j \text{ and there is an edge } \{i, j\} \\ 0 & \text{otherwise.} \end{cases} \tag{3.2.14}$$

BGP further extends the graph cut objective, by associating each vertex i a positive weight, denoted by $\mathrm{weight}(i)$, to "balance" the clustering size. Let $\mathbf{W} = [w_{ij}]$ be the diagonal matrix of the vertex weights, and the weight for a subset of vertices $\mathcal{V}_l$ is given by $\mathrm{weight}(\mathcal{V}_l) = \sum_{i \in \mathcal{V}_l} w_{ii}$, then the generalised graph cut objective is defined as

$$\mathcal{Q}(\mathcal{V}_1, \mathcal{V}_2) = \frac{\mathrm{cut}(\mathcal{V}_1, \mathcal{V}_2)}{\mathrm{weight}(\mathcal{V}_1)} + \frac{\mathrm{cut}(\mathcal{V}_1, \mathcal{V}_2)}{\mathrm{weight}(\mathcal{V}_2)}. \tag{3.2.15}$$

Given two different partitionings with the same cut value, minimising the above objective function will favour the "balanced" one, which corresponds to near-equal weight values on subsets $\mathcal{V}_1$ and $\mathcal{V}_2$.

In the BGP model, the weight of each vertex is chosen to equal the sum of the edge weights on it, i.e., $\mathrm{weight}(i) = \sum_k a_{ik}$. As a result, the vertex weight matrix $\mathbf{W}$ equals the degree matrix of the graph, and $w_{ii} = \mathrm{weight}(i) = \sum_k a_{ik}$.

Again, any partition pattern can be represented by the *generalised partition vector* $\boldsymbol{h}$ with elements

$$h_i = \begin{cases} +\sqrt{\frac{\eta_2}{\eta_1}}, & i \in \mathcal{V}_1, \\ -\sqrt{\frac{\eta_1}{\eta_2}}, & i \in \mathcal{V}_2, \end{cases} \tag{3.2.16}$$

where $\eta_1 = \mathrm{weight}(\mathcal{V}_1)$ and $\eta_2 = \mathrm{weight}(\mathcal{V}_2)$.

Let $\boldsymbol{e}$ be the vector with all components 1. It can be seen that $\boldsymbol{h}$ satisfies $\boldsymbol{h}^\top \mathbf{W} \boldsymbol{e} = \sqrt{\frac{\eta_2}{\eta_1}} \sum_{i \in \mathcal{V}_1} \mathrm{weight}(i) - \sqrt{\frac{\eta_1}{\eta_2}} \sum_{i \in \mathcal{V}_2} \mathrm{weight}(i) = 0$ and $\boldsymbol{h}^\top \mathbf{W} \boldsymbol{h} = \sum_{i=1}^{m+n} w_{ii} h_i^2 = \eta_1 + \eta_2 = \mathrm{weight}(\mathcal{V})$.

Thus, Eq. (3.2.15) can be written as

$$\begin{aligned} \mathcal{Q}(\mathcal{V}_1, \mathcal{V}_2) &= (\frac{1}{\eta_1} + \frac{1}{\eta_2}) \cdot \mathrm{cut}(\mathcal{V}_1, \mathcal{V}_2) \\ &= (\frac{1}{\eta_1} + \frac{1}{\eta_2}) \cdot \sum_{i \in \mathcal{V}_1, j \in \mathcal{V}_2} a_{ij} \\ &= (\frac{1}{\eta_1} + \frac{1}{\eta_2}) \cdot \frac{\eta_1 \eta_2}{(\eta_1 + \eta_2)^2} \cdot \sum_{i,j} a_{ij} (h_i - h_j)^2 \\ &= \frac{1}{\eta_1 + \eta_2} \cdot \sum_{i,j} a_{ij} (h_i - h_j)^2 \\ &= \frac{\boldsymbol{h}^\top \mathbf{L} \boldsymbol{h}}{\boldsymbol{h}^\top \mathbf{W} \boldsymbol{h}}. \end{aligned} \tag{3.2.17}$$

Computing the optimal generalised partition vector $\boldsymbol{h}$ is still NP-complete for the above equation, instead, BGP finds a real relaxation solution to this two valued vector.

Finally, the graph cut problem is recast as calculating the real valued vector $\boldsymbol{z}$ such that

$$\min_{\boldsymbol{z}\neq\mathbf{0}} \frac{\boldsymbol{z}^\top\mathbf{L}\boldsymbol{z}}{\boldsymbol{z}^\top\mathbf{W}\boldsymbol{z}}, \text{ subject to } \boldsymbol{z}^\top\mathbf{W}\boldsymbol{e} = 0. \tag{3.2.18}$$

It is solved when $\boldsymbol{z}$ is the eigenvector corresponding to the 2nd smallest eigenvalue λ_2 of the generalised eigenvalue problem,

$$\mathbf{L}\boldsymbol{z} = \lambda\mathbf{W}\boldsymbol{z}. \tag{3.2.19}$$

For multidimensional extension, the eigenvectors corresponding to the 2nd to $(k+1)$th largest eigenvalues are taken as the co-embeddings.

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 equivalently given as

$$\mathbf{Z}_x = \mathbf{D}_x^{-\frac{1}{2}}\mathbf{U}_k, \tag{3.2.20}$$

$$\mathbf{Z}_y = \mathbf{D}_y^{-\frac{1}{2}}\mathbf{V}_k, \tag{3.2.21}$$

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 $\sigma_2 \geq \sigma_3 \geq \ldots \geq \sigma_k$.

The co-embeddings given by Eqs. (3.2.20, 3.2.21) is the same solution to Eq. (3.2.19), which is justified by the following

$$\begin{aligned}
\begin{bmatrix} \mathbf{D}_x & -\mathbf{R} \\ -\mathbf{R}^\top & \mathbf{D}_y \end{bmatrix} \begin{bmatrix} \mathbf{Z}_x^{(q)} \\ \mathbf{Z}_y^{(q)} \end{bmatrix} &= \begin{bmatrix} \mathbf{D}_x\mathbf{Z}_x^{(q)} - \mathbf{R}\mathbf{Z}_y^{(q)} \\ -\mathbf{R}^\top\mathbf{Z}_x^{(q)} + \mathbf{D}_y\mathbf{Z}_y^{(q)} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{D}_x\mathbf{Z}_x^{(q)} - \mu_{q+1}\mathbf{D}_x\mathbf{Z}_x^{q} \\ -\mu_{q+1}\mathbf{D}_y\mathbf{Z}_y^{q} + \mathbf{D}_y\mathbf{Z}_y^{(q)} \end{bmatrix} \\
&= (1-\mu_{q+1}) \begin{bmatrix} \mathbf{D}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_y \end{bmatrix} \begin{bmatrix} \mathbf{Z}_x^{(q)} \\ \mathbf{Z}_y^{(q)} \end{bmatrix},
\end{aligned} \tag{3.2.22}$$

where superscript (q) denote the qth column of the associated matrix.

3.2.3 Correspondence Analysis

CA [88] was original created to analyse contingency tables – the joint embeddings are computed via the decomposition of the chi-squared statistic associated with this table. To begin with, CA first derives the *correspondence matrix* $\mathbf{P}$ from $\mathbf{R}$ by dividing its elements by the total sum of entries:

$$\mathbf{P} = [p_{ij}] = \left[\frac{r_{ij}}{s}\right], \text{ where } s = \sum_{i}^{m}\sum_{j}^{n} r_{ij} \tag{3.2.23}$$

Then, the differences between distributions of the rows and columns in the table are measured by so-called χ^2 *distances*, which are weighted Euclidean distances between normalised rows/columns (calculated by dividing row/column entries by their respective row/column total). In symbols, if we denote by r_i the ith row sum of $\mathbf{P}$, and by c_j its jth column sum, the χ^2 distance between the ith and the kth rows can be given by

$$d_{ik}^2 = \sum_{j=1}^{n} \frac{1}{c_j}\left(\frac{p_{ij}}{r_i} - \frac{p_{kj}}{r_k}\right)^2. \tag{3.2.24}$$

It measures the difference between the sample frequencies, with weights inversely proportional to the column totals. Here, the weight of a column total reflects the information this column provides to the identification of a given row, i.e., we consider columns that are used often do not provide much information as those columns that used rarely.

As the name suggests, χ^2 distances are closely related to the χ^2 statistic $\mathbf{X}^2$, which measures how far the set of samples are spread out from their average value. It is given by

$$\mathbf{X}^2 = s\sum_{i}^{m}\sum_{j}^{n} \frac{(p_{ij} - r_i c_j)^2}{r_i c_j} \tag{3.2.25}$$

where $r_i c_j$ is the expected value for the entry p_{ij} when assuming independence of rows and columns in the table.

The square roots of the terms in χ^2 statistic can be conveniently expressed as

$$\begin{aligned}\mathbf{\Omega} &= \left[\frac{p_{ij} - r_i c_j}{\sqrt{r_i c_j}}\right] \\ &= \mathbf{D}_x^{-\frac{1}{2}}(\mathbf{P} - \boldsymbol{r}\boldsymbol{c}^T)\mathbf{D}_y^{-\frac{1}{2}} \end{aligned} \tag{3.2.26}$$

where $\boldsymbol{r} = [r_1, r_2, \ldots, r_m]^\top$ and $\boldsymbol{c} = [c_1, c_2, \ldots, c_n]^\top$ are vectors comprising the row and column sums respectively. And $\mathbf{D}_x$ and $\mathbf{D}_y$ are the associated diagonal matrices formed by the vector of $\boldsymbol{r}$ and $\boldsymbol{c}$.

Let $\boldsymbol{e}_i, \boldsymbol{e}_k$ to be the ith and kth column of the $m \times m$ identity matrix respectively. In matrix form, we derive the CA representations for preserving the χ^2 distances in Eq. (3.2.24) by

$$\begin{aligned} d_{ik}^2 &= (\boldsymbol{e}_i - \boldsymbol{e}_k)^\top \mathbf{D}_x^{-1}(\mathbf{P} - \boldsymbol{r}\boldsymbol{c}^\top)\mathbf{D}_y^{-1}(\mathbf{P} - \boldsymbol{r}\boldsymbol{c}^\top)^\top \mathbf{D}_x^{-1}(\boldsymbol{e}_i - \boldsymbol{e}_k) \\ &= (\boldsymbol{e}_i - \boldsymbol{e}_k)^\top \mathbf{D}_x^{-\frac{1}{2}}(\mathbf{D}_x^{-\frac{1}{2}}(\mathbf{P} - \boldsymbol{r}\boldsymbol{c}^\top)\mathbf{D}_y^{-\frac{1}{2}})(\mathbf{D}_y^{-\frac{1}{2}}(\mathbf{P} - \boldsymbol{r}\boldsymbol{c}^\top)^\top \mathbf{D}_x^{-\frac{1}{2}})\mathbf{D}_x^{-\frac{1}{2}}(\boldsymbol{e}_i - \boldsymbol{e}_k) \\ &= \|(\boldsymbol{e}_i - \boldsymbol{e}_k)^\top \mathbf{D}_x^{-\frac{1}{2}}\mathbf{U}\boldsymbol{\Sigma}\|_2^2 \end{aligned} \tag{3.2.27}$$

where $\mathbf{U}$, $\mathbf{V}$ and $\boldsymbol{\Sigma}$ are the matrices containing the left and right singular vectors, and the ordered positive singular values ($\sigma_1 \geq \sigma_2 \geq \ldots$), respectively, of the χ^2 statistic matrix $\boldsymbol{\Omega}$ (other variations [102] of CA compute the output matrices equivalently from the SVD of $\mathbf{D}_x^{-\frac{1}{2}}\mathbf{R}\mathbf{D}_y^{-\frac{1}{2}}$).

Hence, the χ^2 distances between any row objects in $\mathbf{R}$ can be reconstructed exactly by the associated Euclidean distances between rows of $\mathbf{D}_x^{-\frac{1}{2}}\mathbf{U}\boldsymbol{\Sigma}$. In the other words, taking the row embedding according to

$$\hat{\mathbf{Z}}_x = \mathbf{D}_x^{-\frac{1}{2}}\mathbf{U}\boldsymbol{\Sigma}, \tag{3.2.28}$$

can preserve the row objects distances perfectly in the original space.

Similarly, the column χ^2 distances are retained by taking the embedding matrix as

$$\hat{\mathbf{Z}}_y = \mathbf{D}_y^{-\frac{1}{2}}\mathbf{V}\boldsymbol{\Sigma}. \tag{3.2.29}$$

In fact, CA computes the optimal k dimensional embedding as

$$\mathbf{Z}_x = \mathbf{D}_x^{-\frac{1}{2}}\mathbf{U}_k\boldsymbol{\Sigma}_k, \tag{3.2.30}$$

$$\mathbf{Z}_y = \mathbf{D}_y^{-\frac{1}{2}}\mathbf{V}_k\boldsymbol{\Sigma}_k, \tag{3.2.31}$$

where the subscript on $\mathbf{U}$, $\mathbf{V}$ and $\boldsymbol{\Sigma}$ indicates the k columns in $\mathbf{U}$, $\mathbf{V}$ and the fist k rows and columns of Σ are used. This representation captures quantities of χ^2 statistics (or inertia, see details in [88] [103]) in the sense of least squares approximation since

$$\|\mathbf{D}_x^{\frac{1}{2}}\mathbf{Z}_x\|_2^2 = \|\mathbf{U}_k\mathbf{\Sigma}_k\|_2^2 = \sum_{i=1}^{k}\mu_i^2 \leq \|\mathbf{\Omega}\|_F = \frac{\mathbf{X}^2}{s},$$
$$\|\mathbf{D}_y^{\frac{1}{2}}\mathbf{Z}_y\|_2^2 = \|\mathbf{V}_k\mathbf{\Sigma}_k\|_2^2 = \sum_{i=1}^{k}\mu_i^2 \leq \|\mathbf{\Omega}\|_F = \frac{\mathbf{X}^2}{s},$$

where $\|.\|_F$ represent the Frobenius norm of a matrix.

The CA algorithm is also summarised in Table 3.1.

3.2.4 Automatic Co-embedding with Adaptive Shaping

ACAS [28] is a parametric matrix factorisation method based on exploiting the commonalities amongst the existing models of CA, Latent Semantic Indexing and other methods proposed in [95], [96]. Formally, it firstly scales the relational matrix according to

$$\hat{\mathbf{R}} = \mathbf{S}_x^{-\frac{1}{2}}\mathbf{R}\mathbf{S}_y^{-\frac{1}{2}}, \tag{3.2.32}$$

where the scaling matrices $\mathbf{S}_x$ and $\mathbf{S}_y$ are generalisations 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

$$s_i^{(x)} = \begin{cases} 1, & \text{if } p = 0, \\ \left(\sum_{j=1}^{n} r_{ij}^p\right)^{\frac{1}{p}}, & \text{if } p \geq 1, \\ \max(r_{i1}, r_{i2}, \ldots, r_{in}), & \text{if } p = \infty, \end{cases} \tag{3.2.33}$$

and

$$s_j^{(y)} = \begin{cases} 1, & \text{if } p = 0, \\ \left(\sum_{i=1}^{m} r_{ij}^p\right)^{\frac{1}{p}}, & \text{if } p \geq 1, \\ \max(r_{1j}, r_{2j}, \ldots, r_{mj}), & \text{if } p = \infty. \end{cases} \tag{3.2.34}$$

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

$$\mathbf{Z}_x = \mathbf{S}_x^{-\alpha}\mathbf{U}_k\Sigma_k^{\beta}, \tag{3.2.35}$$
$$\mathbf{Z}_y = \mathbf{S}_y^{-\alpha}\mathbf{V}_k\Sigma_k^{\beta}, \tag{3.2.36}$$

where $\mathbf{U}_k$, $\mathbf{V}_k$ and $\mathbf{\Sigma}_k$ are as defined before.

To see why ACAS is constructed in this way, we compare it with various co-embedding methods in terms of the matrix on which Singular Value Decomposition (SVD) is performed and the co-embeddings computations in Table 3.1. It can be seen that all the existing co-embedding algorithms listed in Table 3.1 can be generated by the ACAS model by using different values for model parameters p, α and β. For example, with $p = 1$, $\alpha = \frac{1}{2}$ and $\beta = 0$, we obtain the BGP model, while setting $p = 0$, $\alpha = 0$ and $\beta = 1$ yields the Latent Semantic Indexing model (LSI).

Table 3.1: A summary of different co-embeddings methods, the second column shows the matrix on which SVD is performed, and the co-embedding computations are listed in the third and fourth columns. This table is taken from [28].

Method	SVD	$\mathbf{Z}_x$	$\mathbf{Z}_y$
LSI [104], CFRM [105]	$\mathbf{R}$	$\mathbf{U}_k$	$\mathbf{V}_k$
BGP	$\mathbf{D}_x^{-\frac{1}{2}}\mathbf{R}\mathbf{D}_y^{-\frac{1}{2}}$	$\mathbf{D}_x^{-\frac{1}{2}}\mathbf{U}_k$	$\mathbf{D}_y^{-\frac{1}{2}}\mathbf{V}_k$
CA	$\mathbf{D}_x^{-\frac{1}{2}}\mathbf{R}\mathbf{D}_y^{-\frac{1}{2}}$	$\mathbf{D}_x^{-\frac{1}{2}}\mathbf{U}_k\mathbf{\Sigma}_k$	$\mathbf{D}_y^{-\frac{1}{2}}\mathbf{V}_k\mathbf{\Sigma}_k$
CORT [28]	$\mathbf{R}$	$\mathbf{U}_k\left(\mathbf{\Sigma}_k + \mathbf{\Sigma}_k^2\right)^{\frac{1}{2}}$	$\mathbf{V}_k\left(\mathbf{\Sigma}_k + \mathbf{\Sigma}_k^2\right)^{\frac{1}{2}}$
ACAS [28]	$\mathbf{S}_x^{-\frac{1}{2}}\mathbf{R}\mathbf{S}_y^{-\frac{1}{2}}$	$\mathbf{S}_x^{-\alpha}\mathbf{U}_k\mathbf{\Sigma}_k^{\beta}$	$\mathbf{S}_y^{-\alpha}\mathbf{U}_k\mathbf{\Sigma}_k^{\beta}$

With only three model variables p, α and β, ACAS flexibly adapts the shape of the computed co-embeddings to the given input measurements in $\mathbf{R}$. But the model parameters have to be learned from only the input information. Thus, ACAS identifies its optimal parameters by maximising the conformity between the ground-truth matrix $\mathbf{R}$ and an approximate relation matrix $\mathbf{R}_z = \{r_{ij}^{(z)}\}$. $\mathbf{R}_z$ is computed from the model's output co-embeddings $\mathbf{Z}_x$ and $\mathbf{Z}_y$ as

$$r_{ij}^{(z)} = \exp\left(\frac{-\left\|\boldsymbol{z}_i^{(x)} - \boldsymbol{z}_j^{(y)}\right\|_2^2}{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left\|\boldsymbol{z}_i^{(x)} - \boldsymbol{z}_j^{(y)}\right\|_2^2}\right). \tag{3.2.37}$$

The mismatch between $\mathbf{R}$ and $\mathbf{R}_z$ is measured by a quantisation-based criterion

$$\min_{p,\alpha,\beta} D_Q(\mathbf{R}, \mathbf{R}_z) = \|Q(\mathbf{R}) - Q(\mathbf{R}_z))\|_F, \tag{3.2.38}$$

where $\|\cdot\|_F$ is the Frobenius norm, and $Q(\cdot)$ is a quantisation function that transforms the element values in a relational matrix $\mathbf{R}$ into a fixed set of quantities. Specifically, the quantisation employed by ACAS relies on the q-quantiles of all the values within $\mathbf{R}$, denoted by $\boldsymbol{p} = [p_1, \dots, p_{q-1}]^\top$. It then quantiles each ijth element of $\mathbf{R}$ according to

$$Q(r_{ij}) = \begin{cases} 1, & \text{if } r_{ij} \leq p_1 \\ t, & \text{if } p_{t-1} \leq r_{ij} \leq p_t, \quad t = 2, \dots, q-1 \\ q, & \text{if } p_{q-1} < r_{ij}. \end{cases} \tag{3.2.39}$$

Finally, this model is trained by a very simple optimisation procedure, such as grid search, genetic algorithm, or simulated annealing, with the predefined searching ranges of model parameters.

3.3 The Proposed Framework

3.3.1 Model Construction

We first consider the simpler problem of mapping the pairwise relationships contained in matrix $\mathbf{R}$ to a line. We let $\boldsymbol{z}_x = [z_{x_1}, \dots, z_{x_m}]^\top$ and $\boldsymbol{z}_y = [z_{y_1}, \dots, z_{y_n}]^\top$ 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 generic criterion for choosing a good map for the points $\boldsymbol{z}_y$ is to minimise a series of cost functions for all objects x_i, each expressed as

$$f_{x_i}(\boldsymbol{z}_y) = (z_{x_i} - z_{y_1})^2 w_{i1} + (z_{x_i} - z_{y_2})^2 w_{i2} + \ldots + (z_{x_i} - z_{y_n})^2 w_{in}. \tag{3.3.1}$$

This criterion is similar to embedding methods, such as the LE [56], where the distances between the embedded points are driven to correspond to those of the original patterns through similarity weights w_{ij}. In Eq.(3.3.1), 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 normalised 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.(3.3.1) to all embedded points $\{z_{x_i}\}_{i=1}^m$, generates m different minimising 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 optimisation through an aggregate cost function

$$\hat{\mathbf{F}}(\boldsymbol{z}_y) = \sum_{i=1}^m \Big(\sum_{j=1}^n r_{ij}\Big)^{\eta_1} f_{x_i}(\boldsymbol{z}_y), \tag{3.3.2}$$

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 minimisation 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 normalised weights estimated from $\mathbf{R}$ to Eq.(3.3.1) and substitute in Eq.(3.3.2) we have

$$\begin{aligned}
\hat{\mathbf{F}}(\boldsymbol{z}_y) &= \sum_{i=1}^m \Big(\sum_{j=1}^n r_{ij}\Big)^{\eta_1} \sum_{j=1}^n (z_{x_i} - z_{y_j})^2 w_{ij} \\
&= \sum_{i=1}^m \Big(\sum_{j=1}^n r_{ij}\Big)^{\eta_1} \sum_{j=1}^n (z_{x_i} - z_{y_j})^2 \frac{r_{ij}}{\sum_{j=1}^n r_{ij}} \\
&= \sum_{i=1}^m \sum_{j=1}^n (z_{x_i} - z_{y_j})^2 r_{ij}^{(x)},
\end{aligned} \tag{3.3.3}$$

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

$$\hat{\mathbf{G}}(\boldsymbol{z}_x) = \sum_{j=1}^n \sum_{i=1}^m (z_{y_j} - z_{x_i})^2 r_{ij}^{(y)}, \tag{3.3.4}$$

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 optimisation is discussed in Section 3.3.2.

The minimisation problems in Eqs.(3.3.3,3.3.4) can be expressed in matrix forms as

$$\hat{\mathbf{F}}(\boldsymbol{z}_y) = \boldsymbol{z}_x^\top \mathbf{D}_{r,x} \boldsymbol{z}_x + \boldsymbol{z}_y^\top \mathbf{D}_{c,x} \boldsymbol{z}_y - 2\boldsymbol{z}_x^\top \mathbf{R}_x \boldsymbol{z}_y, \tag{3.3.5}$$

$$\hat{\mathbf{G}}(\boldsymbol{z}_x) = \boldsymbol{z}_x^\top \mathbf{D}_{r,y} \boldsymbol{z}_x + \boldsymbol{z}_y^\top \mathbf{D}_{c,y} \boldsymbol{z}_y - 2\boldsymbol{z}_x^\top \mathbf{R}_y \boldsymbol{z}_y, \tag{3.3.6}$$

where $\mathbf{R}_x = \left[r_{ij}^{(x)}\right]$, $\mathbf{R}_y = \left[r_{ij}^{(y)}\right]$. $\mathbf{D}_{r,x}$ 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.(3.3.5,3.3.6) we have the equivalent objective functions

$$\mathbf{F}(\boldsymbol{z}_y) = \boldsymbol{z}_y^\top \mathbf{D}_{c,x} \boldsymbol{z}_y - 2\boldsymbol{z}_x^\top \mathbf{R}_x \boldsymbol{z}_y, \tag{3.3.7}$$

$$\mathbf{G}(\boldsymbol{z}_x) = \boldsymbol{z}_x^\top \mathbf{D}_{r,y} \boldsymbol{z}_x - 2\boldsymbol{z}_x^\top \mathbf{R}_y \boldsymbol{z}_y. \tag{3.3.8}$$

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

$$\mathbf{R}_x = \left[r_{ij} \left(\sum_{j=1}^{n} r_{ij} \right)^{\eta_1 - 1} \right] = \mathbf{D}_r^{\eta_1 - 1}\mathbf{R} = \mathbf{D}_r^{\eta_r}\mathbf{R}, \tag{3.3.9}$$

$$\mathbf{R}_y = \left[r_{ij} \left(\sum_{i=1}^{m} r_{ij} \right)^{\eta_2 - 1} \right] = \mathbf{R}\mathbf{D}_c^{\eta_2 - 1} = \mathbf{R}\mathbf{D}_c^{\eta_c}, \tag{3.3.10}$$

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

Given a vector $\boldsymbol{z}_x$, the minimisation 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 minimisation 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 optimisations, 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

$$\boldsymbol{z}_x^\top \mathbf{D}_{r,y} \boldsymbol{z}_x = 1, \tag{3.3.11}$$

$$\boldsymbol{z}_y^\top \mathbf{D}_{c,x} \boldsymbol{z}_y = \zeta. \tag{3.3.12}$$

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.

3.3.2 Co-Embedding Generation

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

$$\begin{aligned}\mathcal{L}(\boldsymbol{z}_y, \mu_1) &= \boldsymbol{z}_y^\top \mathbf{D}_{c,x} \boldsymbol{z}_y - 2\boldsymbol{z}_x^\top \mathbf{R}_x \boldsymbol{z}_y - \mu_1(\boldsymbol{z}_y^\top \mathbf{D}_{c,x} \boldsymbol{z}_y - \zeta) \\ &= (1-\mu_1)\boldsymbol{z}_y^\top \mathbf{D}_{c,x} \boldsymbol{z}_y - 2\boldsymbol{z}_x^\top \mathbf{R}_x \boldsymbol{z}_y + \mu_1 \zeta, \qquad (3.3.13)\end{aligned}$$

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

$$\frac{\partial \mathcal{L}(\boldsymbol{z}_y, \mu_1)}{\partial \boldsymbol{z}_y} = 2(1-\mu_1)\mathbf{D}_{c,x}\boldsymbol{z}_y - 2\mathbf{R}_x^\top \boldsymbol{z}_x = 0. \qquad (3.3.14)$$

Combining Eqs.(3.3.12,3.3.14), yields

$$\boldsymbol{z}_y = \pm\alpha(\boldsymbol{z}_x)\mathbf{D}_{c,x}^{-1}\mathbf{R}_x^\top \boldsymbol{z}_x, \qquad (3.3.15)$$

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.(3.3.7), leads to a simpler expression given by

$$\begin{aligned}\mathbf{F}(\boldsymbol{z}_y) &= \zeta \mp 2\alpha(\boldsymbol{z}_x)\boldsymbol{z}_x^\top \mathbf{R}_x \mathbf{D}_{c,x}^{-1} \mathbf{R}_x^\top \boldsymbol{z}_x \\ &= \zeta \mp 2\alpha(\boldsymbol{z}_x)\frac{\zeta}{\alpha(\boldsymbol{z}_x)^2} \; = \; \zeta \mp \frac{2\zeta}{\alpha(\boldsymbol{z}_x)}. \qquad (3.3.16)\end{aligned}$$

It can therefore be seen that, since $\zeta - \frac{2\zeta}{\alpha(\boldsymbol{z}_x)} < \zeta + \frac{2\zeta}{\alpha(\boldsymbol{z}_x)}$, the minimising embedding is obtained by the positive branch of Eq.(3.3.15) as

$$\boldsymbol{z}_y^* = \underset{\substack{\boldsymbol{z}_y \in \mathcal{R}^n, \\ \boldsymbol{z}_y^\top \mathbf{D}_{c,x} \boldsymbol{z}_y = \zeta}}{\operatorname{argmin}} \; \mathbf{F}(\boldsymbol{z}_y) = \alpha(\boldsymbol{z}_x)\mathbf{D}_{c,x}^{-1}\mathbf{R}_x^\top \boldsymbol{z}_x. \qquad (3.3.17)$$

We now consider the minimisation of $\mathbf{G}(\boldsymbol{z}_x)$, given the embedding $\boldsymbol{z}_y$ for the column objects, under the constraint Eq.(3.3.11). The associated Lagrangian is

$$\begin{aligned}\mathcal{L}(\boldsymbol{z}_x, \mu_2) &= \boldsymbol{z}_x^\top \mathbf{D}_{r,y} \boldsymbol{z}_x - 2\boldsymbol{z}_x^\top \mathbf{R}_y \boldsymbol{z}_y - \mu_2(\boldsymbol{z}_x^\top \mathbf{D}_{r,y} \boldsymbol{z}_x - 1) \\ &= (1-\mu_2)\boldsymbol{z}_x^\top \mathbf{D}_{r,y} \boldsymbol{z}_x - 2\boldsymbol{z}_x^\top \mathbf{R}_y \boldsymbol{z}_y + \mu_2, \qquad (3.3.18)\end{aligned}$$

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

$$\boldsymbol{z}_x^* = \underset{\substack{\boldsymbol{z}_x \in \mathfrak{R}^m, \\ \boldsymbol{z}_x^\top \mathbf{D}_{r,y} \boldsymbol{z}_x = 1}}{\operatorname{argmin}} \mathbf{G}(\boldsymbol{z}_x) = \beta(\boldsymbol{z}_y) \mathbf{D}_{r,y}^{-1} \mathbf{R}_y \boldsymbol{z}_y, \tag{3.3.19}$$

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 3.3.1, a desired co-embedding $(\boldsymbol{z}_x^*, \boldsymbol{z}_y^*)$ should mutually satisfy both optimisation problems. Consequently, using Eqs.(3.3.17,3.3.19), we can explicitly make use of this interdependency to express $\boldsymbol{z}_x^*$ via

$$\boldsymbol{z}_x^* = \beta(\boldsymbol{z}_y^*) \mathbf{D}_{r,y}^{-1} \mathbf{R}_y \; \alpha(\boldsymbol{z}_x^*) \mathbf{D}_{c,x}^{-1} \mathbf{R}_x^\top \boldsymbol{z}_x^* = \alpha(\boldsymbol{z}_x^*) \beta(\boldsymbol{z}_y^*) \mathbf{T} \boldsymbol{z}_x^*, \tag{3.3.20}$$

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.(3.3.20), 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 $\mathbf{T\Psi} = \mathbf{\Psi\Lambda}$, with $\mathbf{\Psi} = [\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_m]$ being the eigenvector matrix and $\mathbf{\Lambda} = \mathrm{diag}([\lambda_1, \lambda_2, \ldots, \lambda_m])$ the diagonal matrix of eigenvalues, we can take the sought embedding to be

$$\boldsymbol{z}_x^* = \frac{1}{\sqrt{\boldsymbol{\psi}_q^\top \mathbf{D}_{r,y} \boldsymbol{\psi}_q}} \boldsymbol{\psi}_q, \tag{3.3.21}$$

where the solving eigenvector $\boldsymbol{\psi}_q$ (the choice of q is addressed in Section 3.3.2) is scaled accordingly to satisfy the constraint in Eq.(3.3.11). Subsequently, the paired embedding for the column objects can be calculated directly from Eq.(3.3.17) 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

$$\begin{aligned} \alpha(\boldsymbol{z}_x^*)^2 \beta(\boldsymbol{z}_y^*)^2 &= \frac{1}{\left(\frac{1}{\alpha(\boldsymbol{z}_x^*)} \boldsymbol{z}_y^*\right)^\top \mathbf{R}_y^\top \mathbf{D}_{r,y}^{-1} \mathbf{R}_y \left(\frac{1}{\alpha(\boldsymbol{z}_x^*)} \boldsymbol{z}_y^*\right)} \\ &= \frac{1}{(\boldsymbol{z}_x^{*\top} \mathbf{R}_x \mathbf{D}_{c,x}^{-1}) \; \mathbf{R}_y^\top \mathbf{D}_{r,y}^{-1} \mathbf{R}_y \; (\mathbf{D}_{c,x}^{-1} \mathbf{R}_x^\top \boldsymbol{z}_x^*)} \\ &= \frac{1}{\boldsymbol{z}_x^{*\top} \mathbf{T}^\top \mathbf{D}_{r,y} \mathbf{T} \boldsymbol{z}_x^*} = \frac{1}{\lambda_q^2 \boldsymbol{z}_x^{*\top} \mathbf{D}_{r,y} \boldsymbol{z}_x^*} = \frac{1}{\lambda_q^2}, \end{aligned} \tag{3.3.22}$$

which rely on Eqs.(3.3.17,3.3.21).

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.(3.3.9,3.3.10), 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

$$\begin{aligned}
\mathbf{T} &= \mathbf{D}_{r,y}^{-1}\mathbf{R}\mathbf{D}_c^{\eta_c}\mathbf{D}_{c,x}^{-1}\mathbf{R}^\top\mathbf{D}_r^{\eta_r} \\
&= \mathbf{D}_r^{-\frac{\eta_r}{2}}\mathbf{D}_{r,y}^{-1}\mathbf{D}_r^{\frac{\eta_r}{2}}\mathbf{R}\mathbf{D}_c^{\frac{\eta_c}{2}}\mathbf{D}_{c,x}^{-1}\mathbf{D}_c^{\frac{\eta_c}{2}}\mathbf{R}^\top\mathbf{D}_r^{\frac{\eta_r}{2}}\mathbf{D}_r^{\frac{\eta_r}{2}} \\
&= \mathbf{D}_r^{-\frac{\eta_r}{2}}\mathbf{D}_{r,y}^{-\frac{1}{2}}\left(\mathbf{D}_{r,y}^{-\frac{1}{2}}\mathbf{D}_r^{\frac{\eta_r}{2}}\mathbf{R}\mathbf{D}_c^{\frac{\eta_c}{2}}\mathbf{D}_{c,x}^{-\frac{1}{2}}\right)\left(\mathbf{D}_{c,x}^{-\frac{1}{2}}\mathbf{D}_c^{\frac{\eta_c}{2}}\mathbf{R}^\top\mathbf{D}_r^{\frac{\eta_r}{2}}\mathbf{D}_{r,y}^{-\frac{1}{2}}\right)\mathbf{D}_{r,y}^{\frac{1}{2}}\mathbf{D}_r^{\frac{\eta_r}{2}} \\
&= \mathbf{P}^{-1}\mathbf{A}^\top\mathbf{A}\mathbf{P}, \qquad (3.3.23)
\end{aligned}$$

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.

Eigenvector selection

So far we have shown the form of the sought co-embedding $(\boldsymbol{z}_x^*, \boldsymbol{z}_y^*)$ from Eqs. (3.3.17,3.3.21). Because of the interdependency between the two sets of objects, we must minimise the two objective functions $\mathbf{F}(\boldsymbol{z}_y)$ and $\mathbf{G}(\boldsymbol{z}_x)$ simultaneously. From Eq.(3.3.16), 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 minimised. 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.(3.3.21).

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

$$\mathbf{T}\mathbf{1}_m = \mathbf{D}_{r,y}^{-1}\mathbf{R}_y\mathbf{D}_{c,x}^{-1}\mathbf{R}_x^\top\mathbf{1}_m = \mathbf{D}_{r,y}^{-1}\mathbf{R}_y\mathbf{1}_n = \mathbf{1}_m, \qquad (3.3.24)$$

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.(3.3.24) 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.(3.3.21) 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 3.3.1, 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.

3.3.3 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 visualisation 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 utilised.

In the computed co-embedding, the scale constraints of Eqs.(3.3.11,3.3.12) need to be maintained for all axes, as

$$\operatorname{diag}(\mathbf{Z}_x^\top \mathbf{D}_{r,y} \mathbf{Z}_x) = \mathbf{1}_k, \tag{3.3.25}$$

$$\operatorname{diag}(\mathbf{Z}_y^\top \mathbf{D}_{c,x} \mathbf{Z}_y) = \boldsymbol{\zeta}. \tag{3.3.26}$$

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 optimisation. Specifically, from Eq.(3.3.23), 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, $\mathbf{P}\boldsymbol{\Psi}$ is orthogonal. This is equivalent to $\boldsymbol{\Psi}^\top\mathbf{P}^2\boldsymbol{\Psi}$ 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

$$\mathbf{Z}_x^{(q)} = \frac{1}{\sqrt{\boldsymbol{\psi}_{q+1}^\top \mathbf{D}_{r,y} \boldsymbol{\psi}_{q+1}}} \boldsymbol{\psi}_{q+1}, \tag{3.3.27}$$

$$\mathbf{Z}_y^{(q)} = \alpha(\mathbf{Z}_x^{(q)}, \zeta_q)\, \mathbf{D}_{c,x}^{-1} \mathbf{R}_x^\top \mathbf{Z}_x^{(q)}, \tag{3.3.28}$$

where the quantity $\alpha(\cdot)$ is as defined for Eq.(3.3.15), but it now depends also on the qth scale parameter and is equal to $\sqrt{\frac{\zeta_q}{\mathbf{Z}_x^{(q)\top} \mathbf{R}_x \mathbf{D}_{c,x}^{-1} \mathbf{R}_x^\top \mathbf{Z}_x^{(q)}}}$.

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 3.3.1 to scale the objectives, and as they indirectly parameterise 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 3.3.4. 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

$$\mathbf{Z}_x^{(q)} = \left(\frac{\lambda_{q+1}}{\lambda_2}\right)^\gamma \frac{1}{\sqrt{\boldsymbol{\psi}_{q+1}^\top \mathbf{D}_{r,y} \boldsymbol{\psi}_{q+1}}} \boldsymbol{\psi}_{q+1}, \tag{3.3.29}$$

$$\mathbf{Z}_y^{(q)} = \frac{\xi}{\sqrt{\lambda_{q+1}}} \mathbf{D}_{c,x}^{-1} \mathbf{R}_x^\top \mathbf{Z}_x^{(q)}. \tag{3.3.30}$$

3.3.4 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 [28], 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 minimise the normalised sum of the element-wise products $\frac{\mathrm{Tr}(\mathbf{R}\mathbf{Q}^\top)}{\sqrt{\mathrm{Tr}\left(\mathbf{R}\mathbf{R}^\top\right)\mathrm{Tr}\left(\mathbf{Q}\mathbf{Q}^\top\right)}}$ 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 quantisation have previously been used in the evaluation of ACAS [28].

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 neighbourhoods 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 neighbours 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 neighbours of the ith row object. The neighbourhood is established by using the raw similarities within $\mathbf{R}$, and the parameters k_r and k_c which can be pre-assigned 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 neighbours here are based on the smallest entries in the columns or rows).

This type of mutual neighbourhood 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 minimising the quantity

$$\Gamma(\mathbf{R}, \mathbf{Q}) = \mathbf{1}_m^\top \left(K(\mathbf{R}) - K(\mathbf{Q}) \odot K(\mathbf{R}) \right) \mathbf{1}_n, \tag{3.3.31}$$

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 neighbourhoods in $\mathbf{R}$ but not $\mathbf{Q}$. The measure assumes that loss of local neighbourhood structure from the original similarity matrix breaks down the initial requirement for the co-embedding to preserve local proximity information.

The overall set of operations for the proposed method is summarised in Algorithm 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 $O(mn)$.

3.4 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 [106]. We examine both qualitative and quantitative aspects of the comparison, which correspond to effectiveness in data visualisation and supervised machine learning. We split the experimentation into three main parts, solving three different data visualisation and analysis tasks using a total of 15 datasets:

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

```
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
    neighbourhood control parameters $k_r$, $k_c$.
Initialisation:
 1: Set restrictions on the search range of the model parameters, as: $\eta_1, \eta_2 \in [0, 10]$,
    $\xi \in (0, 3]$, and $\gamma \in [0, 3]$.
 2: Set some starting values for these model parameters (depending on the search
    method employed).
Main loop: (repeated as long as step $(10)$ cannot reduce the model cost $\Gamma(\mathbf{R}, \mathbf{Q})$
    any further):
 1: Set $\mathbf{D}_r = \text{diag}(\mathbf{R}\mathbf{1}_n)$ and $\mathbf{D}_c = \text{diag}(\mathbf{R}^\top \mathbf{1}_m)$.
 2: 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.(3.3.9,3.3.10).
 3: Set $\mathbf{D}_{r,y} = \text{diag}(\mathbf{R}_y \mathbf{1}_n)$ and $\mathbf{D}_{c,x} = \text{diag}(\mathbf{R}_x^\top \mathbf{1}_m)$.
 4: Construct the matrix $\mathbf{T} = \mathbf{D}_{r,y}^{-1}\mathbf{R}_y\mathbf{D}_{c,x}^{-1}\mathbf{R}_x^\top$.
 5: Perform an eigen-decomposition of $\mathbf{T}$, where $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_m]$ is the eigenvector
    matrix and $\boldsymbol{\Lambda} = \text{diag}([\lambda_1, \ldots, \lambda_m])$ contains the eigenvalues in descending
    order.
 6: Calculate each $q$th column (where $q = 1, \ldots, k$) of $\mathbf{Z}_x$ and of $\mathbf{Z}_y$ using
    Eqs.(3.3.29,3.3.30).
 7: 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 $\mathbf{Q} \in \mathcal{R}^{m \times n}$.
 8: Calculate the neighbourhood indicator structures $K(\mathbf{R})$ and $K(\mathbf{Q})$, as defined
    in Section 3.3.4.
 9: Compute the model cost $\Gamma(\mathbf{R}, \mathbf{Q})$ using Eq.(3.3.31).
10: If the cost is less than the minimum found so far, store the values of the current
    model parameters $\eta_1$, $\eta_2$, $\xi$ and $\gamma$ and update them to the next search values
    (depending on the employed search procedure and resolution of the search).
Output: The optimal co-embedding $(\mathbf{Z}_x, \mathbf{Z}_y)$ and their associated optimal model
    parameters.
```

- The reconstruction of the 2D distribution of data objects given the partial similarities between them (Section 3.4.1). 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 3.4.2). Four document collections containing clinical trials [85], Reuters new articles [85], 20 newsgroup documents[2] and online reviews [107] are used.

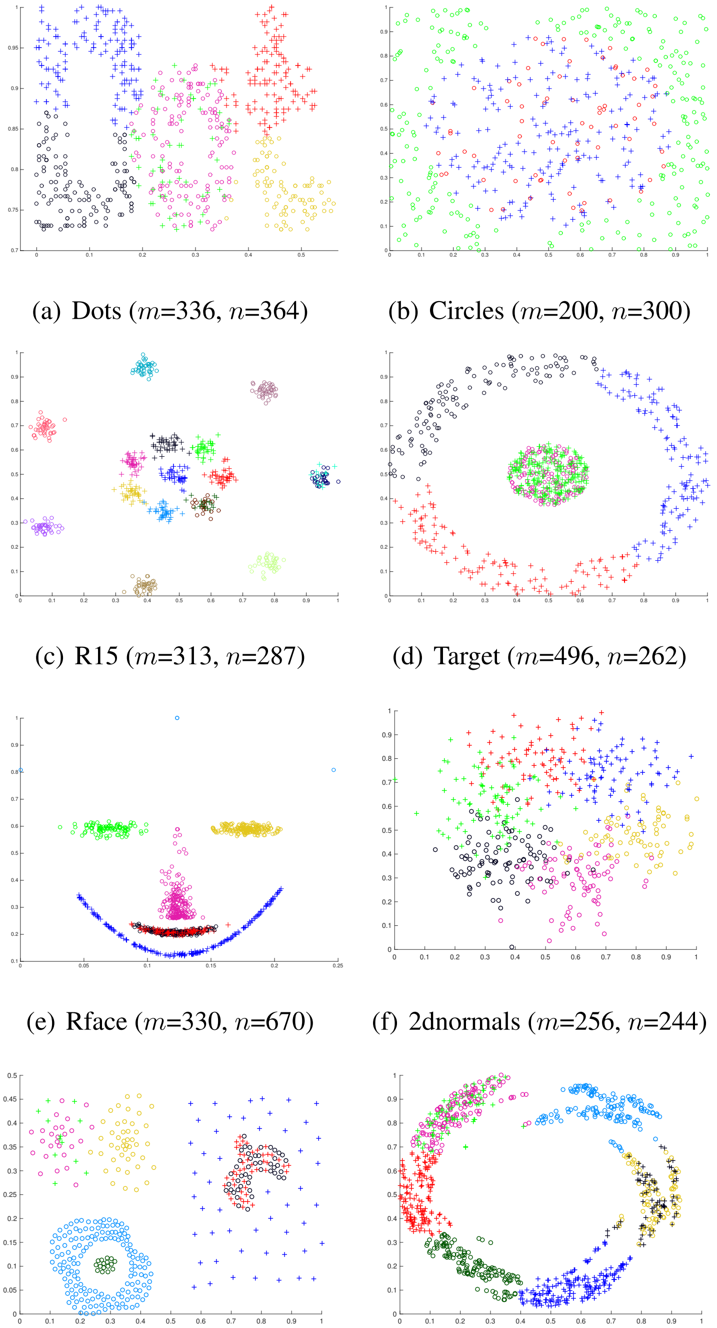
- The learning of low-dimensional representations of objects based on link information contained in knowledge graphs (Section 3.4.3). We use three datasets [108] with the citation networks between the Cora and Citeseer documents, as well as the co-occurrence network between industrial companies.

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 neighbourhood 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 3.4.4 analyses 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.(3.3.31) to be its minimising 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.

[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`. The co-cluster points in each data sets are randomly assigned to the two groups.

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

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



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

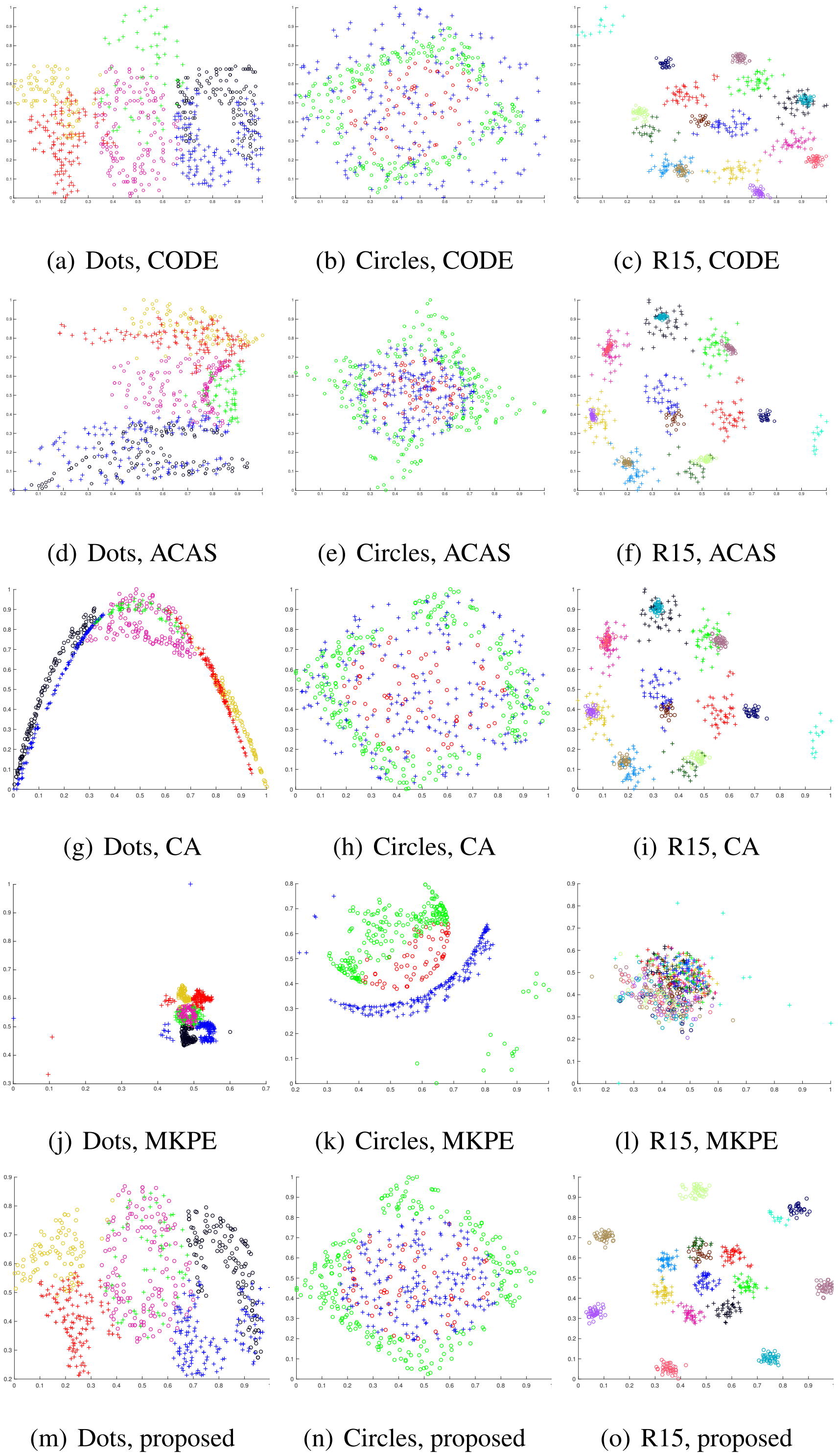
3.4.1 Reconstruction of Synthetic 2D Data Points

The eight synthetic datasets are shown in Figure 3.1, 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

$$r_{ij} = \exp\left(\frac{-mn\|\boldsymbol{x}_i - \boldsymbol{y}_j\|_2^2}{\sum_{i=1}^{m}\sum_{j=1}^{n}\|\boldsymbol{x}_i - \boldsymbol{y}_j\|_2^2}\right), \tag{3.4.1}$$

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 3.2-3.4. It can be observed that the representations generated by the proposed method resemble more the original arrangements of Figure 3.1. 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 3.3(b) and that of CA in Figure 3.3(e) reliably represent the eyes and nose components of the original set in Figure 3.1(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 3.3(e) totally distorts the relative positions of the different components. For the Target dataset of Figure 3.1(d), CA, ACAS and CODE in Figure 3.3(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 3.1(d), however, CODE segregates the co-cluster, while ACAS preserves it but compresses one of the structures. The proposed method is shown in Figure 3.3(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



(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

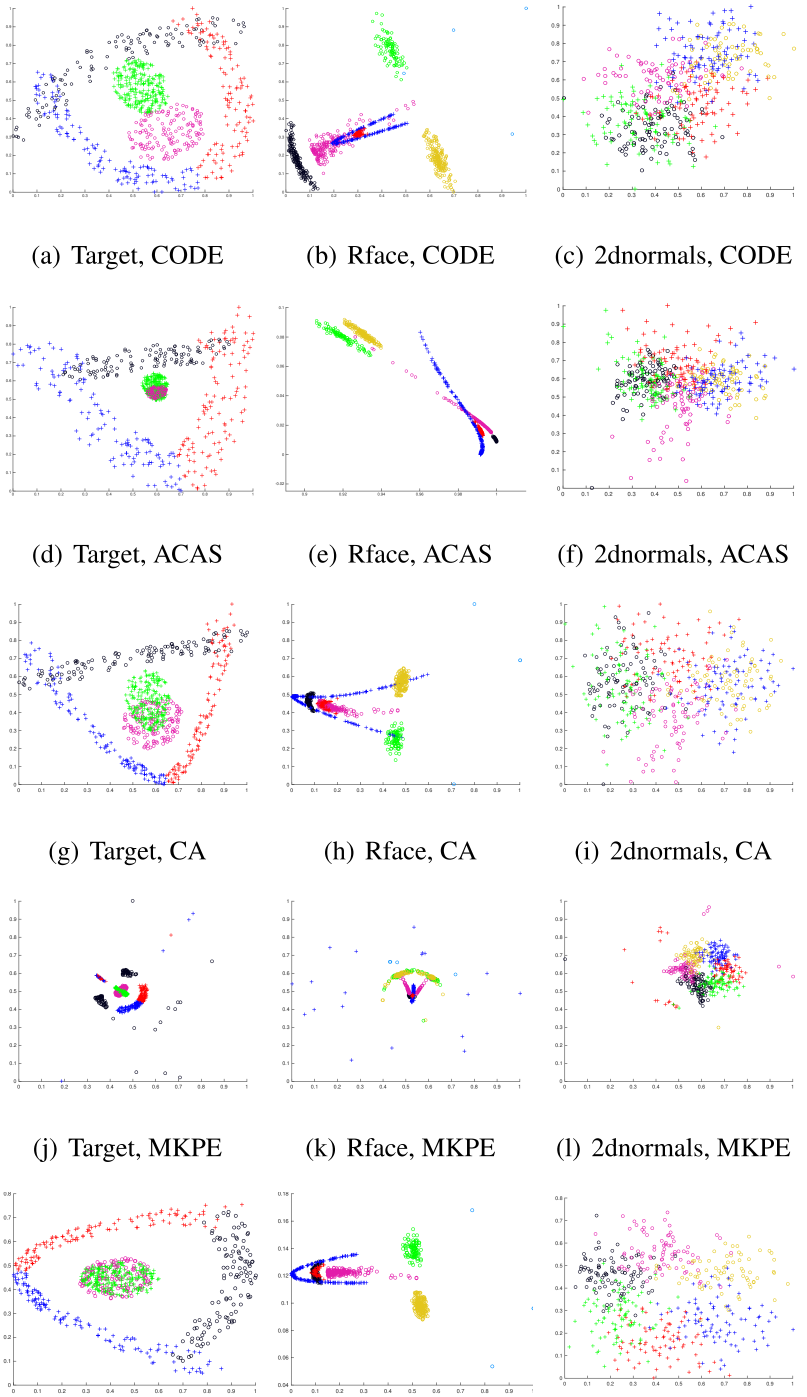
(m) Dots, proposed (n) Circles, proposed (o) R15, proposed

Figure 3.2: Co-embeddings generated by different algorithms, for the synthetic datasets of Dots, Circles and R15 displayed in Figure 3.1. Co-embedding axes are scaled within $[0, 1]$.

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.

3.4.2 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 centre for each document class. Then, we calculate the Euclidean distance between the words and each class centre in the co-embedding space. Sets of words that are closest to the document class centres 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 classifier for all experiments in this section. The classifier uses, for simplicity, a linear kernel, without any data scaling or standardisation applied to the predictors. It employs sequential minimal



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

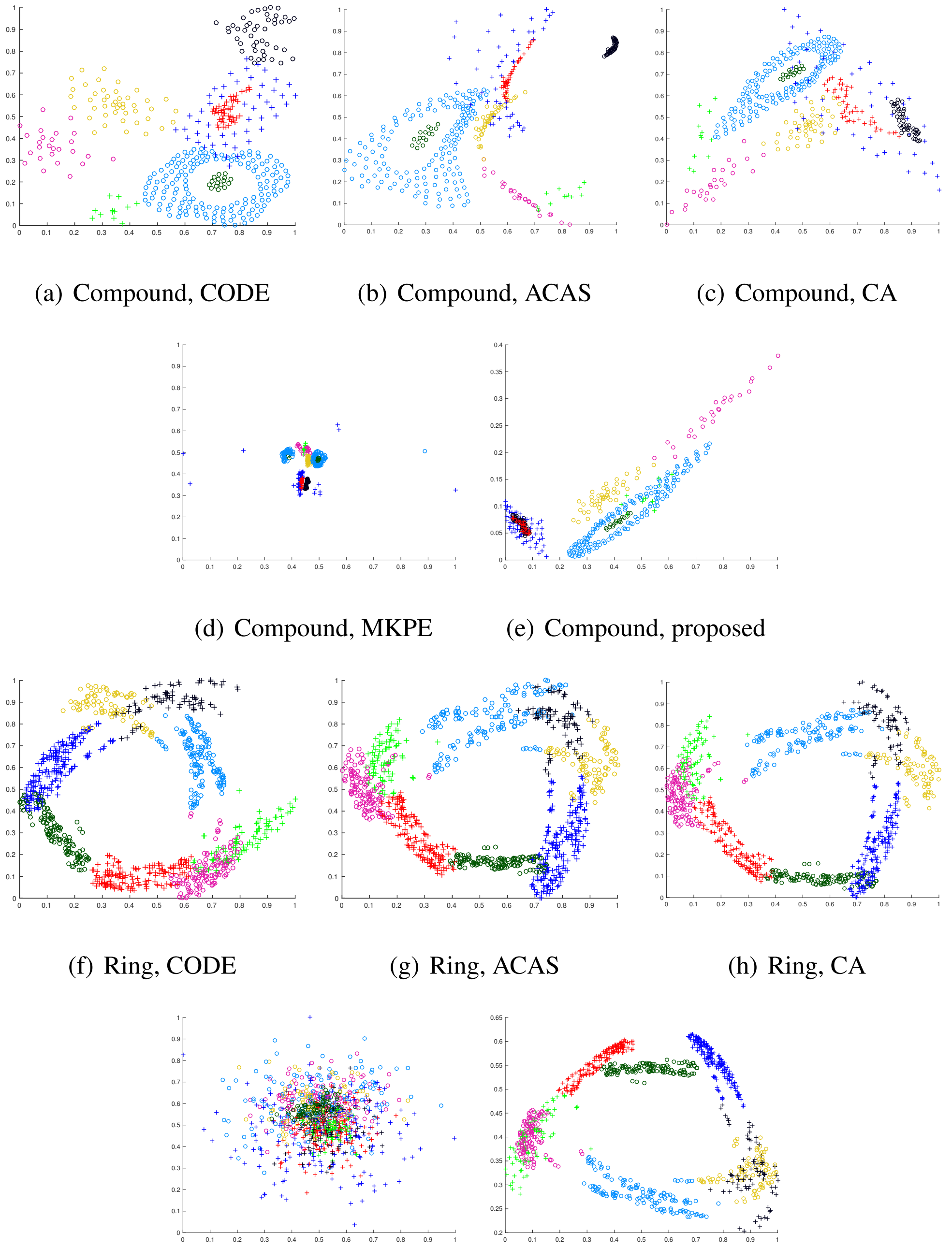
(d) Target, ACAS (e) Rface, ACAS (f) 2dnormals, ACAS

(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 3.3: Co-embeddings generated by different algorithms, for the synthetic datasets of Target, Rface and 2dnormals displayed in Figure 3.1. 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

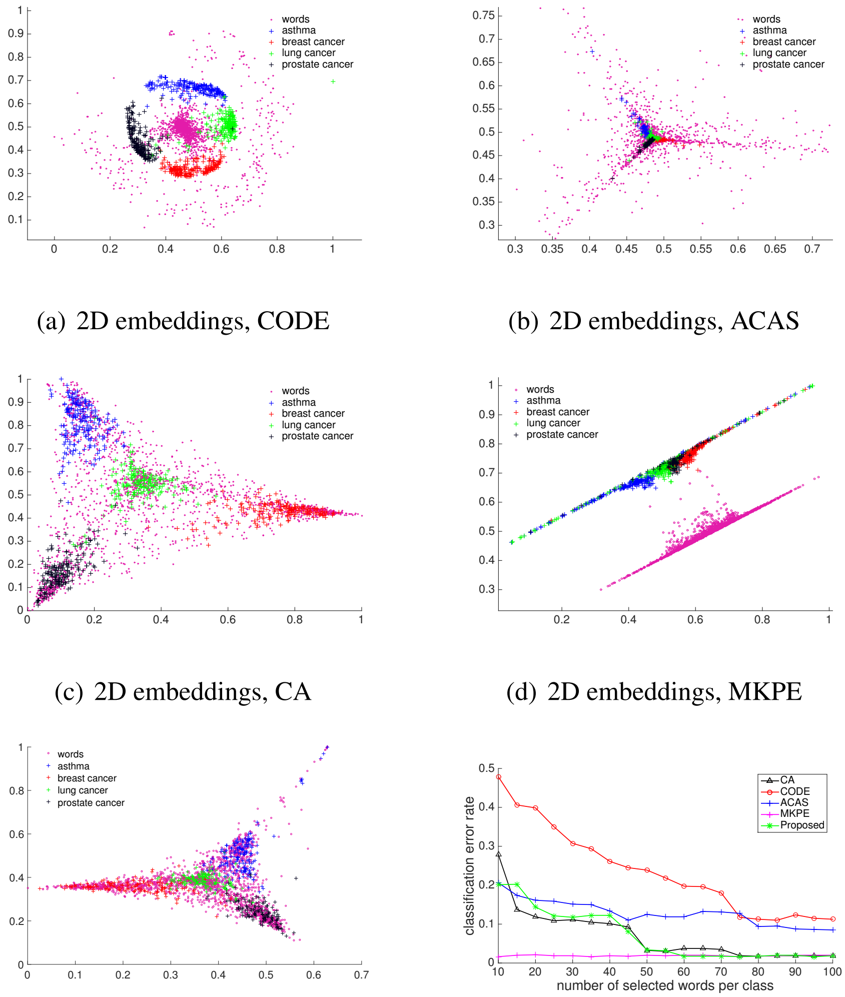
(i) Ring, MKPE (j) Ring, proposed

Figure 3.4: Co-embeddings generated by different algorithms, for the synthetic datasets compound and ring displayed in Figure 3.1. Co-embedding axes are scaled within $[0, 1]$.

optimisation for finding the optimal hyperplane, and a regularisation 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 3.5, 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 3.5(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 (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 visualisation 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 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, leukaemia, depression and schizophrenia. Another dataset is the Reuters news article collection,



(a) 2D embeddings, CODE

(b) 2D embeddings, ACAS

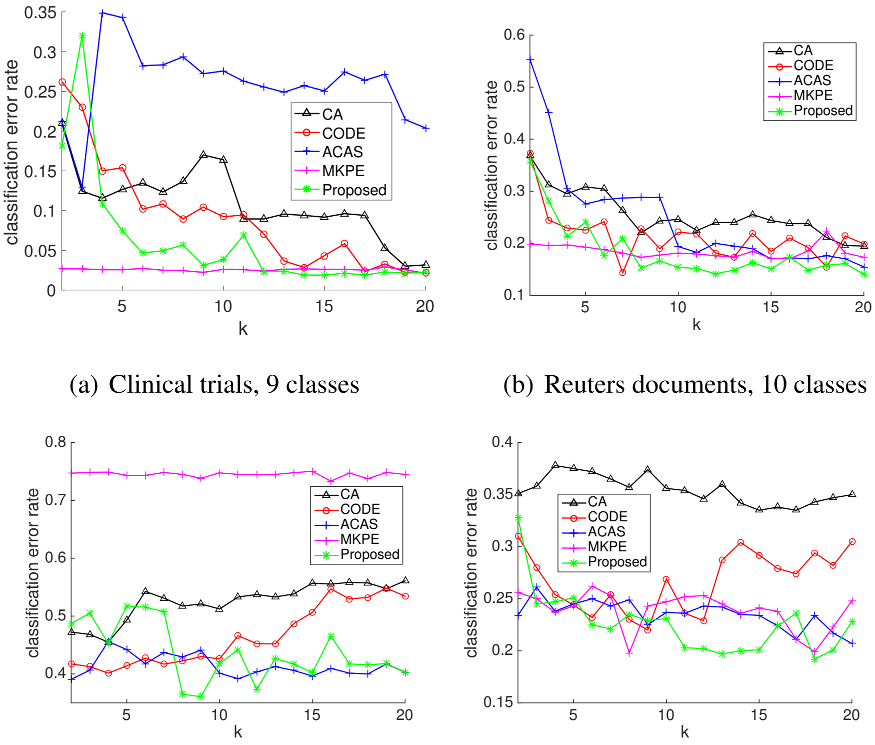
(c) 2D embeddings, CA

(d) 2D embeddings, MKPE

(e) 2D embeddings, proposed

(f) Classification error comparison

Figure 3.5: 2D demonstration of co-embeddings generated for 800 clinical trials and 1,780 words belonging to four classes by different methods and their classification error comparison. Each document (marked by “+”) is a member of group $\mathcal{X}$ and belongs to one of the four topics (plotted in different colour). Each member of group $\mathcal{Y}$ (marked by “•”) is a word object.



(a) Clinical trials, 9 classes

(b) Reuters documents, 10 classes

(c) 20Newsgroups documents, 20 classes

(d) Online reviews, 5 classes

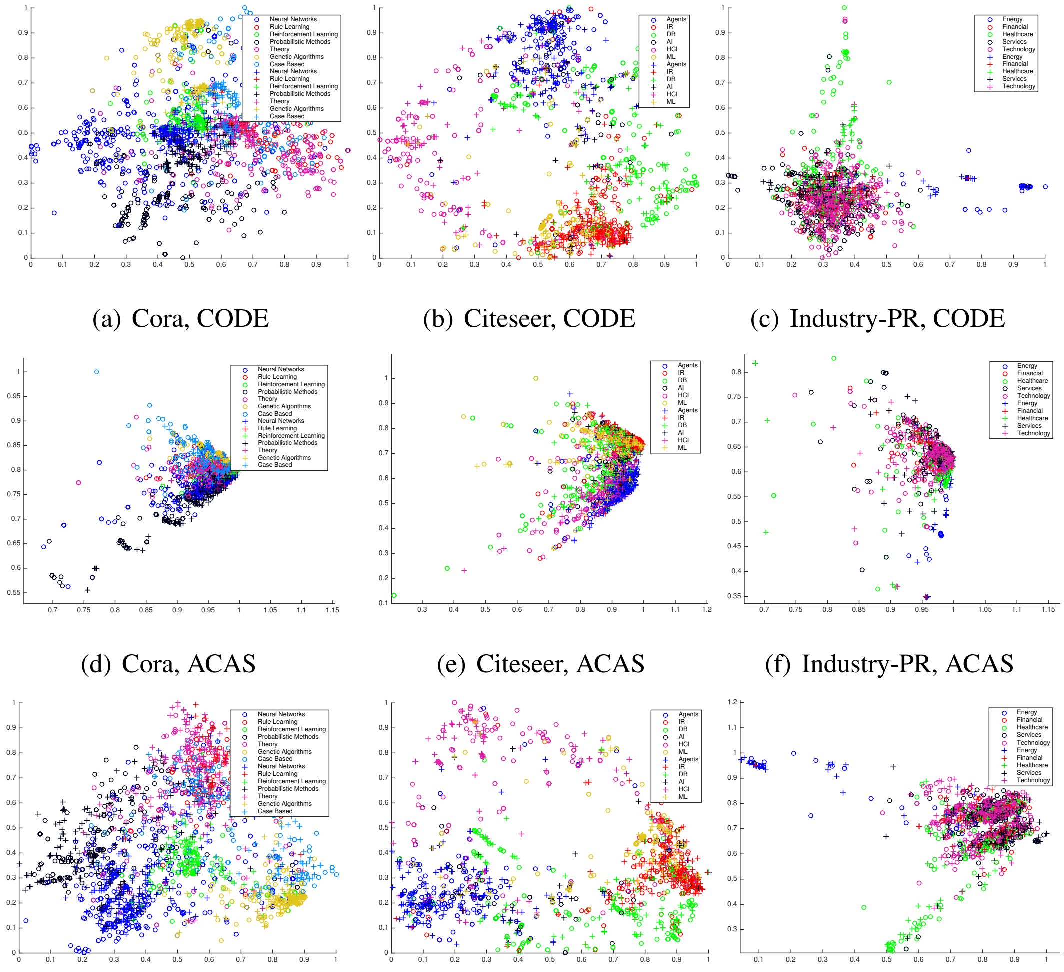
Figure 3.6: Comparison of the classification error rates of different algorithms, for varying the number k of the selected words that are closest to each class centre using different document collections.

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 **R**. As all datasets contain multiple classes, it is insufficient to learn 2D embeddings to characterise 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 3.6 that the proposed method possesses lower or comparable error rates compared to existing ones. Overall, taking into account all the results demonstrated in Figures 3.5 and 3.6, the proposed method exhibits the most consistently good performance in preserving the relation measurements between documents and words.

3.4.3 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 analyse 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 **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 **D**, is used to obtain the elements of the similarity matrix **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 [109] to examine how well the learned embeddings preserve the relation information in **R**. The goal is to compare the ranks of the closeness between each row object and all the column

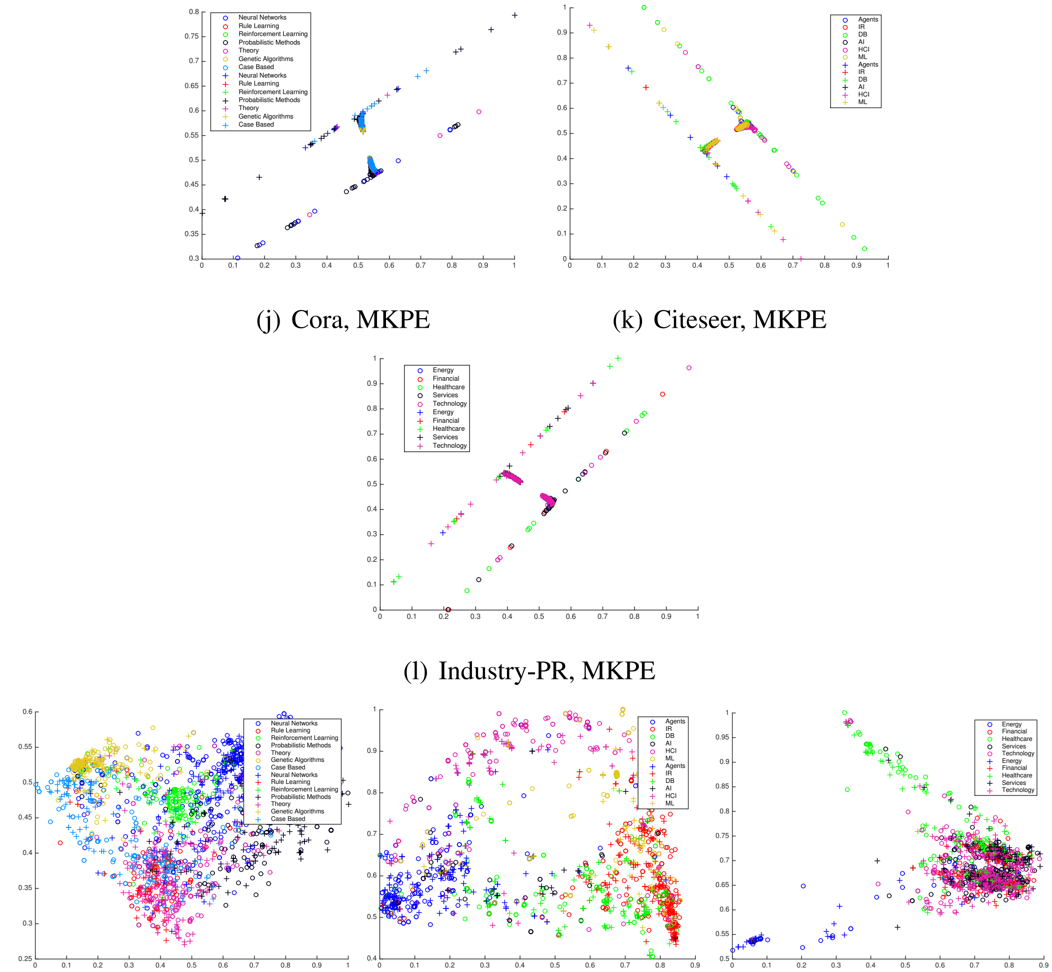


(a) Cora, CODE (b) Citeseer, CODE (c) Industry-PR, CODE

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

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

Figure 3.7: 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 colours. *(cont.)*



(j) Cora, MKPE (k) Citeseer, MKPE

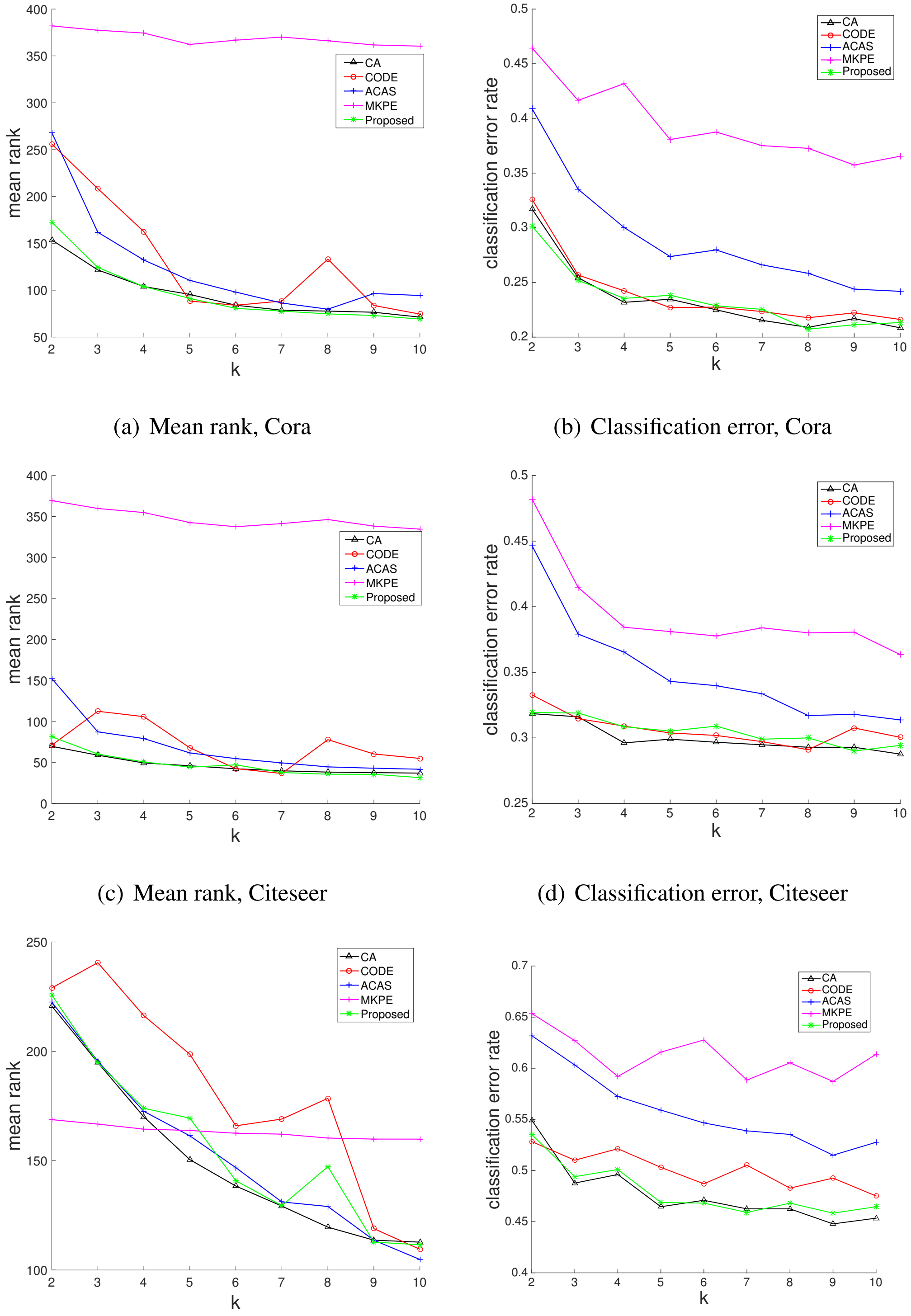
(l) Industry-PR, MKPE

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

Figure 3.7 (Cont.): 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 colours.

objects based on **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 **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



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

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-neighbour classifier.

Firstly, we visually demonstrate the learned 2D co-embeddings for the three datasets in Figures 3.7 and 3.7 (Cont.). 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 3.7(b) and 3.7(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 3.8 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.

3.4.4 Further Analysis of the Proposed Method

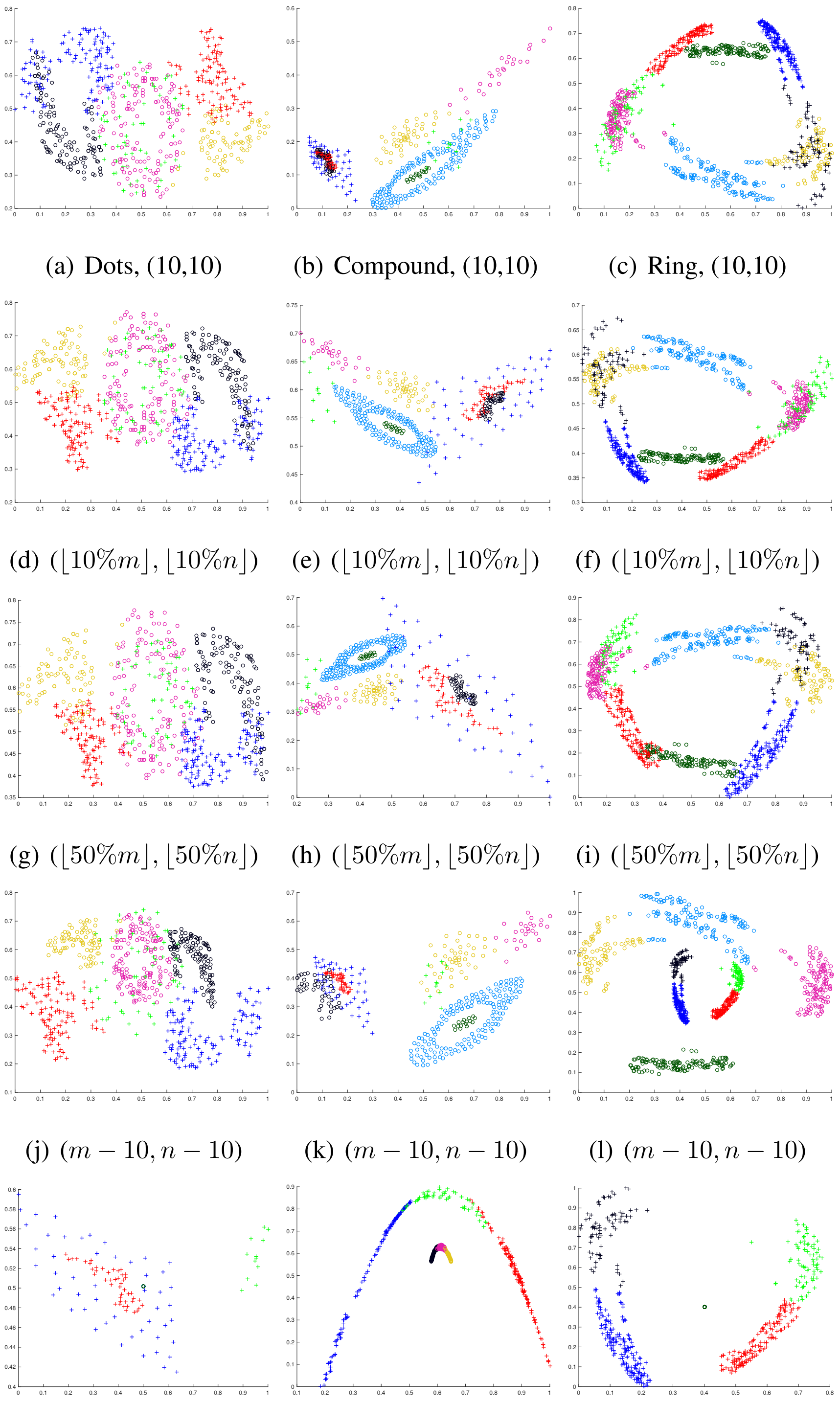
So far, the proposed algorithm provides the best performance for the synthetic datasets, while MKPE performs the worst (see Section 3.4.1). For document-word representation learning, the proposed algorithm and MKPE yield better quantitative performances, but MKPE provides separate document and word embedding distributions, which are less attractive for visualisation purposes (see Section 3.4.2). With regard to link data evaluation, the proposed algorithm and CA provide better performances in terms of mean rank and classification error (see Section 3.4.3). 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 quantisation parameter for ACAS, and the neighbourhood 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 quantisation parameter, especially when processing data with complex geometric distributions (see Figures 3.2(e) and 3.3(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 neighbourhood 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 neighbourhood structure than to enforce a global matching that considers both proximate and distant objects. To demonstrate the effect of varying settings of k_r and k_c, in Figures 3.9 and 3.10 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 neighbours, as well as $k_r = m - 10$ and $k_c = n - 10$ to exemplify the case of large neighbourhood ranges that is close to the extreme case of all the objects being considered.

We compare Figures 3.9 and 3.10 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 neighbours 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



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

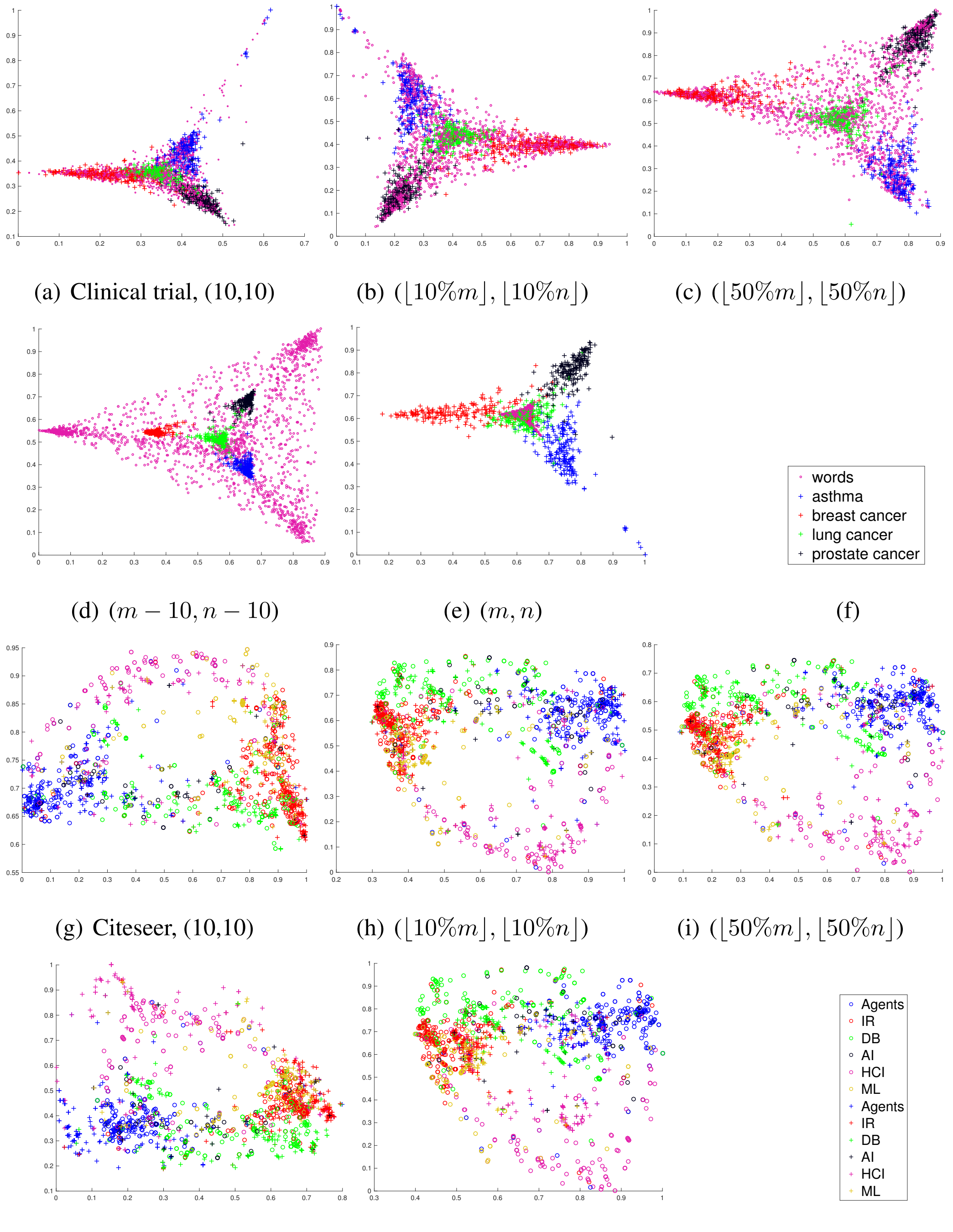
(d) $(\lfloor 10\% m \rfloor, \lfloor 10\% n \rfloor)$ (e) $(\lfloor 10\% m \rfloor, \lfloor 10\% n \rfloor)$ (f) $(\lfloor 10\% m \rfloor, \lfloor 10\% n \rfloor)$

(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 3.9: 2D embeddings generated by the proposed algorithm with varying settings of the neighbourhood 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) $(\lfloor 10\%m \rfloor, \lfloor 10\%n \rfloor)$

(c) $(\lfloor 50\%m \rfloor, \lfloor 50\%n \rfloor)$

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

(e) (m, n)

(f)

(g) Citeseer, (10,10)

(h) $(\lfloor 10\%m \rfloor, \lfloor 10\%n \rfloor)$

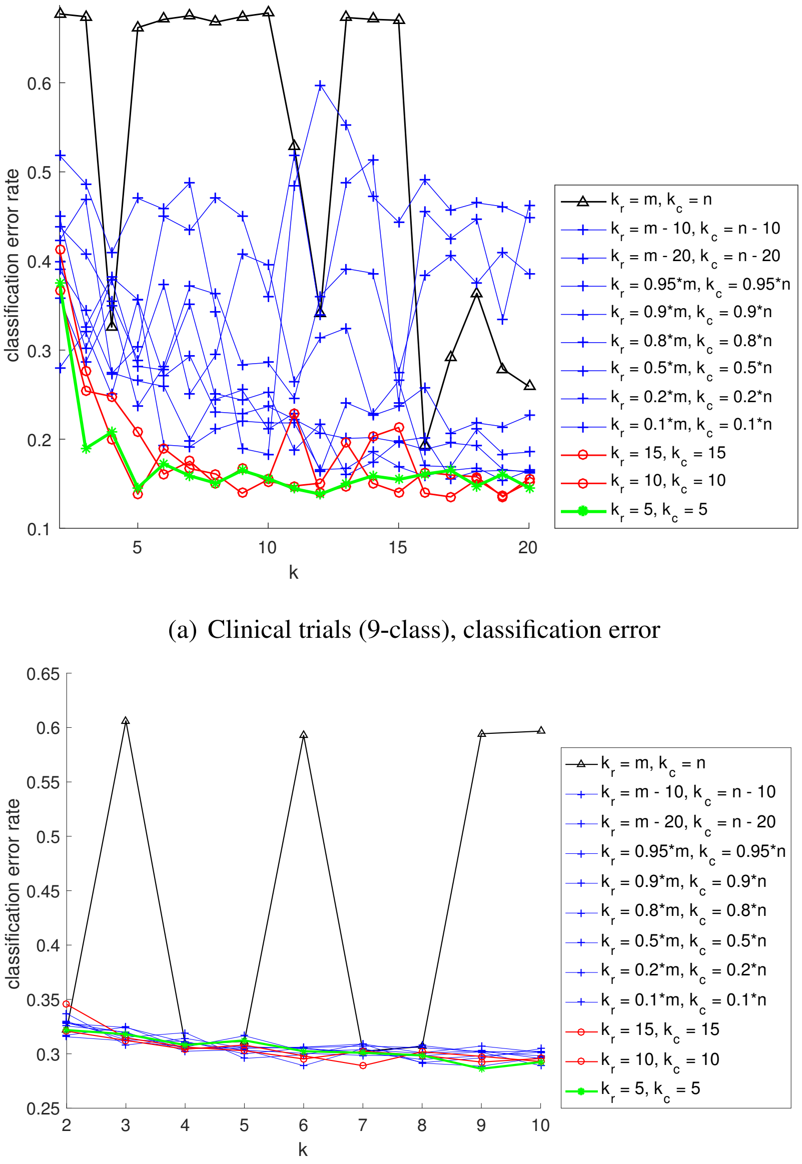
(i) $(\lfloor 50\%m \rfloor, \lfloor 50\%n \rfloor)$

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

(k) (m, n)

(l)

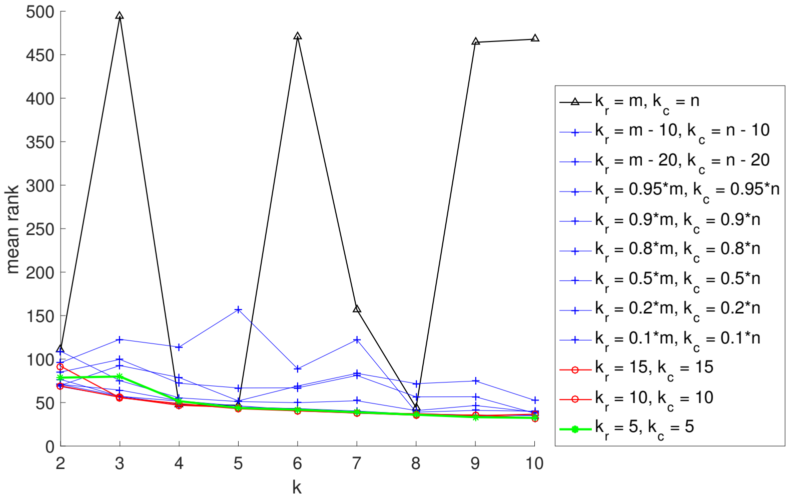
Figure 3.10: 2D embeddings generated by the proposed algorithm with varying settings of the neighbourhood control parameters k_r and k_c using the: (a-e) 4-class clinical trials, and (f-j) Citeseer dataset.



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

(b) Citeseer, classification error

Figure 3.11: Performance comparison of the proposed algorithm under varying settings of the neighbourhood 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. *(cont.)*



(c) Citeseer, mean rank

Figure 3.11 (Cont.): Performance comparison of the proposed algorithm under varying settings of the neighbourhood 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.

shape of the embedded data patterns starts to show significant distortion when large neighbours, e.g., $k_r = m - 10$ and $k_c = n - 10$, are used; for example Figure 3.9(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 3.9(m) some classes almost shrink to single points, and in Figure 3.10(e), the words shrink to the centre 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 Figures 3.11 and 3.11 (Cont.). 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 neighbourhood is kept small, similarly good performances can be obtained.

Finally, we compare the computational time of the proposed and existing methods

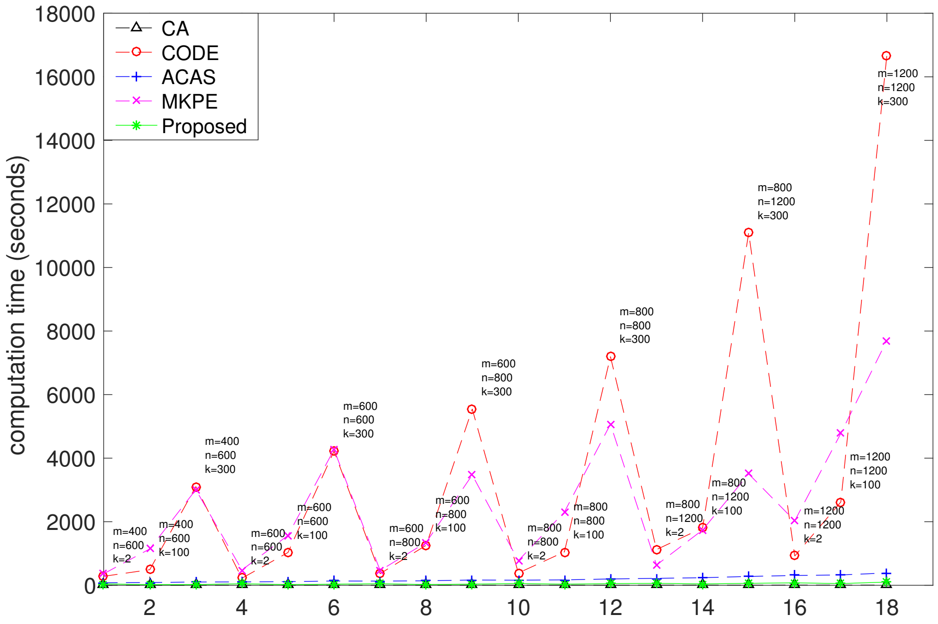


Figure 3.12: Computational cost comparison of different methods for increasing number of data size (n, m) and embedding dimension (k). The incremental integer values on x-axis mark the different settings of data size.

in Figure 3.12. It can be seen that gradient-based optimisation methods 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.

3.5 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 visualisation 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 optimisation to be achieved via simple matrix factorisation. 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.

The proposed work has some limitations. In the model construction, the distances between the embedded points are expected to correspond to those of the original patterns. This implicitly assumes that the original association values are comparable and can be sorted within either rows or columns. But this assumption can hardly be true in most real-world datasets. For example, the co-occurrence rates between documents and terms are inherently multi-modality that they should be explained via different unimodal sources. Also, the model parameters are identified by matching the useful mutual neighbourhood patterns in the original data and the generated co-embeddings. Though these neighbourhood patterns do correspond to important structural characteristics of either a joint embedding or a relational matrix, they might not be well-marked in an extremely sparse input — due to the sparsity of the input relational matrix, a large proportion of mutual neighbourhoods is identified by association of 0s.

Instead of unfolding the multi-modality information directly from the relational measurements, the multi-modalities of the associations can also be assorted and provided in the first place. In particular, the data can be represented by a multi-relational graph with multiple edges correspond to different types of relations. There are rich sources for such data type and this so-called multi-relational data forms our topic of study in the next chapter.

Chapter 4

Knowledge Graph Embedding

4.1 Introduction

The above chapter deals with bipartite relations with positive numerical quantities as the associations between different objects. When the associations are accurate enough to reflect objects' latent geometric structure, it has been demonstrated that some of the heterogeneous co-embedding algorithms are capable of recovering the original data patterns in the embedding space. But in the real world cases, the relations may come from multiple sources with different functions that cannot be simply explained as a measure of closeness between objects. This gives rise to the need for processing the multi-relational data [110], where the objects are interlinked by various relation types. Such data representation is very versatile, it can adapt to capturing various relations by including the attributes, correlations or classes of related objects. For example, the attributes of an object can be readily represented by introducing relation type "hasAttrribute" as edges connecting this object and its associated attributes. The main resource of multi-relational data is the web-based KGs, also referred to as knowledge bases [70]. A KG stores information in a graph structured format, such as a directed graph whose nodes (entities) represent the objects and edges (links) correspond to the relation types between objects. An example of a small KG is shown in Figure 4.1.

In the recent years, much work has been invested into the construction of

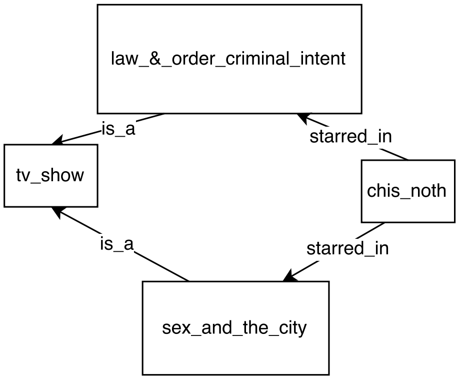


Figure 4.1: Real world facts stored as a KG, of which the triplet form is expressed as (head_entity, link, tail_entity), i.e. *(chris_noth, starred_in, sex_and_the_city), sex_and_the_city, is_a, tv_show).*

large KGs, including Wordnet [49], YAGO [111], DBpedia [112], Freebase [113], NELL [114] and the Google's Knowledge Vault project [115]. These contain highly structured information and are useful in many artificial intelligence related tasks, i.e. word-sense disambiguation [71, 72], search engine [73, 74], question answering [75]. However, despite being very large (usually containing millions of nodes and billions of edges), most KGs are very noisy and far from being complete, because large databases are either constructed collaboratively by an open group of volunteers or automatically extracted from unstructured or semi-structured text via rules, regular expressions, or machine learning and natural language processing techniques [116]. Taking Freebase as an example, which is a large collaborative knowledge base harvested from resources, such as individual and user-submitted Wiki contributions, there are 71% of around 3 million people with no known place of birth [117] within the database. Consequently, one major goal of KG analysis is to develop numerical models that suggest the likely truth of new facts and correct unlikely facts based on the existing data within KGs.

Since the KGs can correspond to massive volumes of knowledge, it is often prohibitively expensive to subject them for processing to symbolic models [118–120] or inference models [20, 121–127]. Latent representation models have therefore been receiving increasing attention. These are capable of embedding entities into a continuous vector space and converting links to mathematical operations (e.g., linear, bilinear transformation, etc.) between entity vectors with reasonable computational costs [23,24,67–69]. TransE [128] is a representative of such models, that requires minimal parameterisation and achieves very good performance. It assumes that the relationships in KGs are hierarchical and uses translations to represent them, where a single low-dimensional vector is employed to represent each targeted relationship. Its intuitive, highly scalable and effective design has driven the development of a number of translation-based algorithms [129–132], of which main benefits include constraining the translations within the relation-specific space and incorporating extra information (i.e., relation paths over the knowledge graphs) into the translation-based energy function.

In this work, we focus on further improving translation-based relation modelling. Our key idea is that more complex link representations could be constructed to reflect more accurately the different roles of each relation type. This is fundamentally different from the assumption made in most existing works, that only distinguish the link representations among different relation types. In real-world applications, the entity can always have exactly one meaning facilitated by the KG construction stage. However, links can be more complex and they usually correlate with each other, which makes them much harder to analyse. Therefore, a more careful design is required to model link representations. Here, we show an example that the same link can possess different characteristics when being involved with different entity pairs, by considering the typical hierarchical link of “descendantOf”. For instance, if both facts of *(person_A, descendantOf, person_B)* and *(person_B, descendantOf, person_C)* are true, *(person_A, descendantOf, person_C)* must also be true according to the hierarchical property of the "descendantOf" relationship, although it takes a longer range of dependencies than the former two triplets. It is obvious that “descendantOf" has as a direct link role with *(person_A, person_B)* and an indirect role with *(person_A, person_C)*. Existing works do not explicitly consider the different roles of the same link in different entity pairs. We propose a new translation strategy to address this, which although maps the entities and links within the same unified vector space, it models the multiple facets of each link by projecting the link vector on the relevant entity pair space to create more flexible interactions. The proposed algorithm is referred to as Translating on Pairwise Entity Space (TransPES). It is trained on a ranking based objective function using stochastic gradient descent, and is compared with multiple state-of-the-art methods in the field, using two commonly used benchmark datasets on link prediction. To facilitate a deeper analysis of the link prediction behaviour, we also propose a new way for partitioning the testing relational triplets to demonstrate how the algorithm behaves on different arrangements of test data.

The remainder of this chapter is organised as follows. In Section 4.2 and 4.3, a review of the previous works is provided for multi-relational learning along with their model design and limitations. The mathematical formulation of our model and the associated analysis are presented in Section 4.4. Related experiments and evaluations are conducted in Section 4.5. The work is concluded along with future directions in Section 4.6.

4.2 A Brief Review

Early works on modelling multi-relational data employ graphical models, such as Bayesian clustering frameworks [20, 121–124] or Markov logic networks [125–127]. Most of these models cannot be applied to analyse large-scale relational databases due to their high cost of inference. Another line of work treats the multi-relational data as 3-dimensional adjacency tensors, and applies tensor factorisation techniques [65–67] to analyse its link structure. One representative work is RESCAL [67], which models entities as latent feature vectors and relation types as matrices containing pairwise interaction weights between entities, and optimises efficiently the model variables via alternating least squares. It achieves state-of-the-art accuracies for many benchmark datasets, and has been applied for link prediction on entire KGs, such as YAGO and DBpedia [111, 112].

Although the size of the adjacency tensor for modelling KGs can be very large, only a small fraction among all possible relations between entities are likely to be correct. For example, there are over 450,000 actors and over 250,000 movies stored in Freebase [113], but each actor stars only in a few movies [31]. To efficiently deal with the sparse relationships in KGs, Structured Embedding (SE) model [23] introduces a powerful ranking loss for learning entity embeddings. This stimulates the development of a group of neural network models, such as Latent Factor Model (LFM) [68], neural tensor networks [69], and Semantic Matching Energy (SME) models [24], which design respective score functions to fit the likely true relations utilising different operations between the latent entity representations. These models seem to be appropriate as they attempt to model any kind of interactions through universal numerical operations. However, they are computationally expensive and are likely to suffer from overfitting with regard to very sparse relations, and this fails to capture intrinsic properties of the relations leading to weak model interpretability.

It has been shown in [133] that the word vectors learned from free text, coincidentally represent some hierarchical relationships as translations between word vectors; e.g., vec(“Germany") + vec(“capital") is closest (translated) to vec(“Berlin"). This motivates the first translation-based (or called distant) model TransE [128], which is light on parameterisation, but outperforms all former methods in link prediction on very large KGs. The appealing performance and scalability of this simple model have inspired the development of many others [129–132] that build upon the translation operations. Specifically, TransH [130] and TransR [131] assume that there is a link space for each relation type and project the entity embeddings to each link space before translation. They have shown consistent and significant improvements compared to TransE on some very large KGs. A thorough survey on relational learning techniques for analysing KGs can be found in [70].

4.3 Previous Methods

A knowledge graph $\mathcal{D}$ consists of a set of links between a fixed set of entities. Let $\mathcal{E} = \{e_1, \ldots, e_{N_e}\}$ denotes the entity set and $\mathcal{R} = \{r_1, \ldots, r_{N_r}\}$ the link set. Relation information indicated by $\mathcal{D}$ can be converted to relation triplets such as (e_h, r_ℓ, e_t), where $e_h, e_t \in \mathcal{E}$ are referred as the head and tail, respectively, and $r_\ell \in \mathcal{R}$ the link (or relation type). For example, (*Champa*, *formOfgoverment*, *Monarchy*) is one of such relation triplets, where the head entity “*Champa*" and the tail entity

“*Monarchy*" is linked by the relation type “*formOfgoverment*". For convenience, we denote the relation triplet (e_h, r_ℓ, e_t) as (h, ℓ, t) by referring only to the indices of the entities and links. Given a set of known links within $\mathcal{D}$, the goal is to infer unknown links and correct known but mistaken links in order to complete $\mathcal{D}$. One way to solve this task is to learn an energy function $E(h, \ell, t)$ on the set of all possible triplets in $\mathcal{E} \times \mathcal{R} \times \mathcal{E}$, so that a triplet representing a true existing link between two entities is assembled with a low energy, otherwise with a high energy. By default, most methods represent the head entity and tail entity as vectors of $\boldsymbol{e}_h$ and $\boldsymbol{e}_t$, respectively. We will use vectors $\boldsymbol{e}_h$ and $\boldsymbol{e}_t$ as their entity representations repeatedly throughout this chapter.

In the later sections, we first summarise the non-translation models which represent the early approaches in this field. Then, we demonstrate some existing translation models, e.g., their mathematical formulations, model limitations, to derive the motivations of our model. Although different training strategies can be used, all these models mainly differ in their design of the energy function $E(h, \ell, t)$ to fit the existing triplets. We also depict these embedding models in Table 4.1, i.e., their energy function $\mathrm{E}(h, \ell, t)$ and the model complexity (the number of parameters).

4.3.1 Non-Translation Models

RESCAL [67] explains triples via pairwise interactions of the latent features in the entity vectors, its energy function is written in bilinear form as

$$E(h, \ell, t) = -\boldsymbol{e}_h^\top \mathbf{W}_\ell \boldsymbol{e}_t, \ \mathbf{W}_\ell \in \mathbb{R}^{k \times k}. \tag{4.3.1}$$

$\mathbf{W}_\ell$ is a weight matrix for link ℓ, its entries w_{ij} specify how much the latent features i and j interact in the ℓ-th relation.

SE [23] transforms the head and tail entity vectors into an embedding space by the head and tail relation matrices before comparing their distances

$$E(h, \ell, t) = -\|\mathbf{W}_{\ell h} \boldsymbol{e}_h - \mathbf{W}_{\ell t} \boldsymbol{e}_t\|, \ \mathbf{W}_{\ell h}, \mathbf{W}_{\ell t} \in \mathbb{R}^{k \times k}. \tag{4.3.2}$$

These head and tail relation matrices aim to define different relation-specific measures between entities.

Table 4.1: Different embedding models with the associated energy function and parameters specified. Both k and d are the dimensions of embedding space, we assume k to be the final reduction dimensionality ($k < d$ in general). $\bar{\times}_3$ denotes the n-mode vector-tensor product along the 3rd mode.

Model	Energy function $\mathrm{E}(h, \ell, t)$	#Parameters
Unstructured [24]	$\\|\boldsymbol{e}_h - \boldsymbol{e}_t\\|$	$\mathcal{O}(N_e k)$
RESCAL [67]	$-\boldsymbol{e}_h^\top \mathbf{W}_\ell \boldsymbol{e}_t,\ \mathbf{W}_\ell \in \mathbb{R}^{k\times k}$	$\mathcal{O}(N_e k + N_r k^2)$
SE [23]	$-\\|\mathbf{W}_{\ell h}\boldsymbol{e}_h - \mathbf{W}_{\ell t}\boldsymbol{e}_t\\|,\ \mathbf{W}_{\ell h}, \mathbf{W}_{\ell t} \in \mathbb{R}^{k\times k}$	$\mathcal{O}(N_e k + 2N_r k^2)$
SME(linear) [24]	$-(\mathbf{W}_{h1}\boldsymbol{e}_h^\top + \mathbf{W}_{h2}\boldsymbol{e}_\ell^\top + \boldsymbol{b}_h^\top)^\top(\mathbf{W}_{t1}\boldsymbol{e}_t^\top + \mathbf{W}_{t2}\boldsymbol{e}_\ell^\top + \boldsymbol{b}_t^\top)$ $\mathbf{W}_{h1}, \mathbf{W}_{h2}, \mathbf{W}_{t1}, \mathbf{W}_{t_2} \in \mathbb{R}^{k\times d},\ \boldsymbol{b}_h, \boldsymbol{b}_t \in \mathbb{R}^k$	$\mathcal{O}(N_e d + N_r d + 4kd)$
SME(bilinear) [24]	$-((\mathbf{W}_h \bar{\times}_3 \boldsymbol{e}_\ell^\top)\boldsymbol{e}_h^\top + \boldsymbol{b}_h^\top)((\mathbf{W}_t \bar{\times}_3 \boldsymbol{e}_\ell^\top)\boldsymbol{e}_t^\top + \boldsymbol{b}_t^\top)$ $\mathbf{W}_h, \mathbf{W}_t \in \mathbb{R}^{k\times d\times d},\ \boldsymbol{b}_h, \boldsymbol{b}_t \in \mathbb{R}^k$	$\mathcal{O}(N_e d + N_r d + 2kd^2)$
LFM [68]	$-\boldsymbol{e}_h^\top(\sum_{j=1}^s \alpha_j^\ell \boldsymbol{u}_j \boldsymbol{v}_j^\top)\boldsymbol{e}_t$ $\boldsymbol{u}_j, \boldsymbol{v}_j \in \mathbb{R}^k,\ s < k$	$\mathcal{O}(N_e k + 2N_r sk)$
TransE [128]	$\\|\boldsymbol{e}_h + \boldsymbol{r}_\ell - \boldsymbol{e}_t\\|$	$\mathcal{O}(N_e k + N_r k)$
TransH [130]	$\\|(\boldsymbol{e}_h - \boldsymbol{w}_\ell^\top \boldsymbol{e}_h \boldsymbol{w}_\ell) + \boldsymbol{r}_\ell - (\boldsymbol{e}_t - \boldsymbol{w}_\ell^\top \boldsymbol{e}_t \boldsymbol{w}_\ell)\\|$ $\boldsymbol{w}_\ell \in \mathbb{R}^k$	$\mathcal{O}(N_e k + 2N_r k)$
TransR [131]	$\\|\mathbf{P}_\ell \boldsymbol{e}_h + \boldsymbol{r}_\ell - \mathbf{P}_\ell \boldsymbol{e}_t\\|$ $\mathbf{P}_\ell \in \mathbb{R}^{k\times d}$	$\mathcal{O}(N_e k + N_r k + N_r kd)$
TransPES	$\\|\boldsymbol{e}_h + \mathbf{P}_{ht}\boldsymbol{r}_\ell - \boldsymbol{e}_t\\|$	$\mathcal{O}(N_e k + N_r k)$

SME [24] proposes a neural network framework for encoding the energy function, under which two models have been developed. One defines the energy function as

$$E(h,\ell,t) = -(\mathbf{W}_{h1}\boldsymbol{e}_h^\top + \mathbf{W}_{h2}\boldsymbol{e}_\ell^\top + \boldsymbol{b}_h^\top)^\top(\mathbf{W}_{t1}\boldsymbol{e}_t^\top + \mathbf{W}_{t2}\boldsymbol{e}_\ell^\top + \boldsymbol{b}_t^\top), \tag{4.3.3}$$

where $\mathbf{W}_{h1}, \mathbf{W}_{h2}, \mathbf{W}_{t1}, \mathbf{W}_{t_2} \in \mathbb{R}^{k\times d}$ (weights), $\boldsymbol{b}_h, \boldsymbol{b}_t \in \mathbb{R}^k$ (biases). This model is denoted as SME (linear) since the terms in the braces are in linear form.

The second model uses 3-modes tensors as core weights

$$E(h,\ell,t) = -((\mathbf{W}_h\bar{\times}_3\boldsymbol{e}_\ell^\top)\boldsymbol{e}_h^\top + \boldsymbol{b}_h^\top)((\mathbf{W}_t\bar{\times}_3\boldsymbol{e}_\ell^\top)\boldsymbol{e}_t^\top + \boldsymbol{b}_t^\top), \tag{4.3.4}$$

where $\mathbf{W}_h, \mathbf{W}_t \in \mathbb{R}^{k\times d\times d}$, $\boldsymbol{b}_h, \boldsymbol{b}_t \in \mathbb{R}^k$. $\bar{\times}_3$ denotes the n-mode vector tensor product along the 3rd mode. Correspondingly, this model is denoted as SME (bilinear).

The advantage of SME models over the previous ones is that every relation is denoted by a vector $\boldsymbol{r}_\ell$ instead of matrices, which can reduce a significant number of parameters.

LFM [68] uses a similar energy function as RESCAL, its energy function is equivalent as Eq. (4.3.1), but it requires all relation matrices $\mathbf{W}_\ell$ decompose over a common set of rank one matrices

$$\mathbf{W}_\ell = \sum_{j=1}^{s} \alpha_j^\ell \boldsymbol{u}_j \boldsymbol{v}_j^\top, \text{ for } \boldsymbol{u}_j, \boldsymbol{v}_j \in \mathbb{R}^k. \tag{4.3.5}$$

This decreases the overall number of parameters compared to RESCAL, and enables a more efficient optimisation procedure.

In summary, these models consist of a large number of parameters and have only been applied to process small KGs. Though they have continually progressed towards lighter parameterisation, their performances do not degrade much.

4.3.2 Translation Methods

Given the effectiveness, efficiency and plausible interpretability of the translation based relational learning technique, we aim to model the KG information more accurately by addressing limitations of these models. The most commonly used translation model TransE [128] employs the following energy function

$$\mathrm{E}(h,\ell,t) = \|\boldsymbol{e}_h + \boldsymbol{r}_\ell - \boldsymbol{e}_t\|, \tag{4.3.6}$$

where $\|\cdot\|$ denotes a norm of the input vector, e.g., the Euclidean norm, and $\boldsymbol{e}_h, \boldsymbol{r}_\ell, \boldsymbol{e}_t$ are the embedding vectors of head entity, relation type and the tail entity, respectively, distributed in the same representation space. A correct relation triplet (h, ℓ, t) possesses a low energy value while an incorrect one high. This means that, in the ideal case, $\boldsymbol{e}_t$ should be the nearest neighbour of the vector $\boldsymbol{e}_h + \boldsymbol{r}_\ell$ for a true triplet (h, ℓ, t), or should be far away from $\boldsymbol{e}_t$ for an incorrect triplet. This assumption posed by Eq. (4.3.6) can be oversimplified when processing one-to-many links. These are defined as links ℓ contained in many correct triplets (h, ℓ, t_1), (h, ℓ, t_2), $\ldots$, (h, ℓ, t_n). One example, is the “isa" link extracted from the sentence “Bob Dylan was a song writer, singer, performer, book author and film actor", based on which the following list of relation triplets can be generated

head	*link*	*tail*
(*BobDylan*,	*isa*,	*SongWriter*),
(*BobDylan*,	*isa*,	*Singer*),
(*BobDylan*,	*isa*,	*Performer*),
(*BobDylan*,	*isa*,	*BookAuthor*),
(*BobDylan*,	*isa*,	*FilmActor*).

For this type of links, TransE will return equal embeddings $\boldsymbol{e}_{t_1} = \boldsymbol{e}_{t_2} = \ldots = \boldsymbol{e}_{t_n}$ in the ideal case of zero error. Such an output fails to distinguish between different tail entities. Similarly, it can also fail to distinguish different links that are valid for the same entity pair; for instance, equal embeddings will be returned for the two different links of ”presidentOf” and “placeOfbirth” to represent the two triplets of *(Obama, presidentOf, USA)* and *(Obama, placeOfbirth, USA)* in the ideal zero error case.

To overcome this shortcoming, various modifications of the above energy function have been proposed. For instance, TransM [129] allows more flexibility to model the one-to-many links by introducing a link-specific weight w_ℓ, with which the modified energy function is defined as

$$\mathrm{E}(h, \ell, t) = w_\ell \|\boldsymbol{e}_h + \boldsymbol{r}_\ell - \boldsymbol{e}_t\|. \tag{4.3.7}$$

It imposes smaller weights to one-to-many links to prevent zero error cases, so that their associated many-side entity embeddings (i.e., $\boldsymbol{e}_{t_1}, \boldsymbol{e}_{t_2}, \ldots, \boldsymbol{e}_{t_n}$ for the one-to-many link l) could possess different representations. Another modification is TransH [130], which assumes that an entity should be assigned to different representations when being involved with different links. The entity embeddings $\boldsymbol{e}_h$ and $\boldsymbol{e}_t$ are first projected to the hyperplane of the link ℓ, denoted as $\boldsymbol{e}_h^{\perp\ell}$ and $\boldsymbol{e}_t^{\perp\ell}$, based on which the energy function is formulated as

$$\mathrm{E}(h, \ell, t) = \|\boldsymbol{e}_h^{\perp\ell} + \boldsymbol{r}_\ell - \boldsymbol{e}_t^{\perp\ell}\|_2^2. \tag{4.3.8}$$

In this case, different representations are allowed to represent the many-side embeddings for the one-to-many link l even for the zero case as long as they share the same projected representation $\boldsymbol{e}_{t_1}^{\perp\ell} = \boldsymbol{e}_{t_2}^{\perp\ell} = \cdots = \boldsymbol{e}_{t_n}^{\perp\ell}$. TransR [131] further expands this idea by allowing entities and links to be distributed in different spaces of different dimensions d and k, respectively. It introduces a set of $k \times d$ projection matrices $\{\mathbf{P}_\ell\}_{\ell=1}^{N_r}$ to align the two spaces over each link, leading to the following energy function

$$\mathrm{E}(h, \ell, t) = \|\mathbf{P}_\ell \boldsymbol{e}_h + \boldsymbol{r}_\ell - \mathbf{P}_\ell \boldsymbol{e}_t\|_2^2. \tag{4.3.9}$$

In distance calculation, both TransH and TransR employ a fixed embedding representation for each link, but parameterise an entity in different ways to reflect the role difference between links, that is, $\boldsymbol{e}^{\perp\ell}$ by TransH and $\mathbf{P}_\ell \boldsymbol{e}$ by TransR. However, it is more reasonable to fix the embedding representation for entities, but allow the opportunity to propagate relation information through entities. This is because of the true nature of a KG, where an entity has exactly one meaning or refers to exactly one thing, but links can correlate with each other. Assume there exist entities $c_1, c_2, \ldots, d_1, d_2, \ldots, e_1, e_2, \ldots$ belonging to three classes of C, D, E, and assume that the class structure can be reflected by the link information. Another advantage of characterising entities with fixed embedding representation is to show naturally the within-class closeness and between-class dispersion in the same space, so that it is possible to transfer the instance-based inference to the class-based inference, e.g., from $(c_i, r_1, d_j) \wedge (d_j, r_2, , e_k) \Rightarrow (c_i, r_3, e_k)$ to $(C, r_1, D) \wedge (D, r_2, E) \Rightarrow (C, r_3, E)$. A third advantage of representing entities with fixed embeddings but varying the link representation for different entity pairs, is that it offers the potential of addressing better the hierarchical relation structure. For example, the relation type like “descendentOf" can appear in multiple relation triplets such as *(person_A, descendentOf, person_B)* and *(person_B, descendentOf, person_C)*, based on which *(person_A, descendentOf, person_C)* can be inferred. Existing translation-based algorithms, as mentioned above, may not perform well to infer such relation, because their model expressive power can be limited by fixing the link representation of “descendentOf" regardless of which entity pairs it is involved with. Instead, by using different representations for “descendentOf", the model can become more flexible and formulate more accurately the interaction between“descendentOf" and different entity pairs of *(person_A, person_B)*, *(person_B, person_C)* and *(person_A, person_C)*.

4.4 The Proposed Method

4.4.1 Model Construction

The energy function of an input relation triplet is parameterised over not only three individual k-dimensional embedding vectors of its head, tail and link, but also a set of $k \times k$ transformation matrices $\{\mathbf{P}_{ht}\}_{h,t}$. Different matrices are constructed for different head-tail entity pairs (h, t) to create a bespoke link representation for a given entity pair. We formulate the energy function as

$$\mathrm{E}(h, \ell, t) = \|\boldsymbol{e}_h + \mathbf{P}_{ht}\boldsymbol{r}_\ell - \boldsymbol{e}_t\|_2, \tag{4.4.1}$$

where, apart from the l_2-norm, other ones or dissimilarity measures can be used.

To reduce the computational cost, instead of optimising the transformation matrices, each $\mathbf{P}_{ht}$ is computed as a matrix that projects a k-dimensional vector onto the space spanned by the two (typically independent) k-dimensional entity vectors $\boldsymbol{e}_h$ and $\boldsymbol{e}_t$. Letting the columns of the $k \times 2$ matrix $\mathbf{E}_{ht}$ be the two entity embedding vectors $\boldsymbol{e}_h$ and $\boldsymbol{e}_t$, then the projection vector of $\boldsymbol{r}_\ell$ must be the linear combination of the columns of $\mathbf{E}_{ht}$. We denote the combination coefficient vector as $\boldsymbol{x}$. Clearly, vector $(\boldsymbol{r}_\ell - \mathbf{E}_{ht}\boldsymbol{x})$ is perpendicular to the column vectors of $\mathbf{E}_{ht}$ due to the property of orthogonal projection operator. Formally, it leads to the following equation

$$\mathbf{E}_{ht}^{\top}(\boldsymbol{r}_{\ell} - \mathbf{E}_{ht}\boldsymbol{x}) = 0, \tag{4.4.2}$$

which can readily be solved by $\boldsymbol{x} = \left(\mathbf{E}_{ht}^{\top}\mathbf{E}_{ht}\right)^{-1}\mathbf{E}_{ht}^{T}\boldsymbol{r}_{\ell}$ when the columns of $\mathbf{E}_{ht}$ are linearly independent.

Correspondingly, the projection vector $\mathbf{E}_{ht}\boldsymbol{x}$ is given by

$$\mathbf{E}_{ht}\boldsymbol{x} = \mathbf{E}_{ht}\left(\mathbf{E}_{ht}^{\top}\mathbf{E}_{ht}\right)^{-1}\mathbf{E}_{ht}^{T}\boldsymbol{r}_{\ell}. \tag{4.4.3}$$

And the analytic form of the orthogonal projector $\mathbf{P}_{ht}$ is defined as

$$\mathbf{P}_{ht} = \mathbf{E}_{ht}\left(\mathbf{E}_{ht}^{T}\mathbf{E}_{ht}\right)^{-1}\mathbf{E}_{ht}^{T}. \tag{4.4.4}$$

To regularise and make the process more numerically flexible, Eq.(4.4.4) is modified according to

$$\mathbf{P}_{ht} = \mathbf{E}_{ht}\left(\mathbf{E}_{ht}^{T}\mathbf{E}_{ht} + \xi\mathbf{I}\right)^{-1}\mathbf{E}_{ht}^{T}, \tag{4.4.5}$$

where $\xi > 0$. Using Eq.(4.4.5), for sufficiently small ξ, the transformed vector $\mathbf{P}_{ht}\boldsymbol{r}_{\ell}$ lies very close to the entity subspace spanned by $\boldsymbol{e}_h$ and $\boldsymbol{e}_t$. This can be seen because

$$\begin{aligned}
&\mathbf{E}_{ht}^{T}\left(\mathbf{I} - \mathbf{P}_{ht}\right)\boldsymbol{r}_{\ell} \\
=&\mathbf{E}_{ht}^{T}\left(\boldsymbol{r}_{\ell} - \mathbf{E}_{ht}\left(\mathbf{E}_{ht}^{T}\mathbf{E}_{ht} + \xi\mathbf{I}\right)^{-1}\mathbf{E}_{ht}^{T}\boldsymbol{r}_{\ell}\right) \\
=&\mathbf{E}_{ht}^{T}\boldsymbol{r}_{\ell} - \left(\mathbf{E}_{ht}^{T}\mathbf{E}_{ht} + \xi\mathbf{I} - \xi\mathbf{I}\right)\left(\mathbf{E}_{ht}^{T}\mathbf{E}_{ht} + \xi\mathbf{I}\right)^{-1}\mathbf{E}_{ht}^{T}\boldsymbol{r}_{\ell} \\
=&\xi\left(\mathbf{E}_{ht}^{T}\mathbf{E}_{ht} + \xi\mathbf{I}\right)^{-1}\mathbf{E}_{ht}^{T}\boldsymbol{r}_{\ell},
\end{aligned} \tag{4.4.6}$$

which shows that for any $\boldsymbol{r}_{\ell}$ we have $\lim_{\xi\to 0}\mathbf{E}_{ht}^{T}\left(\mathbf{I} - \mathbf{P}_{ht}\right)\boldsymbol{r}_{\ell} = \mathbf{0}$.

In TransR, different dimensionalities for the two embedding spaces of (d) entities and (k) links are allowed, and a set of $k \times d$ transformation matrices are employed to align the two spaces over links. Differently here, we assume equal dimensionality (k) of the two spaces, and employ a set of $k \times k$ transformation matrices to align the two spaces over entity pairs. The benefit of using equal dimensionality, is that it enables to derive an analytic form of the projection matrix as in Eq.(4.4.5) without additional effort to optimise it. For TransR, when $d > k$, the information stored in an entity embedding is compressed to a lower-dimensional vector. When $d < k$, the entity embedding is expanded to a higher-dimensional vector. However, all the expanded entities are distributed within a subspace of the link space, of which the rank of the expanded entity matrix is no more than d. Also, given the fact that the number of existing links (relation types) is usually much less than the number of existing entities in a KG, it is not necessary to increase the freedom of the link space, e.g., a higher dimensionality than the entity space. Thus, setting $d \geq k$ is more reasonable than $d < k$, and $d = k$ allows the minimal information loss, which is also the adapted setting reported in the TransR work. Because of these, we enforce equal dimensionality between the two spaces, aiming at obtaining a more mathematically convenient solution for the projection matrices without sacrificing the expressive power of the model.

4.4.2 Model Training

Given a set of known links between entities, a set of valid triplets can be constructed, which is referred to as the positive triplet set and denoted by $\mathcal{D}^+$. For each positive triplet $(h, \ell, t) \in \mathcal{D}^+$, a set of corrupted triplets can be generated by replacing either its head or tail entity with a different one, as

$$\begin{aligned}\mathcal{D}^-_{h,l,t} = & \Big\{(h^{'}, \ell, t) | h^{'} \in \{1, 2, \ldots, N_e\}, (h^{'}, \ell, t) \notin \mathcal{D}^+\Big\} \cup \\ & \Big\{(h, \ell, t^{'}) | t^{'} \in \{1, 2, \ldots, N_e\}, (h, \ell, t^{'}) \notin \mathcal{D}^+\Big\}\end{aligned}$$

Minimising the energy function in Eq.(4.4.1) parameterised via the entity and link embeddings, is equivalent to the optimisation of these embedding vectors to encourage the maximum discrimination between the positive and negative triplets. To achieve this, a margin-based ranking loss is employed, given as

$$\mathcal{L}_m = \sum_{(h,\ell,t)\in\mathcal{D}^+} \sum_{(h^{'},\ell,t^{'})\in\mathcal{D}^-_{h,\ell,t}} \Big[\gamma + \mathrm{E}(h, \ell, t) - \mathrm{E}(h^{'}, \ell, t^{'})\Big]_+, \tag{4.4.7}$$

where $[x]_+ \triangleq \max(0, x)$ denotes the positive part of the input x, and $\gamma > 0$ is a user-set margin parameter.

A length constraint $\|\boldsymbol{e}_i\|_2 \leq 1$ for each entity embedding is considered to prevent the training process from trivially minimising $\mathcal{L}_m$ by arbitrarily increasing the scale of the entity embedding. This constraint can be incorporated into the cost function $\mathcal{L}_m$ as $\sum_{i=1}^{N_e} [\|\boldsymbol{e}_i\|_2^2 - 1]_+$. We also add a regularisation term for the link embedding vectors $\{\boldsymbol{r}_j\}_{j=1}^{N_r}$. This leads to the regularised cost function

$$\mathcal{L} = \mathcal{L}_m + \lambda_1 \sum_{i=1}^{N_e} \left[\|\boldsymbol{e}_i\|_2^2 - 1\right]_+ + \lambda_2 \sum_{j=1}^{N_r} \|\boldsymbol{r}_j\|_2^2, \tag{4.4.8}$$

where $\lambda_1 > 0$ is the scale control parameter and $\lambda_2 > 0$ is the regularisation parameter. Finally, the following optimisation problem is to be solved

$$\underset{\{\boldsymbol{e}_i\}_{i=1}^{N_e}, \{\boldsymbol{r}_j\}_{j=1}^{N_r}}{\operatorname{argmin}} \mathcal{L}\left(\{\boldsymbol{e}_i\}_{i=1}^{N_e}, \{\boldsymbol{r}_j\}_{j=1}^{N_r}, \theta\right), \tag{4.4.9}$$

where $\theta = \{\gamma, \xi, \lambda_1, \lambda_2, k\}$ comprises the user parameter set, that includes one margin parameter, three regularisation ones, and the embedding dimensionality.

The pseudocode for the proposed algorithm is provided in Algorithm 2. Similar to the optimisation procedure used in [128], a stochastic gradient descent approach in minibatch mode is used. All embedding vectors for entities and relations are first initialised following the random procedure in [134]. At each main iteration, a set of positive triplets for minibatch training is randomly sampled from the training set and the corresponding corrupted triplets are generated from each individual positive triplet in this set. After a minibatch, the gradient is computed and the model parameters are updated. The algorithm terminates after a fixed number of iterations.

4.4.3 Discussion

Here we conduct some further analysis and discussion of the proposed algorithm with regard to its connections to TransE and model complexity. It can be seen from Eqs.(4.3.6) and (4.4.1) that TransE formulates a true relation triplet as $\boldsymbol{r}_\ell = \boldsymbol{e}_t - \boldsymbol{e}_h$, while the proposed algorithm as $\mathbf{P}_{ht}\boldsymbol{r}_\ell = \boldsymbol{e}_t - \boldsymbol{e}_h$ to enable the modelling of more complexed relations. For instance, given three true triplets (h, ℓ, m), (m, ℓ, t) and (h, ℓ, t), a potential solution of TransE with low energy can be self-contradictory, e.g., $\boldsymbol{r}_\ell = \boldsymbol{e}_m - \boldsymbol{e}_h = \boldsymbol{e}_t - \boldsymbol{e}_m = \boldsymbol{e}_t - \boldsymbol{e}_h$ in the ideal case of zero error. By allowing different representation $\mathbf{P}_{ht}\boldsymbol{r}_\ell$ for the same link r for different entity pairs (h, t), TransPES can overcome this effect.

Algorithm 2 Pseudocode for TransPES.

Input: Training set $\mathcal{D} = \{(h, \ell, t)\}$, entity and link sets $\mathcal{E}$ and $\mathcal{R}$, user-provided parameter set $\theta = \{\gamma, \xi, \lambda_1, \lambda_2, k\}$, triplet minibatch of size b.

```
1. **Initialisation:**
    $\boldsymbol{r} \leftarrow$ uniform $(-\frac{6}{\sqrt{k}}, \frac{6}{k})$ for each $\boldsymbol{r} \in \mathcal{R}$
    $\boldsymbol{r} \leftarrow \boldsymbol{r}/\|\boldsymbol{r}\|$ for each $\boldsymbol{r} \in \mathcal{R}$
    $\boldsymbol{e} \leftarrow$ uniform $(-\frac{6}{\sqrt{k}}, \frac{6}{k})$ for each $\boldsymbol{e} \in \mathcal{E}$
    $\boldsymbol{e} \leftarrow \boldsymbol{e}/\|\boldsymbol{e}\|$ for each $\boldsymbol{e} \in \mathcal{E}$
2. **Loop:**
    $\mathcal{D}_{batch} \leftarrow$ sample from $\mathcal{D}$
    $\mathcal{T}_{batch} \leftarrow \emptyset$
    **for** $(h, l, t) \in \mathcal{D}$ **do**
        $(h^{'}, \ell, t^{'}) \leftarrow$ sample from $\mathcal{D}^{-}_{(h,l,t)}$
        $\mathcal{T}_{batch} \leftarrow \mathcal{T}_{batch} \cup \{((h, \ell, t), (h^{'}, \ell, t^{'}))\}$
    **end for**
    $\mathcal{E}_{batch} \leftarrow$ head and tail set from $\mathcal{T}_{batch}$
    $\mathcal{R}_{batch} \leftarrow$ link set from $\mathcal{T}_{batch}$
    Gradient descent update of embeddings using $\mathcal{E}_{batch}$ and $\mathcal{R}_{batch}$
   **end loop**
```

On the other hand, assume the relation ℓ adheres to some deterministic rules, e.g., (h, ℓ, t) can be inferred from (h, ℓ, m) and (m, ℓ, t). This transitivity pattern can be potentially modelled by using three planes H_1, H_2 and H_3, on which the projected embeddings $\boldsymbol{r}_{\ell_1}, \boldsymbol{r}_{\ell_2}, \boldsymbol{r}_{\ell_3}$ for link ℓ satisfy $\boldsymbol{r}_{\ell_1} + \boldsymbol{r}_{\ell_2} = \boldsymbol{r}_{\ell_3}$. This can be achieved by the proposed algorithm with the entities $\boldsymbol{e}_h, \boldsymbol{e}_m, \boldsymbol{e}_t$ pairwisely spanning these three planes, that is, a spanned space H_{hm} of $\boldsymbol{e}_h$ and $\boldsymbol{e}_m$ “close" to the plane H_1, H_{mt} “close" to H_2, and H_{ht} “close" to H_3. By “close", we mean that the angle between the two planes is small. Subsequently, the learned lower energies of triplets (h, ℓ, m) and (m, ℓ, t), will lead to the lower energy of (h, ℓ, t), because

$$\begin{aligned} \|\boldsymbol{e}_h + \mathbf{P}_{ht}\boldsymbol{r}_\ell - \boldsymbol{e}_t\| &\approx \|\boldsymbol{e}_h + \boldsymbol{r}_{\ell_3} - \boldsymbol{e}_t\| \\ &= \|\boldsymbol{e}_h + \boldsymbol{r}_{\ell_1} - \boldsymbol{e}_m + \boldsymbol{e}_m + \boldsymbol{r}_{\ell_2} - \boldsymbol{e}_t\| \\ &\leq \|\boldsymbol{e}_h + \boldsymbol{r}_{\ell_1} - \boldsymbol{e}_m\| + \|\boldsymbol{e}_m + \boldsymbol{r}_{\ell_2} - \boldsymbol{e}_t\| \\ &\approx \|\boldsymbol{e}_h + \mathbf{P}_{hm}\boldsymbol{r}_\ell - \boldsymbol{e}_m\| + \|\boldsymbol{e}_m + \mathbf{P}_{mt}\boldsymbol{r}_\ell - \boldsymbol{e}_t\|. \end{aligned} \tag{4.4.10}$$

This indicates the possibility of encoding $(h, \ell, m) + (m, \ell, t) \Rightarrow (h, \ell, t)$ into the three spanned spaces that satisfy $\mathbf{P}_{hm}\boldsymbol{r}_\ell \approx \mathbf{P}_{hm}\boldsymbol{r}_\ell + \mathbf{P}_{mt}\boldsymbol{r}_\ell$.

Let $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n$ denote the entities that appear together with link ℓ in the true relation triplets. If one only considers to reduce the energy of correct triplets in TransE, the optimal link vector $\boldsymbol{r}_\ell^*$ must be contained in the subspace spanned by the corresponding entity embeddings. Any components added to the link embedding that are not in this subspace will increase the energy of correct triplets in TransE. However, during the training based on ranking loss, the energy of incorrect triplets is also to be maximised by seeking appropriate solution for $\boldsymbol{r}_\ell$. This will inevitably drag the learned link embedding vector $\boldsymbol{r}_\ell$ away from the optimal one $\boldsymbol{r}_\ell^*$. Differently, the proposed algorithm has the potential to learn from an incorrect triplet in the complementary space of its corresponding correct one, so that its influence over the optimal link vector $\boldsymbol{r}_\ell^*$ is automatically ignored. This enables the reduction of the energy for correct triplets and the increase of the energy for incorrect ones, simultaneously. To encourage consideration of incorrect triplets, a smaller λ_2 can be used to suppress the regularisation term of $\lambda_2 \sum_{j=1}^{N_r} \|\boldsymbol{r}_j\|_2^2$ as in Eq.(4.4.8) by amplifying the effect of link embeddings.

We also compare our model with the embedding models in Table 4.1. Since the data size is usually very large, all these models are scaled linearly in the number of entities N_e and linearly in the number of relations N_r. But scalability problem is still a big concern. Taking the RESCAL algorithm as an example, the model complexity $N_e k + N_r k^2$ is quadratic in the number of embedding dimension k and it is likely to be huge even for a moderate dimension value k . Consequently, our proposed model takes only the minimum parameterisation that is also linear in the embedding dimension k. Although it needs to compute the inverse term $\left(\mathbf{E}_{ht}^T \mathbf{E}_{ht} + \xi \mathbf{I}\right)^{-1}$ in every evaluation of $\mathbf{P}_{ht}$ in the energy function $\mathrm{E}(h, \ell, t)$, it's computational efficient as $\left(\mathbf{E}_{ht}^T \mathbf{E}_{ht} + \xi \mathbf{I}\right)$ is a small matrix of 2-by-2.

4.4.4 Data Partition Scheme for Evaluation

When evaluating a link prediction task given a KG, in addition to computing an overall performance using all the test relation triplets, researchers are looking in more detail into the different types of relation triplets and analyse how a model behaves over these different triplet types. The work in [128] suggests to group the relation triplets into the four categories of: 1-to-1, 1-to-many, many-to-1 and many-to-many, according to the cardinalities of their head and tail entities. For instance, a given triplet is classified into 1-to-many if its head entity can appear together with many tail entities in other triplets, but its tail entity only appears in this given triplet.

We propose here an alternative split of the relation triplets based on human inference logic. Specifically, it is natural for human intelligence to infer the existence of a reverse form of a given relation triplet. This can be denoted as to infer (t, ℓ^{-1}, h) from (h, ℓ, t), where ℓ^{-1} denotes the inverse link of ℓ. We list three relation triplet examples in Table 4.2 that appear in Freebase [113]. In each example, an original relation triplet and its reverse version that truly exist in the database are displayed, e.g., “/base/fight/crime type/includes crimes" is reverse of “/people/cause of death/parent cause of death". Another type of relation triplet that is natural for human to infer is the reciprocal relation, for which swapping the positions of the head and tail entities does not affect the validity of the relation triplet, e.g., links such as “MarriedTo" and “AliasTo”. This can be denoted as to infer (t, ℓ, h) from a known triplet (h, ℓ, t) when

Table 4.2: Examples of reverse triplets.

		head	relation type	tail
E1	original	/m/012hw	/people/cause_of_death/parent_cause_of_death	/m/051_y
	reverse	/m/051_y	/base/fight/crime_type/includes_crimes	/m/012hw
E2	original	/m/0hkb8	/architecture/structure/architectural_style	/m/0f447
	reverse	/m/0f447	/architecture/architectural_style/examples	/m/0hkb8
E3	original	/m/0czp_	/award/award_category/category_of	/m/0g_w
	reverse	/m/0g_w	/award/award/category	/m/0czp_

the link ℓ is reciprocal. Taking out these two types of straightforward inference, the other inference requires more complex logic.

Our assumption is that, since human can easily infer the reverse and reciprocal triplet from the given original one, the link prediction model should be able to achieve the same. Thus, it is interesting to group the relation triplets to three categories of “reciprocal type", “reverse type", and “the other" that requires more complex logic to infer. We define the collection of known relation triplets for the model to learn from as the training set, and the testing triplets for performance evaluation as the test set. The following split is applied to the test set: If a testing triplet (h, ℓ, t) is reciprocal, (t, ℓ, h) should be found in the training set. If a test triplet (h, ℓ, t) belongs to the reverse type, its reverse form (t, ℓ^{-1}, h) should appear in the training set. However, it is not easy to identify the reverse relation for any given relation type due to the lack of information. So we relax the condition to that, if (h, ℓ, t) is a reverse type, $(t, *, h)$ should exist in the training set without specifying the involved link. After identifying the reciprocal and reverse triplets, the remaining ones in the test set are categorised as “the others". Individual evaluation over each category of the testing triplet provides deeper insight on the studied model.

4.5 Experiments

4.5.1 Datasets and Experimental Setup

The proposed algorithm is compared with ten state-of-the-art translation models from the literature (see Table 4.4), evaluated using two benchmark link prediction datasets of WN18 [24] and FB15k [23] extracted from the two large real-world knowledge bases of Wordnet [49] and Freebase [113], respectively. We provide below some brief description of WN18 and FB15K datasets, and show their statistics in Table 4.3.

- The *WN18* dataset contains a total of 40,943 entities, 18 relational types and 151,442 relation triples. It is extracted from the large English lexical database Wordnet, which groups words into sets of cognitive synonyms (synsets) and interlinks these synsets by means of a small number of semantic relations, such as *synonymy*, *antonymy*, *meronymy* and *hypernymy*. One example of a typical triplet is *(_range_NN_3, _hypernym, _tract_NN_1)*, which means the third meaning of the noun "range" is a hypernym of the first sense of the noun "tract".

- The *FB15k* dataset contains a total of 14,951 entities, 1345 relation types and 592,213 relation triples. It is created by adopting the frequently occurrent entities and relationships in Freebase, which is a massive online collection database consisting of general human knowledge. It organises the human knowledge data directly in the triplet form of *(head, link, tail)*. Typical triplet examples are *(Peasant Magic, album, Solipsitalism)*, *(Barack Obama, religion, Christianity)* and *(Orange France, place-founded, Paris)*.

The proposed algorithm is compared with ten state-of-the-art translation models in terms of link prediction performance. Essentially, every model is trained by optimising a score function (or an energy function in our case) to assemble the likely relation triples with higher scores (or lower energies) than the unlikely relations. This function can thus give its estimation of the likely score (or energy) for every true

Table 4.3: Statistics of datasets.

Dataset	WN18	FB15k
Relationships	18	$1,345$
Entities	$40,943$	$14,951$
Train	$141,442$	$483,142$
Valid	$5,000$	$50,000$
Test	$5,000$	$59,071$

triplet in the test set. The following evaluation metrics based on the predicted score (or energy) are used:

- *Mean rank* measures how well the predicted scores (or energies) correlate with the true triplets [23]. For each correct triplet in the test set, we first construct the corrupt triples by replacing the head entity with all the entities in the knowledge base. The scores (or energies) of these corrupted triples are first computed and then sorted in descending (or ascending for energy) order, and the rank for the correct head entity is stored. This procedure is repeated by replacing the tail entity of each correct triple with all the entities in the knowledge base to obtain the rank for each correct tail entity. The average of all these predicted ranks in the test set is used to report the performance.

- *Hits@10* is another measure of the correlation between the predicted scores (or energies) and the true triplets [23]. Following exactly the same ranking procedure as in the mean rank evaluation, hits@10 is the proportion of the correct triplets ranked within top 10 of all the evaluated ones.

Previous work [128] suggests to filter out corrupted triplets that appear to be valid ones in the given triplets (for all the training, validation and test sets), as they should not be counted as errors. We conduct this filtering procedure to calculate filtered mean rank and hits@10 performance. To distinguish performance computed with and without the filtering procedure, we refer it as *raw* without filtering and *filtered* with the filtering procedure.

The same training, validation and test splits provided by [128] are used to evaluate the proposed model. The resulting performance is compared against performances of the state-of-the-art models that are reported in the literature [128] [130] [131] using their recommended settings as stated in the papers. Parameters of TransPES were tuned using validation set based on simple grid searches. The learning rate was searched among $\{0.1, 0.01, 0.002\}$, the dimension of the entity and link embedding k among $\{20, 50, 100\}$, the regularisation parameter for scaling control λ_1 was assigned as a constant value 1, batch size B among $\{50, 100, 200\}$, the regularisation parameter λ_2 among {0.1,0.01,0.001} and the margin γ between 0 and 1 with step size of 0.1. The regularisation parameter ξ is fixed as a small positive value 10^{-8}. For both datasets, the epochs round was set as not more than 1000 times. The best model among the last 100 epochs was selected according to the mean rank and hits@10 performance of the validation set. An open-source implementation of TransPES is available from the project webpage[1]. The optimal configurations returned by the searching procedure are $k = 20, B = 100, \lambda_2 = 0.01, \gamma = 0.7$ for WN18 and $k = 100, B = 100, \lambda_2 = 0.01, \gamma = 0.4$ for FB15k.

4.5.2 Performance Comparison

Performance of the proposed and competing methods are reported in Table 4.4. The proposed TransPES provides in general better performance than the competing ones, particularly for the larger and more complex dataset FB15k containing $1,345$ relation types. Although TransR provides good performance for the smaller dataset WN18, it performs less wells for the larger dataset FB15K. In terms of optimisation complexity, TransR requires to optimise much more variables than TransE and TransPES.

We also demonstrate how the TransPES performance changes against different settings of the embedding dimensionality (k), regularisation parameter (λ_2) and margin parameter (γ) using the FB15K dataset. In each implementation, two parameters are fixed as the ones in the optimal configuration, different settings of the third parameter within the searching range are examined, for which the raw and filtered mean ranks, also the filtered hits@10 performance for both the validation and test

[1] `https://github.com/while519/TranPES.git.`

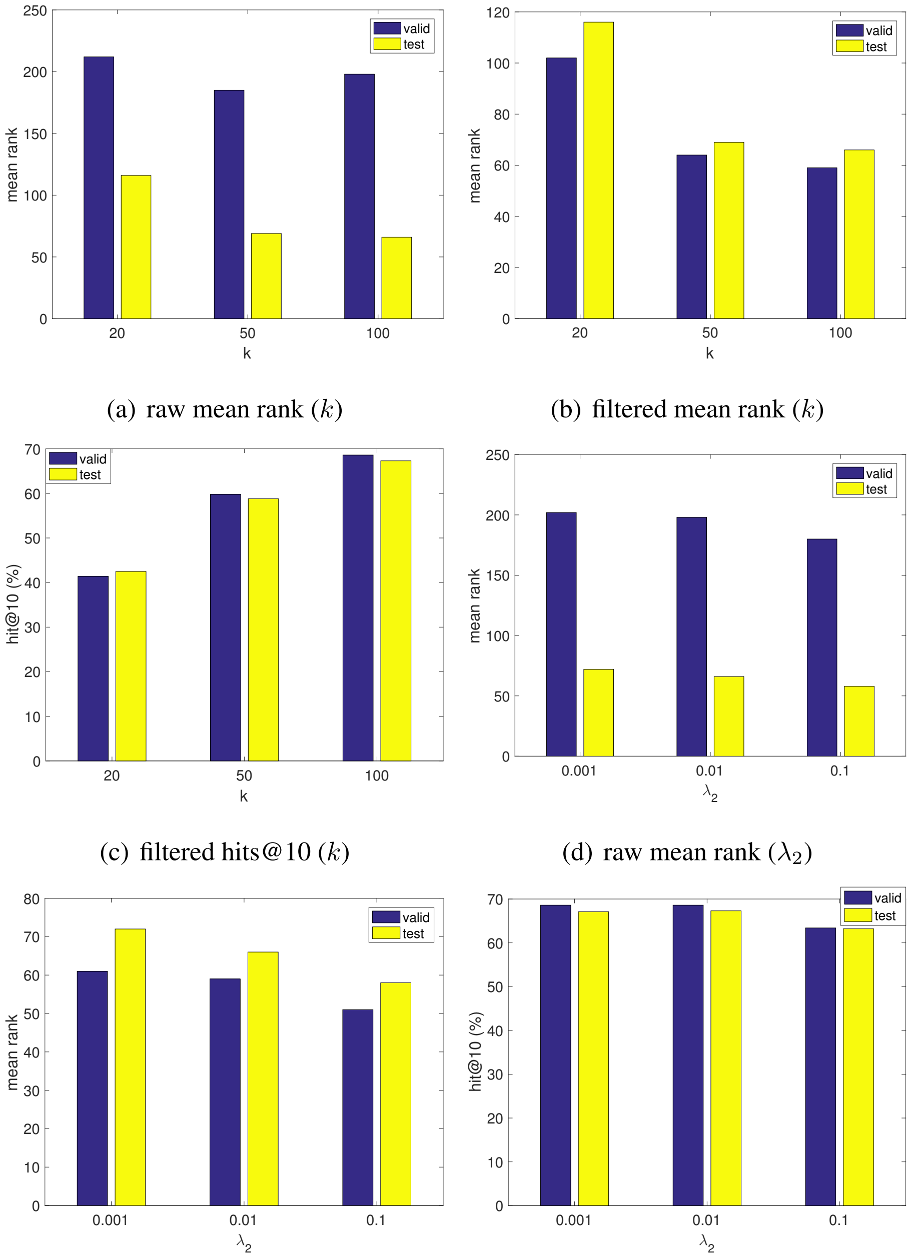
Table 4.4: Performance comparison for WN18 and FB15k datasets. The best performance is highlighted in bold, and second best underlined.

Dataset	WN18				FB15k			
Metric	Mean Rank		Hits@10(%)		Mean Rank		Hits@10(%)	
	Raw	Filter	Raw	Filtered	Raw	Filtered	Raw	Filtered
Unstructured [24]	315	304	35.3	38.2	1,074	979	4.5	6.3
RESCAL [67]	1,180	1,163	37.2	52.8	828	683	28.4	44.1
SE [23]	1,011	985	68.5	80.5	273	162	28.8	39.8
SME(linear) [24]	545	533	65.1	74.1	274	154	30.7	40.8
SME(bilinear) [24]	526	509	54.7	61.3	284	158	31.3	41.3
LFM [68]	469	456	71.4	81.6	283	164	26.0	33.1
TransE [128]	263	251	75.4	89.2	243	125	34.9	47.1
TransH [130]	318	303	75.4	86.7	$\underline{211}$	84	42.5	58.5
TransR [131]	$\underline{232}$	$\underline{219}$	$\underline{78.3}$	$\underline{91.7}$	226	$\underline{78}$	43.8	65.5
CTransR [131]	243	230	**78.9**	**92.3**	233	82	$\underline{44}$	$\underline{66.3}$
TransPES	**223**	**212**	71.6	81.3	**198**	**66**	**48.05**	**67.3**

sets are reported in Figures 4.2 and 4.2 (Cont.) . It can be seen that TransPES is less sensitive to the regularisation parameter λ_2 than to the embedding dimension k and margin parameter γ.

We further analyse the performance of the large dataset FB15K in detail using the detailed evaluation protocol suggested in [128], which classifies the hits@10 results according to four categories of relationship including 1-to-1 (1-1), 1-to-many (1-M), many-to-1 (M-1) and many-to-many (M-M).The corresponding results are shown in Table 4.5. It can be seen from the table that the proposed algorithm consistently outperforms most the competing ones, provides similarly good performance as the cluster-based TransR (CTransR). As expected, TransPES provides satisfactory performance to predict head entity in the 1-to-1, 1-to-many relationships and predict tail entity in the 1-to-1 and many-to-1 relationships.

We conduct deeper analysis for the FB15k dataset using the proposed evaluation scheme as explained in Section 4.4.4, based on which 4,336 (7.3%) reciprocal triplets and 41,481 (70.2%) reverse triplets are identified, and the remaining triplets correspond to “the others" type. Most test triplets can find their reciprocal or repetitive



(a) raw mean rank (k)

(b) filtered mean rank (k)

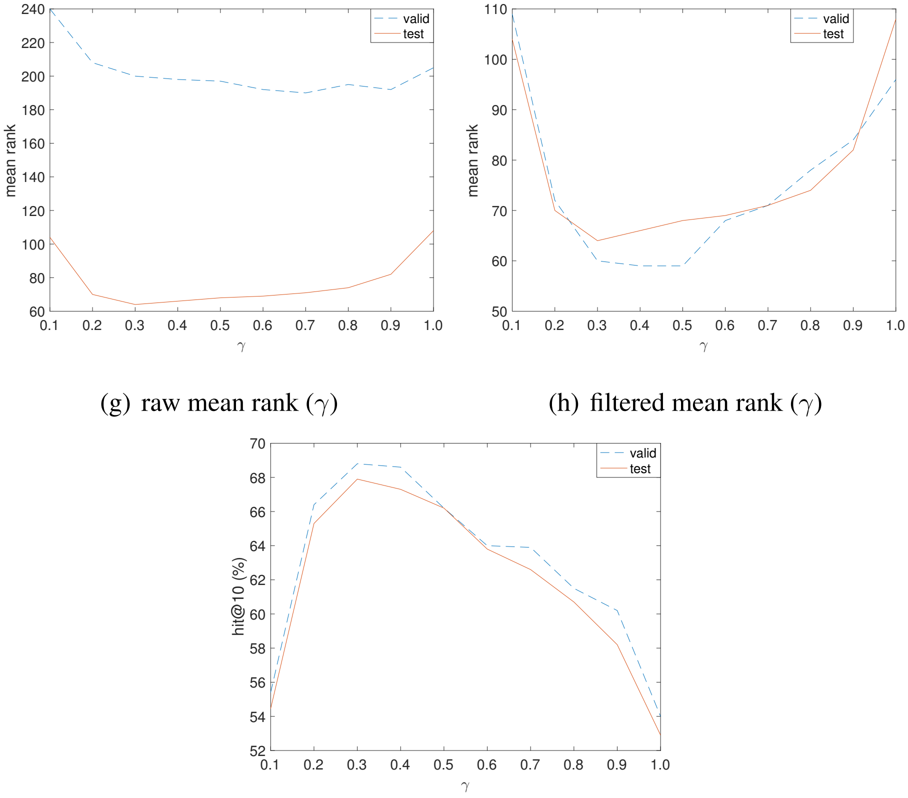
(c) filtered hits@10 (k)

(d) raw mean rank (λ_2)

(e) filtered mean rank ($\lambda 2$)

(f) filtered hits@10 (λ_2)

Figure 4.2: Illustration of the performance change of TransPES against each of its three algorithm parameters (k, λ_2, γ) in terms of the raw and filtered mean rank, also the filtered hits@10 measurements, evaluated using validation and test sets marked as “valid" and “test" respectively in each plot. *(cont.)*



(g) raw mean rank (γ)

(h) filtered mean rank (γ)

(i) filtered hits@10 (γ)

Figure 4.2 (Cont.): Illustration of the performance change of TransPES against each of its three algorithm parameters (k, λ_2, γ) in terms of the raw and filtered mean rank, also the filtered hits@10 measurements, evaluated using validation and test sets marked as “valid" and “test" respectively in each plot.

Table 4.5: Detailed evaluation on FB15k. Best performance is highlighted in bold, and second best underlined.

Tasks	Predicting Head (Hits@10)				Predicting Tail (Hits@10)			
Relation Category	1-1	1-M	M-1	M-M	1-1	1-M	M-1	M-M
Unstructured [24]	34.5	2.5	6.1	6.6	34.3	4.2	1.9	6.6
SE [23]	35.6	62.6	17.2	37.5	34.9	14.6	68.3	41.3
SME(linear) [24]	35.1	53.7	19.0	40.3	32.7	14.9	61.6	43.3
SME(bilinear) [24]	30.9	69.6	19.9	38.6	28.2	13.1	76.0	41.8
TransE [128]	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH [130]	66.7	81.7	30.2	57.4	63.7	30.1	83.2	60.8
TransR [131]	76.9	77.9	38.1	66.9	76.2	38.4	76.2	69.1
CTransR [131]	**78.6**	77.8	36.4	**68.0**	77.4	37.8	78.0	**70.3**
TranPES	78.0	**88.6**	**38.9**	67.3	**78.9**	**42.1**	**84.2**	69.8

Table 4.6: Link prediction comparison between TransE and TransPES over the reciprocal, reverse and other triplets in the test set of FB15k data.

Methods	TransE		TranPES	
Metrics	MAR	Hits@10(%)	MAR	Hits@10 (%)
Reciprocal	46	58.8	10	82.1
Reverse	75	56.9	28	72.4
Others	157	48.9	204	46.6

forms in the training set to support the inference. In Table 4.6, we compare the TransE and TransPES performance by examining how well they infer the reciprocal and reverse type of triples in the test set in Table 4.6. It can be seen from the table that the proposed algorithm achieves much better results (58.8% vs. 82.1% on the reciprocal triplets and 56.9% vs. 72.4% on the reverse ones). On the other hand, for the more challenging triplets of “the other" type, both algorithms experience a very large decrease in the performance.

4.6 Conclusion

We have presented a new translation-based relational learning algorithm to encode relation triplets in KGs using link and entity embeddings, under the constraint of employing simple operations, such as vector addition and projection to encode interlinkages in KGs and maintain very low computational cost and better model interpretability. Facing the challenge of accurately modelling complex relation logic via simple operations, the key is to unfold the relation logic by determining appropriate subspaces to work on. The proposed TransPES allows multiple representations for a single relation type to model its multimodality behaviour when interacting with different entity pairs, and employs fixed embedding representation for entities to permit smooth propagation of information across the graph. Interactions between links and entities are formulated in different spaces spanned by different entity pairs to offer bespoke link presentation for a targeted entity pair. Performance comparison with state-of-the-art methods and in-depth analysis of the algorithm behaviour based on different test data partitions demonstrate the superiority of the proposed algorithm.

In Table 4.6, we have conducted deeper analysis for the FB15k dataset using the proposed evaluation scheme, and find that the TransE and TransPES algorithms perform much better on the simple *reciprocal* and *repetitive* types of testing triples. From here we conclude that the existing relational learning algorithms could make better use of the triplets' reciprocal and repetitive forms in the training set to support inference than other dependencies. And we foresee that a better encoding and interpretability of the more complex dependency structures within the KGs is highly demanded for the future research in this field.

As we have discussed in Chapter 3, we expect that the inclusion of the relation type information can help to mitigate the difficulties of modelling the multi-modality linkage structures. However, it turns out that these sources of information themselves are noise and ambiguous by nature, and thus they can even introduce more issues. Unravelling the multi-modality properties of data will continually be an inevitable challenge for the relational learning algorithms. We will see in the next chapter, instead of labelling the relationships with type information, the linkage network combines the node description/content information to find ways to benefit the relational data learning tasks (e.g., link prediction, semantic representations).

Chapter 5

Link Prediction in Document Networks

5.1 Introduction

In the former chapters, we have developed novel embedding approaches for analysing and manipulating different types of relational structures. While traditional learning algorithms have a long history of modelling attribute/propositional data that is characterised by its high dimensional feature vectors of attributes. It is possible to combine the relational structure with the rich information of data attribute representations to improve the relational learning tasks, i.e, suggesting new and conceivable connections, clustering data into functional related sets. In this chapter, we consider joining these two types of data information into a complex network, where nodes correspond to objects with attribute representations and edges to relationships between objects. In its most simple form, we study the document network data introduced in Section 1.1.1, where it requires the attributes to be fixed and defined in the same homogeneous set, e.g., attributes might correspond to occurrence counts of terms in a fixed vocabulary when representing text or raw intensity value of each pixel when representing images. In this chapter, we discuss the document network in the context of text, that is, the nodes correspond to documents, and the edges are the citations/hyperlinks between documents.

Document network data is ubiquitous, existing throughout human generated text contents, such as citation/co-author networks in the scientific publications, friendship networks in online websites like Facebook, Flickr, and Twitter, and hyperlinked networks of webpages. Apparently, the context of the nodes along with the additional links between them should be useful for exploring various aspects of the intrinsic nature of the data. To this end, various tasks have been explored for the document network data in recent years, including improving the discovery of hidden "topic" factors in a corpus [33, 135–138], enhancing the detection of document clusters [139], inferring the topical influences over linkages [140] and inspecting the topic changes over time [141]. All these approaches model the observed linkage structure as ground truth information that has been encoded into the model (e.g., interdependencies networks between documents, linkage-based attributes) and cannot generalise to new linkages outside of the training data.

Link prediction is one fundamental task in relational learning that can potentially benefit from the fusion of both the linkage structure and the content data, which form the document network data. In the traditional network analysis, it only makes use of the observed linkage structure to help to infer new likely links. This is unfavourable since most linkage structures are constructed by human agents, which are laborious and thus far from being incomplete. Otherwise, the linkage structures are extracted by some automatic data mining techniques, causing the data to be very noisy. To support link predictions, conventional models also require a node to have a moderate number of observed links for providing a predictive distribution. But this is not likely to be the case since most linkage networks are very sparse, e.g., the average number of links for the documents in Cora citation dataset is only four. As such, one can hardly build a reliable model to generalise well based on the linkage structure alone. Differently, in the real world application, the acquisition of the content data is normally inexpensive and they can be highly useful for link prediction. For example, the text content of scientific articles which describes the themes of the papers are strongly related to the authors' selection of cited papers, the hyperlinks between webpages often signify their relevance in talking topics and items. Moreover, if no linkages are provided for a single node, we can still take advantage of this node's feature representation to give information about its linkage distribution. Thus, it is important to develop numerical models for improving the link prediction power over traditional network model by jointly utilising the linkage structure and the content data.

In this field, only a few studies are directly designed for addressing the link prediction problem. Most of them are probabilistic generative models [33, 139, 142–146]. These models assume the existence of some latent membership factors that account for both the random generation of text at each node and the links between them. Usually, the words generation in these models is as the same as that in conventional Topic models [27, 147], where each document comprises of different proportions of membership factors and each membership factor is accompanied with a distribution over the set of distinct vocabularies. But the underlying linkage structures are modelled in different ways through either a random generation process or a network regulariser. Other types of approaches have also been proposed for modelling the document networks, such as methods relying on learning a Mahalanobis distance metric [40, 148], and those built on deep learning methods [149, 150]. Some linear embedding approaches [65, 151] can also be extended to handle this link prediction task.

As we have seen in previous chapters, embedding-driven methods have been successfully applied to the scenarios for capturing important data patterns through the attribute representations as well as modelling the node's interdependencies in the linkage structure. It is then straightforward to associate each node with a latent vector representation to connect both types of modalities. Specifically, we have developed a nonlinear embedding model for simultaneously processing the linkage structure and the node attributes in the document network data. It converts the assumed-to-be exist embedding points into conditional probabilities for explaining the likelihood of pairwise linkages, and iteratively modifies the locations of these embedding points by matching these conditional probabilities with our knowledge of the link structure and the node attributes. Two objectives are used, one is a ranking based criterion for modelling the linkage structure, and the other is rest on the Kullback-Leibler divergence for capturing the patterns in the content data.

Then these two objective functions are added up with a regularisation parameter to control their relative importance, and the final cost is simply optimised by the gradient descent methods.

The rest of this chapter is organised as follows. Section 5.2 reviews some of the existing work in this field. In Section 5.3 we present our proposed algorithm with its training procedure. Section 5.4 presents the experimental results and Section 5.5 concludes this work.

5.2 Related Work

5.2.1 Pairwise Link-LDA

Pairwise Link-LDA [142] uses a Latent Dirichlet Allocation (LDA) [27] model for generating the document contents and a Mixed Membership Stochastic Block (MMSB) model [22] for modelling the linkage structure.

In LDA, each document is described as a random mixture over a small number of latent factors, where each factor is defined as a discrete distribution over the collection's vocabulary. Semantically and epistemologically, each factor can be understood as a "topic" in the document since we can browse its high probability words to sense about which something is said for each factor. Each word's creation is then derived by one of the document's topic distributions. The LDA model's generation process is depicted in lines 1-7 in Algorithm 3. Its parameters are as follows: K is the number of topics in the whole corpus, $\boldsymbol{\alpha}$ is a K-dimensional Dirichlet parameter to assign the weights for topics in each document, $\boldsymbol{\beta}_{1:K}$ represents the K topics where each topic is a discrete probability distribution over the corpus' vocabulary.

Lines 8-12 in Algorithm 3 depict the random generation process for the links. For each pair of documents, the presence or absence of a link is represented by a Bernoulli random variable whose parameter is specified by the topics involved in this interaction. Let the K-by-K matrix $\boldsymbol{\eta}$ store these parameters, with the ijth entry η_{ij} correspond to the interaction weight between the topic i and j. In Pairwise Link-LDA, the linkages are directional, and are denoted as binary variables $y_{1:D,1:D}$,

Algorithm 3 Generative process for Pairwise Link-LDA.

```
for each document $d$ do
    Draw topic proportions $\boldsymbol{\theta}_d|\boldsymbol{\alpha} \sim$ Dirichlet($\boldsymbol{\alpha}$).
    for each word $w_{d,n}$ do
        Draw a topic assignment $z_{d,n}|\boldsymbol{\theta}_d \sim$ Multinomial($\boldsymbol{\theta_d}$).
        Draw word $w_{d,n}$ from $w_{d,n}|z_{d,n}, \boldsymbol{\beta}_{1:K} \sim$ Multinomial($\boldsymbol{\beta}_{z_{d,n}}$).
    end for
end for
for each pair of documents $(d, d')$ do
    Draw an outlink topic assignment $z_{d,d'}|\boldsymbol{\theta}_d \sim$ Multinomial($\boldsymbol{\theta}_d$).
    Draw an inlink topic assignment $z_{d',d}|\boldsymbol{\theta}_{d'} \sim$ Multinomial($\boldsymbol{\theta}_{d'}$).
    Draw binary link indicator $y_{d,d'}|\boldsymbol{\eta}, z_{d,d'}, z_{d',d} \sim$ Bernoulli($\eta_{z_{d,d'},z_{d',d}}$).
end for
```

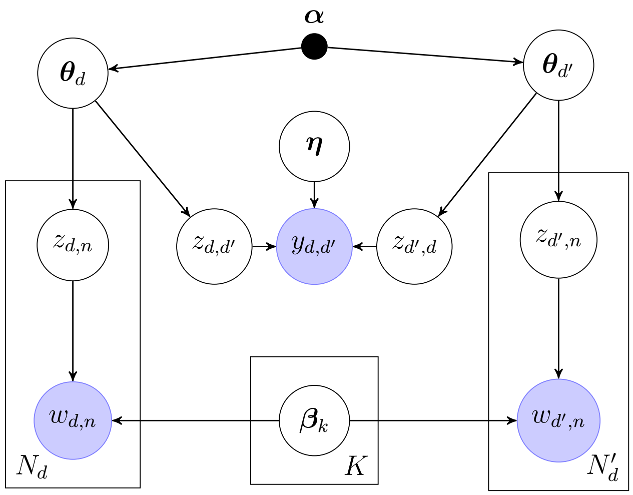


Figure 5.1: Graphical representation of the Pairwise Link-LDA model. This plate only shows the generation of one directional link $y_{d,d'}$ from document d to document d'.

where $y_{i,j} = 1$ if there is a link between the ith and jth document, otherwise $y_{i,j} = 0$. To incur a link from document d to document d', the latent topics $z_{d,d'}$ and $z_{d',d}$ are first sampled from document d and d' respectively, then the corresponding Bernoulli distribution for generating the link $y_{d,d'}$ is conditioned on its mean $\eta_{z_{d,d'},z_{d',d}}$.

The graphical representation for this random generation process for a single pair of documents is shown in Figure 5.1. Compared to previous generative approaches [135, 136], this model is capable of predicting links based on only the content of any new document. However, it does not scale well as it requires to model every pair of documents. The authors simultaneously developed a different model called Link-PLSA-LDA [142, 152], which is more efficient but can only deal with bipartite linkage structure. PMTLM [144] is also developed to address the computational issue in this model, it replaces the MMSB model with a much lighter and simpler link generation process introduced in the physics community [153].

5.2.2 Relational Topic Model

Relational Topic Model (RTM) [33] is a very important work in this field, it has made two subtle but crucial changes over the Pairwise Link-LDA model, which leads to significant computational efficiency and better predictive performance.

The generation process for RTM is denoted in Algorithm 4, note that the parameters are defined similar to the above section. In this generation process, the documents are generated as the same as in Pairwise Link-LDA. Whereas the link probability function is directly related to the latent topic assignments on the words rather than depending on the topic proportions in each document. This is more appropriate since it can avoid the issue of having disparate sets of topics to explain the words and the links. By enforcing the link probability function to be depended on the latent topic assignments $\boldsymbol{z}_d$ and $\boldsymbol{z}_{d'}$, RTM ensures the same set of topics used to generate the links are those used to generate the words. The link probability function can take different forms, such as

$$\phi_\sigma(y = 1) = \sigma\left(\boldsymbol{\eta}^\top(\bar{\boldsymbol{z}}_d \odot \bar{\boldsymbol{z}}_{d'}) + \nu\right), \tag{5.2.1}$$

or

$$\phi_e(y=1) = \exp\left(\boldsymbol{\eta}^\top(\bar{\boldsymbol{z}}_d \odot \bar{\boldsymbol{z}}_{d'}) + \nu\right), \tag{5.2.2}$$

where $\bar{\boldsymbol{z}}_d = \frac{1}{N_d}\sum_n z_{d,n}$, $\sigma(\cdot)$ is the sigmoid function ,the $\odot$ notation denotes the Hadamard product. It is parametrised by the weights in $\boldsymbol{\eta}$ and ν is the bias term. In its designs, the weight vector $\boldsymbol{\eta}$ captures the symmetric similarities between the topics of two documents, making this model only applicable for modelling undirected links.

Another important change in RTM is that it only models the observed links rather than the links between all the documents, which is fundamentally different from the previous approaches [142, 152]. This provides a significant computation advantage since its inference procedure scales linearly in the number of the links rather than the number of document pairs. Also, it is inappropriate to simply regard the unobserved links as nonlinks as the absence of a link cannot be formulated as evidence for $y_{d,d'} = 0$.

The RTM has been extended in multiple ways to model the document networks. In gRTM [145], it is generalised to allow pairwise topic interactions to model asymmetric networks. More importantly, gRTM introduces a regularisation parameter on the link structure to deal with the imbalanced network data. In [146], a nonparametric version of RTM is proposed to automatically determine the number of hidden topics. All these generative models including those described in the former section seek for probabilistic approximate inference methods, such as variational inference and (collapsed) Gibbs sampling, to approximate the posterior, estimate parameters, and make predictions.

5.2.3 Communities from Edge Structure and Node Attributes

Communities from Edge Structure and Node Attributes (CESNA) [139] is a simple probabilistic model for simultaneously modelling the network topology and the node attributes. It has a linear runtime in the network size and is capable of processing very large networks.

In CESNA, it also assumes the patterns exhibited by each document are attributed by some latent factors. Let the K dimensional vector $\boldsymbol{z}_d$ containing the nonnegative

Algorithm 4 Generative process for RTM.

```
for each document $d$ do
    Draw topic proportions $\boldsymbol{\theta}_d|\boldsymbol{\alpha} \sim$ Dirichlet($\boldsymbol{\alpha}$).
    for each word $w_{d,n}$ in d do
        Draw a topic assignment $z_{d,n}|\boldsymbol{\theta}_d \sim$ Multinomial($\boldsymbol{\theta_d}$).
        Draw word $w_{d,n}$ from $w_{d,n}|z_{d,n}, \boldsymbol{\beta}_{1:K} \sim$ Multinomial($\boldsymbol{\beta}_{z_{d,n}}$).
    end for
end for
for each pair of documents $d, d'$ do
    Draw binary link indicator
        $$y_{d,d'}|\boldsymbol{z}_d, \boldsymbol{z}_{d'} \sim \phi(\cdot|\boldsymbol{z}_d, \boldsymbol{z}_{d'}, \boldsymbol{\eta}),$$
    where $\boldsymbol{z}_d = \{z_{d,1}, \ldots, z_{d,n}\}$.
end for
```

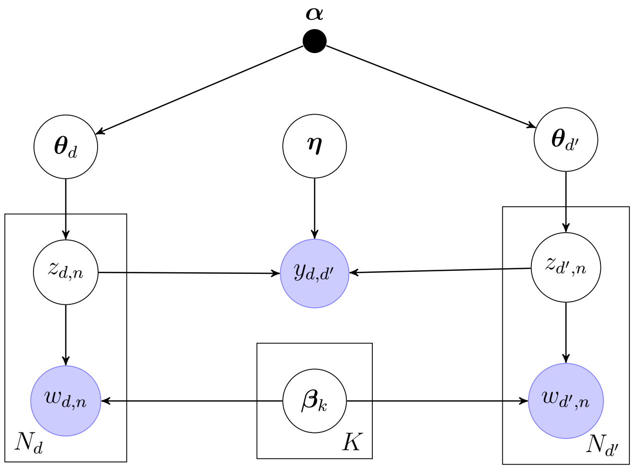


Figure 5.2: Graphical representation of the RTM model. This plate only shows the generation of one undirected link $y_{d,d'}$ between document d and document d'.

affiliation weights for the K hidden factors. For the binary linkage indicator $y_{d,d'}$, it is generated according to

$$\begin{aligned} \eta_{d,d'} &= 1 - \exp(-\boldsymbol{z}_d \odot \boldsymbol{z}_{d'}) \\ y_{d,d'} &\sim \text{Bernoulli}(\eta_{d,d'}) \end{aligned}, \tag{5.2.3}$$

where $\eta_{d,d'}$ is the Bernoulli parameter and $\odot$ is again the Hadamard product.

Similarly, the kth attribute of document d are sampled from

$$\begin{aligned} \xi_{d,k} &= 1/\left(1 + \exp(-\boldsymbol{a}_k^\top \boldsymbol{z}_d)\right) \\ w_{d,k} &\sim \text{Bernoulli}(\xi_{d,k}) \end{aligned}, \tag{5.2.4}$$

where $\xi_{d,k}$ is the Bernoulli parameter and $\boldsymbol{a}_k$ is the weight vector associated with kth attribute. Here, the attributes are assumed to be binary valued. In the case of real-valued attributes, a linear regression model is used. This model is optimised via maximising the likelihood of the observed data attributes and network linkages.

5.2.4 Structure Preserving Metric Learning

Structure Preserving Metric Learning (SPML) [148] is developed as a distance metric learning method, it learns a mapping from features such that the learned distances in the embedding space are tied to the connectivity structure of the network. In other words, the learned locations of the connected nodes in the network should be close and the unconnected nodes should be placed far.

At first, SPML learns a linear scaling on the node features by matrix $\mathbf{L} \in \mathbb{R}^{d \times d}$ that the rescaled distances between nodes i and j are expressed as

$$\begin{aligned} D_{\mathbf{L}}(\boldsymbol{v}_i, \boldsymbol{v}_j) &= \|(\mathbf{L}\boldsymbol{v}_i - \mathbf{L}\boldsymbol{v}_j)\|_2^2 \\ &= (\boldsymbol{v}_i - \boldsymbol{v}_j)^\top (\mathbf{L}^\top \mathbf{L})(\boldsymbol{v}_i - \boldsymbol{v}_j), \end{aligned} \tag{5.2.5}$$

where $\boldsymbol{v}_i$ is the attribute vector for node i. Letting $\mathbf{M} = \mathbf{L}^\top \mathbf{L}$, this rescaled distance metric is equivalently parameterised by a PSD matrix $\mathbf{M} \succeq 0$.

Then, it imposes the same linear constraints as *Structure Preserving Embedding (SPE)* [154] to enforce the network connectivity pattern to be preserved. Let $\mathbf{A} \in$

$\mathbb{R}^{n\times n}$ be the binary adjacency matrix of the document network G, one constraint could be expressed mathematically as

$$D_{\mathbf{L}}(\boldsymbol{v}_i,\boldsymbol{v}_j) \geq (1-\mathbf{A}_{ij})\max_{l}(\mathbf{A}_{il}D_{\mathbf{L}}(\boldsymbol{v}_i,\boldsymbol{v}_l)), \quad \forall i,j. \tag{5.2.6}$$

It means that the distances to all disconnected nodes must be larger than the distance to the farthest connected neighbour.

Alternatively, another constraint could be that the true graph G must have the total maximum weight among all other feasible graphs. Similar to SPE, the weight w_{ij} for each edge is chosen to equal the corresponding negated pairwise distance $-D_{\mathbf{L}}(\boldsymbol{v}_i,\boldsymbol{v}_j)$. Hence, the constraint is expressed as

$$\mathrm{Tr}(\mathbf{W}^\top\mathbf{A}) \geq \mathrm{Tr}(\mathbf{W}^\top\bar{\mathbf{A}}), \tag{5.2.7}$$

which should be valid for all adjacency matrices $\bar{\mathbf{A}}$ in a certain class of graphs, e.g., tree graph, k-regular graph.

Combining these linear constraints with a Frobenius norm regulariser on M, this learning problem is reformulated as a Semi-Definite Programming (SDP) problem and can be solved by a standard SDP solver. However, it does not scale well for very large networks. Instead, SPML adapts it to large scale networks based on a projected stochastic subgradient descent algorithm [155].

5.3 Proposed Formulation

A document network is represented as a simple directed graph $G=(\mathcal{V},\mathcal{E})$ in which we are given a set of n nodes $\mathcal{V}=\{V_1,V_2,\ldots,V_n\}$ and a set of directed edges (denoting as ordered pairs of nodes) in $\mathcal{E}$. In the convention of this paper, we consider each edge $(i,j)\in\mathcal{E}$ as indicating a directed connection from node i to node j. In document network data, the node contents are given as a feature vector of attributes, i.e., the attribute values might correspond to pixels for images or term occurrence frequencies for articles. And we denote the feature vector for node V_i as $\boldsymbol{v}_i$ of dimensionality d.

In what follows, we model each possible edge between n nodes as a random variable that indicates the model's confidence about its existence. Any true directed linkages should possess high probabilities whereas incorrect linkages low for a reasonably well-trained model. We further assume that the probability distribution of the above random variables can be effectively encoded by the hidden representations $\{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n\}$ of the n nodes that lie on a k (typically $k < d$) dimensional Euclidean space. Thus, learning such hidden representations can give us the model's confident score (the conditional probability) about each edge so as to provide a predictive distribution of links for unobserved pairs of nodes. Note that our model does not necessarily require some observed links to be present for a node to make predictions about its links as it can predict links using only a node's attributes vector. This is out of reach for the traditional network models that only utilise the link structures.

In the subsequent sections, we describe how our model is constructed and trained based on both the node content information and the observed directed links within the document network.

5.3.1 Encoding Link Validities as Stochastic Variables

We want to give locations for all the nodes in a common semantic space whereby the incurred geometry can reflect reasonably well of the link structures. Specifically, any node V_i link onto node V_j indicating that the latent representation $\boldsymbol{h}_j$ of node V_j should be put "close" to that of node V_i. By "close", we mean that the outlinked node $\boldsymbol{h}_j$ is closer to $\boldsymbol{h}_i$ than any other nodes in terms of some distance measures (we simply utilise the Euclidean distance as the model's distance metric) in this embedding space. One can readily describe the "closeness" by the conditional probabilities as proportional to either the Gaussian probability density

$$p_{j|i} = \frac{\exp(-\|\boldsymbol{h}_i - \boldsymbol{h}_j\|^2)}{\sum_{l \neq i} \exp(-\|\boldsymbol{h}_i - \boldsymbol{h}_l\|^2)} \tag{5.3.1}$$

or Cauchy distribution's probability density

$$p_{j|i} = \frac{(1 + \|\boldsymbol{h}_i - \boldsymbol{h}_j\|^2)^{-1}}{\sum_{l \neq i} (1 + \|\boldsymbol{h}_i - \boldsymbol{h}_l\|^2)^{-1}} \tag{5.3.2}$$

where $p_{j|i}$ corresponds to our confidence probability of the link from node V_i to node V_j and $p_{i|i}$ is set to zero as we do not allow any nodes to have self-connections. It should be larger for true links than for assumed-to-be false links. These two designs for encoding "closeness" into probabilities have been successfully applied to many dimensionality reduction problems [76, 156, 157].

The conditional probability $p_{j|i}$ in Eqs. (5.3.1, 5.3.2) is closely related to the notion of k-nearest neighbours – for each object it will assign greater probabilities to its closer neighbours. In fact, we choose the heavy-tailed Cauchy distribution over the Gaussian since it has been shown to capable of creating a single map that retains much of the local structure of the data while revealing global structure such as clusters at several scales [157]. Furthermore, it is computationally efficient under Eq. (5.3.2) than under a Gaussian in Eq. (5.3.1) because it does not involve computations of exponential, which as well result in a much stable optimisation procedure during the training phase.

5.3.2 Modelling the Linkage Structure

A simple objective function for finding a suitable placement of the hidden points $\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n$ is to maximise the sum of log-likelihood of correct links

$$\mathrm{f}(\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n) = \sum_{(i,j)\in\mathcal{E}} \log p_{j|i}. \tag{5.3.3}$$

However, this approach is intractable as this model is prone to over generalise and may have "imbalanced" issue for fitting the positive edges in the training set. In our experiment, maximising the above criterion usually leads some pairs of objects to be placed extremely close, exhibiting conditional probability values above 0.9, in contrast, a large proportion of the other positive links still possess very low probability values. Hence, it cannot generalise the testing links well.

Instead of maximising the sum of log-likelihood probabilities of the observed positive links, it is much more reasonable to try to make the probabilities larger for positive links than for the rest links. To achieve this, a margin-based ranking loss is minimised, given as

$$\mathcal{L}_{link} = \sum_{(i,j)\in\mathcal{E}} \sum_{(i,l)\in\mathcal{E}^-} \left[p_{l|i} - p_{j|i} + \gamma\right]_+, \tag{5.3.4}$$

where $[x]_+ \triangleq \max(0, x)$ is the Ramp function that denotes the positive part of the input x, and $\gamma > 0$ is a user-set margin parameter. $\mathcal{E}^-$ is the set for unobserved link structures, in which the linkages are assumed to be false in most network models.

This construction is beneficial as it does not assume that the unobserved links are necessarily false, just that they are "more negative" than the positive ones, which reflects our very incomplete knowledge about the linkage networks. Also, the conditional probability $p_{j|i}$ from node V_i to V_j will not increase exaggeratedly in this model as long as it satisfies $p_{j|i} > p_{l|i} + \gamma$. Therefore, it can alleviate the imbalanced issue mentioned above for taking the maximum log-likelihood in Eq. (5.3.3) as the criterion function.

Usually, the negative set $\mathcal{E}^-$ is very large, resulting in a huge number of terms for summing in $\mathcal{L}_{link}$. To enable a more efficient training process, we replace the loss function with a sequence of stochastic optimisation functions by randomly sampling corrupted links in set $\mathcal{E}^-$ at each iteration. In the construction of each of these sub-loss functions, we randomly sample a corrupted link for each individual positive link, resulting only $|\mathcal{E}|$ (cardinality of set $\mathcal{E}$) terms in total. This is the same strategy that we have used in the Chapter 4.4.2. Such sub-loss functions may take distinct values with quite large deviations at the early iteration. However, as the number of iterations increases, their costs are decreasing in the long run and will converge to a minimal state. Optionally, we can monitor the training process for this ranking loss through the proportion of terms that violate the Ramp function or utilising the sum of the positive linkages' probabilities in Eq. (5.3.4).

5.3.3 Modelling the Attributes Data

In most real world data, there is only a small proportion of linkages that are observed, resulting in a very incomplete linkage network. Taking the citation networks for example, a scientific paper cites the relevant paper that are only perceived by its authors, which usually contains only a tiny fraction of the relevant volumes. On the other hand, the readily available content data should be useful for understanding the linkage structure or enhancing the network by providing new likely linkages. Thus, it is necessary for us to incorporate the content data into the cost function.

Our goal is to learn a set of suitable conditional probability distributions $P_i = \{p_{j|i}, \text{ for } j \neq i\}$ that best fit the linkage structure as well as capturing the content data distribution. Based on the assumption that similar (neighbouring) objects tend to establish linkage connections, we convert the pairwise Euclidean distances between nodes into conditional probabilities that represent similarities. That is to say, nearby nodes will associate with higher conditional probabilities, whereas the conditional probabilities correspond to distant nodes will be miniature. Again, similar to Eqs. (5.3.1, 5.3.2), the conditional probability $q_{j|i}$ conditioned on the content data can be chosen either as

$$q_{j|i} = \frac{\exp(-(\|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2)/2\sigma_i^2)}{\sum_{l\neq i}\exp(-(\|\boldsymbol{v}_i - \boldsymbol{v}_l\|^2)/2\sigma_i^2)}, \tag{5.3.5}$$

or

$$q_{j|i} = \frac{(1 + \|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2/\sigma_i^2)^{-1}}{\sum_{l\neq i}(1 + \|\boldsymbol{v}_i - \boldsymbol{v}_l\|^2/\sigma_i^2)^{-1}}, \tag{5.3.6}$$

where σ_i is the variance of the Gaussian or Cauchy distribution that is centred on ith node attributes $\boldsymbol{v}_i$. And the value of σ_i is determined to give a conditional probability distribution $Q_i = \{q_{j|i}, \text{ for } j \neq i\}$ with fixed Shannon entropy [76] [157], which ensures σ_i to be small for dense region and large for sparse region (see Chapter 2.4.2 for calculation details). Again, we set the value of $q_{i|i}$ to zero since we are not interested in self-to-self connections.

The mismatch between the conditional distribution P_i and Q_i is naturally measured by the Kullback-Leibler divergence, and the cost is given by their sum

$$\mathcal{L}_{content} = \sum_i KL(Q_i\|P_i) = \sum_{i\neq j} q_{j|i}\log\frac{q_{j|i}}{p_{j|i}}. \tag{5.3.7}$$

Considering each term $q_{j|i}\log\frac{q_{j|i}}{p_{j|i}}$, there is a large cost for using small $p_{j|i}$ to match a large $q_{j|i}$. On the contrary, the cost will be moderate when we use a large $p_{j|i}$ to model a small $q_{j|i}$. Therefore, minimising this cost function essentially propagates the locality information within the content representations into the targeted latent representations.

Now that we combine the link and content cost functions by taking the sum of their mean $\bar{\mathcal{L}}_{link}$ and $\bar{\mathcal{L}}_{content}$ up to a weight controlling parameter λ

$$\mathcal{L}_{total} = \bar{\mathcal{L}}_{link} + \lambda\bar{\mathcal{L}}_{content}, \tag{5.3.8}$$

where λ controls the influence from the content data.

5.3.4 Training Procedure

The model training is simply carried out by gradient decent methods. Let d_{ij} denote the squared Euclidean distance between $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$. For the linkage loss function $\bar{\mathcal{L}}_{link}$ (or equivalently $\mathcal{L}_{link}$ in Eq. (5.3.4)), we only need to compute the gradient with respect to any conditional probability $p_{j|i}$, which is given as

$$\frac{\partial p_{j|i}}{\partial \boldsymbol{h}_t} = \begin{cases} 2(1+d_{ij})^{-1}((p_{j|i}^2 - p_{j|i})(\boldsymbol{h}_i - \boldsymbol{h}_j) + \sum_{l \neq j} p_{l|i}^2(\boldsymbol{h}_i - \boldsymbol{h}_l)), & \text{if } t = i, \\ 2(1+d_{ij})^{-1}(p_{j|i}^2 - p_{j|i})(\boldsymbol{h}_j - \boldsymbol{h}_i), & \text{if } t = j, \\ 2(1+d_{ij})^{-1}p_{t|i}^2(\boldsymbol{h}_t - \boldsymbol{h}_i), & \text{if } t \neq i, j. \end{cases} \tag{5.3.9}$$

Each of these gradient terms is easy to compute, involving only the computations of the squared Euclidean distances $\{d_{ij}, \text{for } \forall j \neq i\}$ and the conditional probabilities $\{p_{j|i}, \text{for } \forall j \neq i\}$. Differentiating with respect to the Kullback-Leibler cost function $\mathcal{L}_{content}$ yields

$$\frac{\partial \mathcal{L}_{content}}{\partial \boldsymbol{h}_t} = 2\sum_{l \neq t} \left(q_{l|t} - p_{l|t} + q_{t|l} - p_{t|l}\right)(1+d_{tl})^{-1}(\boldsymbol{h}_t - \boldsymbol{h}_l). \tag{5.3.10}$$

The gradients in Eqs. (5.3.9, 5.3.10) are stable to any configurations of $\{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n\}$ as the coefficients in these gradients are all in the range of $(0, 2)$. It is therefore robust to learning rate misspecification.

The pseudocode is given in Algorithm 5. All hidden points are randomly initialised according to a uniform distribution around zero. At each iteration, a set of corrupted linkages is randomly sampled for computing the surrogate linkage loss. The total cost is then obtained by summing up the linkage and content costs. After each iteration, the model parameters are updated according to the gradient of the combined loss function. We train the model for a fixed number of iterations.

We use a simple strategy for choosing the learning rate. At each iteration, we perform the gradient descent with adaptive learning rate for a fixed number of steps. In our experiment, the learning rate increases at the rate 1.01 if the cost successfully goes down, otherwise it decreases at rate 0.4. Since the cost function and its gradients are robust to any configurations of the hidden representations, we can even start to train at a very large learning rate. The code for the the proposed algorithm is provided on Github[1].

5.4 Experimental Analysis and Results

In this section, we present the performance of our proposed method as well as that of the comparing methods on three real world datasets, of which two are scientific papers with citation links between each other, the other is the webpages consisting of hyperlinks that link onto other webpages. These datasets [2] are processed by removing self-linkages and only preserving those links within the document set. The statistics of these processed datasets *Cora* [158], *Citeseer* [36] and *WebKB* [159] are summarised in Table 5.1. The experimentation is split into two main parts, covering different tasks of quantitative performance comparisons and sensitivity analysis with respect to various model parameters.

Table 5.1: Summary statistics for the three datasets after processing.

Data set	Number of documents	Number of links	Lexicon size
Cora	2,708	5,429	1,432
Citeseer	3,312	4,591	3,703
WebKB	877	1,516	1,703

5.4.1 Mean Rank Evaluation

To quantitatively evaluate our approach, we randomly split the data into training/testing set, then fit our model with both the training linkage pairs and the content feature vectors, the mean rank of the testing links based on the well-learned conditional probability distribution $\{P_i,\ i = 1, 2, \dots n\}$ is reported.

Two baseline methods are employed for comparing performances, of which one computes the ranks of testing linkages using the cosine similarity score based

[1] `https://github.com/while519/Margin-based_NN/tree/master`.

[2] Downloaded from URL `http://www.cs.umd.edu/~sen/lbc-proj/LBC.html`.

Algorithm 5 Pseudocode for the proposed method.

```
Input: feature vectors $\boldsymbol{h}_1, \boldsymbol{h}_2, \dots, \boldsymbol{h}_n$ for $n$ nodes, the linkages between nodes in $\mathcal{E}$, and user-provided regularisation parameter $\lambda > 0$, margin parameter $\gamma$, latent dimensionality $k$.

1. Initialisation:
    $\boldsymbol{h}_i \leftarrow$ uniform $(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each sample $i = 1, 2, \dots, n$
    compute the content conditional probability distributions $\{Q_i,\, i = 1, 2, \dots, n\}$ based on Eq. (5.3.5)
2. Loop:
    $\mathcal{T} \leftarrow \emptyset$
    for $(li, lo) \in \mathcal{E}$ do
        $(li, lo') \leftarrow$ sample from $\bar{\mathcal{E}}$, which is the complement set for $\mathcal{E}$
        $\mathcal{T} \leftarrow \mathcal{T} \cup \{(li, lo, lo')\}$
    end for
    compute the conditional distributions $\{P_i,\, i = 1, 2, \dots, n\}$ according to Eq. (5.3.2)
    calculate the mean of linkage loss
        $\bar{\mathcal{L}}_{link} = \frac{1}{|\mathcal{T}|} \sum_{(i,j,l) \in \mathcal{T}} \left[ p_{l|i} - p_{j|i} + \gamma \right]_+$
    calculate the mean of content loss
        $\bar{\mathcal{L}}_{content} = \frac{1}{n(n-1)} \sum_{i \neq j} q_{j|i} \log \frac{q_{j|i}}{p_{j|i}}$
    Gradient descent update of model parameters with respect to
        $\mathcal{L}_{total} = \bar{\mathcal{L}}_{link} + \lambda \bar{\mathcal{L}}_{content}$
  end loop
```

on the content data (it has the best linkage prediction performance among many other similarity measures in our experiment), the other is trained on the linkage loss function $\mathcal{L}_{link}$ in Eq. (5.3.4) (corresponding to setting $\lambda = 0$ in our proposed model) with available only the linkage structures. The first model that conditioned on the content data is denoted as *Content Only*, and the second model that excluding the effect of content information is tagged with *Linkage Only*.

Two schemes are used for partitioning the data into training/testing set. In Table 5.2, we randomly split the linkage network into training/testing set. In this way, most testing documents have been presented in the training network structure and we are making use of both their attributes and network connections to collectively make predictions of new linkages. On the contrary, the link prediction results in Table 5.3 are purely based on the testing document attributes since we have removed all the attendant links for testing documents in training. For these two tasks, we ask two questions: will the combined use of attributes data and content data help to improve the link prediction performance; and given only the attributes of new documents, how probable are its links.

As no validation set is provided, the parameters of our model are simply set by hand with intuitions. The latent dimensionality k is taken as 20, the regularisation parameter λ is fixed as 1.0, the margin is assigned as equal to the inverse of the data sample size and the algorithm stops if the number of iterations exceeds 2000. For the learning rate, we use a fixed learning rate of 10 in all experiments except for the Citeseer data experiment in Table 5.3, which we have searched the learning rate in the values of $\{10, 100, 1000\}$. We also compare our models for those models reported in Chapter 5.2, i.e., *SPML* , *CESNA*, and *RTM*. For the last two methods, we need to specify their number of topics, which is searched in the set of $\{10, 20, 30\}$ in our experiment.

The proposed method exhibits the best performance on these two tasks. In Table 5.2, it is the only method that can consistently improve performance over the cosine similarity measure (the *Content Only* method in the table). And in the second table, all the methods have dropped their performances but our proposed method still achieves the best performance on the all these datasets. From these two tables,

Table 5.2: Mean ranks of different algorithms for predicting links based on both the attribute representations and the existing linkages. The best performance is highlighted in bold, and second best underlined.

		Citeseer	Cora	Webkb
Content Only	Train	370	526	186
	Test	356	480	204
Link Only	Train	5	8	17
	Test	797	494	147
CESNA	Train	364	87	61
	Test	870	572	166
SPML	Train	467	806	191
	Test	520	794	195
RTM	Train	347	743	142
	Test	388	805	158
Proposed Method	Train	13	44	76
	Test	**343**	**358**	**93**

Table 5.3: Mean ranks of different algorithms for held out documents. The best performance is highlighted in bold, and second best underlined.

		Citeseer	Cora	Webkb
Content Only	Train	365	524	191
	Test	384	512	172
Link Only	Train	5	8	18
	Test	1811	1663	405
CESNA	Train	325	47	45
	Test	1688	1392	463
SPML	Train	438	770	171
	Test	618	868	189
RTM	Train	286	360	195
	Test	456	561	232
Proposed Method	Train	36	56	38
	Test	**351**	**475**	**105**

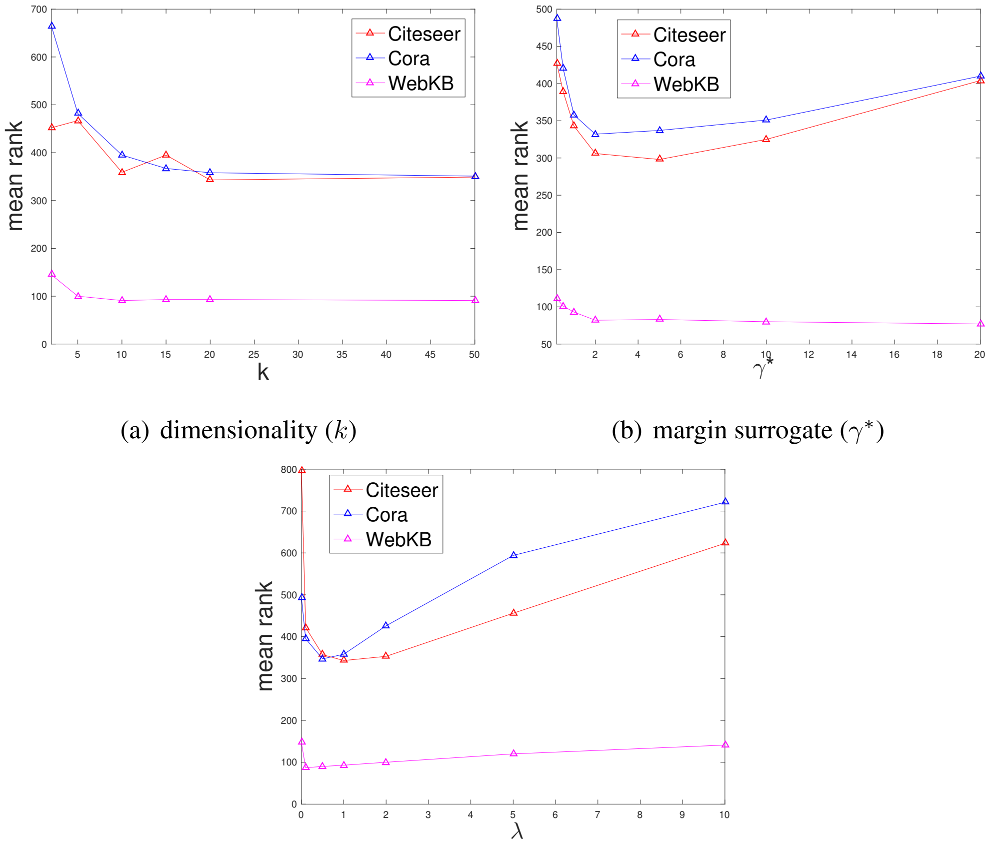
we see that though the existing methods can extract meaningful information from the attributes and the linkage structure, they could not even outperform the cosine similarity measure of *Content Only* for this link prediction task. It is interesting to see that the *Link Only* method's performance is comparable to existing methods in the first table, while it fails to make predictions for the held out documents in the second table. This reflects our data partition schemes that this link-based method can utilise the linkage structure to make predictions in the first data partition scheme but it completely has no information to predict links for held out documents in the second case. *CESNA* generally does not give good results, in the Table 5.3, it only performs slightly better than the *Link Only* method. This may be caused by its weak connections of cost functions that it learns separate parameters for the node attributes and the linkage structures. *SPML* also does not perform well due to its very simple linear assumption, but its performance is consistent in both experiments. Seen from the tables, *RTM* is only good at modelling the Citeseer dataset.

5.4.2 Sensitivity Analysis of Model Parameters

In this section, we conduct sensitivity analysis by examining the performance changes against different settings of the model parameters (k, γ, λ) using the three text datasets *Citeseer*, *Cora* and *WebKb*. The associated mean ranks of test sets are reported in Figure 5.3. Notably, because the optimal margin parameter γ is sensitive to the sample size — for datasets of larger sample size, smaller values of γ are preferred — we investigate instead the influence of the surrogate parameter $\gamma*$, where the value of γ is replaced by $\gamma^*/(\text{sample size})$ in the linkage loss function $\mathcal{L}_{link}$.

Considering the embedding dimensionality k in Figure 5.3(a), the linkage ranks decrease as k grows. This is because of the very different statistics of the high dimensional space from that of low dimensions. For instance, it is possible to place 11 points as equidistant in ten dimensions whereas by no means could one model this in a two-dimensional map. In the performance analysis in Figure 5.3(a), we see that our model is robust to high dimensionality values of k.

In Figure 5.3(b), we can see the effect of the margin parameter γ up to a scale



(a) dimensionality (k)

(b) margin surrogate (γ^*)

(c) regularisation (λ)

Figure 5.3: Illustration of the performance with respect to different settings of the embedding dimensionality k, surrogate margin parameter $\gamma * = \gamma \times$ (sample size) and the regularisation parameter λ on the three used datasets. In each figure, two parameters are fixed as the ones in Section 5.4.1, the performances correspond to the different setting of the third parameter are displayed.

factor. For small values of the margin parameter, minimising the linkage criterion $\mathcal{L}_{link}$ cannot distinguish the positive linkages from the negative ones remarkably. While for very large margin values, the Ramp function will have no effect that the minimising problem will be similar to that in Eq. (5.3.3), which cannot generalise well. The above inference is well-reflected in Figure 5.3(b) for all three datasets. The link prediction performance against to $\gamma*$ is stable in a wide range (e.g., $1 \leq \gamma^* \leq 5$).

The regularisation parameter λ controls the relative influential importance from the content data, its effect on linkage prediction performance is illustrated in Figure 5.3(c). It can be seen that our proposed model is an appropriate combination of the two source of information, as it gives better performance than either purely exploiting the linkage structure (correspond to $\lambda = 0$) or only utilising the content data (correspond to very large λ). The optimal value of λ is situated around 0.5 for all datasets.

5.5 Conclusion

In this work, we have presented a new embedding based technique for handling the link prediction problem in the document network data. The key idea of our approach is to encode both the linkage validities and the nodes neighbourhood information into embedding-based conditional probabilities. Since there are normally a huge number of pairwise links, we reduce the number of training linkages by utilising an efficient margin-based criterion, which is also capable of dealing with the imbalanced linkage structure. In the experiment of three document network datasets, we show that our proposed method improves the link prediction performance over methods using only the content data or the linkage data, and it also gives good predictive performance when making use of purely the attribute representations. Multiple performance comparisons with the state-of-art approaches also demonstrate the superiority of the proposed algorithm.

The proposed algorithm also learns a latent representation from the document network data, this data representation can be used for various unsupervised and supervised learning setups, such as clustering, classification and link predction. But the representation itself is only used to support the computations rather than give a meaningful representation of the data. Since the link prediction performance of the proposed algorithm undoubtably beat all other methods by a large margin in our experiment, it would be promising to extend this model to reveal some meaningful aspects in the data (e.g., categorising or explaining the links in a document network).

Another important challenge for this field is the quality of the datasets. For instance, in a citation network, the motivations for citations are usually very complex (e.g., relevant works, empirical findings and background readings). And the authors are usually aware of only a very small proportion of the relevant works, not to mention those future works. From this viewpoint, most linkage networks are highly noisy and sparse, and may even not be able to benefit the learning algorithms compared with those train on a single data source. In the respect, it is very important for us to identify and study a good resource of document network data.

Chapter 6

Conclusion

6.1 Summary

In this thesis, we have studied the relational data. Generally, a relational data may correspond to an arbitrarily complex database that can be defined and described by an entity-relationship model. In this context, constructing universal relational learning models/systems for processing such a complex data format would not be possible. Thus, we partition the relational data into simpler forms to reveal the significant information within them. Specifically, we have focused our attention on three different types of relational data in this thesis: **a)** bipartite weighted graph **b)** multi-relational data **c)** document network data. And we propose methods based on giving locations for each data pattern in a low dimensional space — the so-called *embedding* approaches. Then the relational structures are encoded in diverse ways, e.g., by the use of distance information, translation/projection operations and conditional probabilities. Extensive experiments on multiple datasets and comparative analysis with the state-of-art existing algorithms in each field demonstrate the superiority of our proposed algorithms. These models are also very efficient and require only a small number of parameters.

The offered good performance of our proposed algorithms comes from varied aspects: **a)** the first co-embedding generation algorithm has identified a soft version of "similarities" within the input association matrix. It assumes that only the values in the same rows or columns are comparable and can be sorted. This is distinct from some previously presented methods (e.g., ACAS, MKPE) that take the whole input values as comparable similarity quantities, and since it is based on a more reliable assumption it straightforwardly captures more reliable patterns in the data. Also, the proposed model identification score respects the same assumption, and is shown to be able to capture the co-cluster patterns in multiple synthetic data sets.**b)** Our second model for KG analysis makes use of the translation operation that is suitable for modelling the hierarchical structures in the text [133]. This assembles the model with only a few parameters — the relation is represented by a compact vector rather than large matrices/tensors. Our model then extend such translation operation to allow multiple representations for a single relation when interacting with different entity pairs . In this procedure, the model does not introduce any extra parameters and retains a better interpretability for multimodal patterns, and the advantageous translation operation is also maintained. **c)** As for the third method, it uses the pairwise margin-based ranking criterion for modelling the link structure in a document network. This criterion takes better care of the unknown linkages in a network and due to this it has been widely applied to the KG analysis problem in Chapter 4. Thus, by incorporating it into a nonlinear dimensionality reduction method, our model has offered consistently better predictive performance compared with existing algorithms.

We have already seen the powerful performance of the proposed embedding methods in various machine learning tasks, e.g., data visualisation, clustering and link prediction. It is necessary to extend embedding methods to extract meaningful and human understandable patterns. For example, in the three applications regarding the co-occurrence data, knowledge graph data and document network data, it would be good if the methods can help to explain the data linkages. And LDA is one such very successful model for explaining the documents by extracting "topic factors". We believe that such meaningful extensions can eventually benefit for human to organise the massive content of the relational data. This goal has also shaped my line of research in this thesis, where I move from modelling co-occurrence data to multi-dimensional data and move from modelling multi-dimensional data to document network data, in order to encode and explain a broader scope of the meaningful human generated big data in our relational learning models.

6.2 Future Work

There are a lot of possible directions for the future research. We briefly outline some interesting directions as follows.

The Co-occurrence Data

The proposed method in Chapter 3 takes the input as a set of association measurements between two groups of heterogeneous objects. It is shown to be capable of recovering the original shape of the synthetical data when the input association matrix reflecting well the similarity values between the two sets of heterogeneous objects. However, in the real world application, the provided associations are usually very sparse and their magnitudes have little effect on their relative strength of associations. This is a prevailing issue in many co-occurrence data sets. Taking the text corpus as an example, the words that occur most frequently may have little to no meaning on their own (e.g., the, great, we), but they have a strong co-occurrence rates in the document corpus. On the contrary, some words that are rarely found in a corpus may be strongly attached to the underlying semantics of the documents. Thus, this is an important research direction when modelling the co-occurrence data, and how to resolve it in the embedding based approaches requires much thought and creativity.

Co-Embeddings and Topic Modelling

Topic modelling [27] is an important research direction in machine learning, natural language processing and information retrieval. In developing the co-embeddings or joint embeddings of the documents and words, our hope is to identify different sets of neighbouring words as potential overlapping "topic clusters" and simultaneously put each document to be close to its topical keywords in the same dimensional space. This idea is presented in Figure 3.5 and a formal and detailed application can be found in [63]. But it is often too restricted and impractical to put the documents and words in the same space for interpreting the documents' topical associations. In the co-embedding models, an appropriate positioning of a very long document that is consist of a large number of topics is hard as it needs to be simultaneously placed close to a disparate set of "topic clusters". In turn, the requirement that different topics should be put close to some same set of documents will inevitably position these topics badly, making them intermingled rather than forming meaningful ”topic clusters”. One way to alleviate this issue is to partition each document into different paragraphs where each paragraph embodies only one or a few topics. Then the input to the co-embedding algorithms would be the associations of the words and the paragraphs. Or it is recommend to have a number of different embedding spaces to interpret the document-topic associations as well as the word-topic associations.

Graph Connectivity Patterns in Multi-dimensional Data Modelling

We have developed a novel embedding method for the multi-dimensional data in Chapter 4, it is also important to consider the use of graph connectivity patterns for predicting links in multi-dimensional data. Such models attract fewer attentions compared to the embedding-based models since they require much more computational efforts. Some theoretical work [160, 161] show that the connectivity-based approaches are often complementary to the embedding approaches, as they are concentrating on different aspects of the dependency structures. Furthermore, they are computationally efficient if some patterns or rules can be explained from only some short paths in the graph. Combining the strengths of embedding and connectivity based models is therefore a promising direction, where some efforts [160, 162–164] are continually devoted to this field.

Co-Embeddings for the Document Network

Document network embodies a linkage network between documents as well as a co-occurrence term-document matrix. Current research works [33, 137, 145] usually employ an LDA [27] model for the word generation with a regulariser based on the linkage structure. In a similar manner, we can handle the document network in a co-embedding generation setting, with the document-word Euclidean distances explaining the co-occurrence statistics and the document-document distances for explaining the linkage structure. And we can simply add them up with a weight controlling parameter to give the global cost function for parameter learning. An ideal mapping of this should comply with both the document-word associations and the document-document linkages in the data. Once the co-embeddings are computed, it could be used for various machine learning applications, e.g., clustering, classification and data visualisation.

References

[1] L. Getoor and B. Taskar, *Introduction to statistical relational learning*. MIT press, 2007.

[2] P. C. W. Davies, *The forces of nature*. CUP Archive, 1979.

[3] R. Costanza, B. G. Norton, and B. D. Haskell, *Ecosystem health: new goals for environmental management*. Island Press, 1992.

[4] S. Muggleton, R. Otero, and A. Tamaddoni-Nezhad, *Inductive logic programming*. Springer, 1992, vol. 38.

[5] K. Kersting and L. De Raedt, "1 Bayesian logic programming: theory and tool," *Statistical Relational Learning*, p. 291, 2007.

[6] S. Muggleton, "Learning stochastic logic programs," *Electron. Trans. Artif. Intell.*, vol. 4, no. B, pp. 141–153, 2000.

[7] M. Richardson and P. Domingos, "Markov logic networks," *Machine learning*, vol. 62, no. 1, pp. 107–136, 2006.

[8] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer, "Learning probabilistic relational models," in *IJCAI*, vol. 99, 1999, pp. 1300–1309.

[9] L. Getoor, N. Friedman, D. Koller, and A. Pfeffer, "Learning probabilistic relational models," in *Relational data mining*. Springer, 2001, pp. 307–335.

[10] D. Heckerman, C. Meek, and D. Koller, "Probabilistic models for relational data," Technical Report MSR-TR-2004-30, Microsoft Research, Tech. Rep., 2004.

[11] P. P.-S. Chen, "The entity-relationship model - toward a unified view of data," *ACM Transactions on Database Systems (TODS)*, vol. 1, no. 1, pp. 9–36, 1976.

[12] B. Taskar, P. Abbeel, and D. Koller, "Discriminative probabilistic models for relational data," in *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2002, pp. 485–492.

[13] J. Neville and D. Jensen, "Relational dependency networks," *Journal of Machine Learning Research*, vol. 8, no. 3, pp. 653–692, 2007.

[14] M. I. Jordan, *Learning in graphical models*. Springer Science & Business Media, 1998, vol. 89.

[15] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 467–475.

[16] M. J. Wainwright, M. I. Jordan *et al.*, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, 2008.

[17] S. Kok and P. Domingos, "Learning markov logic network structure via hypergraph lifting," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 505–512.

[18] J. Davis and P. Domingos, "Bottom-up learning of markov network structure," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 271–278.

[19] J. Van Haaren and J. Davis, "Markov network structure learning: A randomized feature generation approach." in *AAAI*, 2012, pp. 1148–1154.

[20] Z. Xu, V. Tresp, K. Yu, and H.-P. Kriegel, "Infinite hidden relational models," *CoRR*, vol. abs/1206.6864, 2006.

[21] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda, "Learning systems of concepts with an infinite relational model," in *AAAI*, vol. 3, 2006, p. 5.

[22] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *Journal of Machine Learning Research*, vol. 9, no. Sep, pp. 1981–2014, 2008.

[23] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, "Learning structured embeddings of knowledge bases," in *Conference on Artificial Intelligence*, 2011. [Online]. Available: http://infoscience.epfl.ch/record/192344/files/Bordes_AAAI_2011.pdf

[24] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "A semantic matching energy function for learning with multi-relational data," *Machine Learning*, vol. 94, no. 2, pp. 233–259, Feb. 2014. [Online]. Available: http://link.springer.com/article/10.1007/s10994-013-5363-6

[25] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[26] I. Borg and P. J. Groenen, *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

[27] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[28] 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.

[29] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE transactions on knowledge and data engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[30] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 426–434.

[31] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 11–33, 2016.

[32] Q. Lu and L. Getoor, "Link-based classification," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 496–503.

[33] J. Chang and D. M. Blei, "Hierarchical relational models for document networks," *The Annals of Applied Statistics*, pp. 124–150, 2010.

[34] A. McCallum, A. Corrada-Emmanuel, and X. Wang, "Topic and role discovery in social networks," *Computer Science Department Faculty Publication Series*, p. 3, 2005.

[35] X. Wang, N. Mohanty, and A. McCallum, "Group and topic discovery from relations and text," in *Proceedings of the 3rd international workshop on Link discovery*. ACM, 2005, pp. 28–35.

[36] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, p. 93, 2008.

[37] X. Wang and G. Sukthankar, "Multi-label relational neighbor classification using social context features," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 464–472.

[38] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, 2004, pp. 487–494.

[39] M. McPherson, L. Smith-Lovin, and J. M. Cook, “Birds of a feather: Homophily in social networks,” *Annual review of sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[40] M. Yazdani, R. Collobert, and A. Popescu-Belis, “Learning to rank on network data,” in *Mining and Learning with Graphs*, no. EPFL-CONF-192709, 2013.

[41] 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.

[42] S. Chakrabarti, B. Dom, and P. Indyk, “Enhanced hypertext categorization using hyperlinks,” in *ACM SIGMOD Record*, vol. 27, no. 2. ACM, 1998, pp. 307–318.

[43] J. Neville and D. Jensen, “Collective classification with relational dependency networks,” in *Proceedings of the Second International Workshop on Multi-Relational Data Mining*, 2003, pp. 77–91.

[44] D. Jensen, J. Neville, and B. Gallagher, “Why collective inference improves relational classification,” in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 593–598.

[45] C. Li, L. Jin, and S. Mehrotra, “Supporting efficient record linkage for large data sets using mapping techniques,” *World Wide Web*, vol. 9, no. 4, pp. 557–584, 2006.

[46] I. Bhattacharya and L. Getoor, “Entity resolution in graphs,” *Mining graph data*, p. 311, 2006.

[47] L. Otero-Cerdeira, F. J. Rodríguez-Martínez, and A. Gómez-Rodríguez, “Ontology matching: A literature review,” *Expert Systems with Applications*, vol. 42, no. 2, pp. 949–971, 2015.

[48] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, “Duplicate record detection: A survey,” *IEEE Transactions on knowledge and data engineering*, vol. 19, no. 1, pp. 1–16, 2007.

[49] G. A. Miller, “WordNet: a lexical database for English,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995. [Online]. Available: http://dl.acm.org/citation.cfm?id=219748

[50] X. Dong, A. Halevy, and J. Madhavan, “Reference reconciliation in complex information spaces,” in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. ACM, 2005, pp. 85–96.

[51] P. Singla and P. Domingos, “Entity resolution with markov logic,” in *Data Mining, 2006. ICDM’06. Sixth International Conference on*. IEEE, 2006, pp. 572–582.

[52] I. Bhattacharya and L. Getoor, “Collective entity resolution in relational data,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 5, 2007.

[53] S. E. Whang and H. Garcia-Molina, “Joint entity resolution,” in *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*. IEEE, 2012, pp. 294–305.

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

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

[56] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[57] L. Maaten, “Learning a parametric embedding by preserving local structure,” in *International Conference on Artificial Intelligence and Statistics*, 2009, pp. 384–391.

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

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

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

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

[62] 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.

[63] 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.

[64] 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.

[65] A. P. Singh and G. J. Gordon, “Relational learning via collective matrix factorization,” in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 650–658. [Online]. Available: http://dl.acm.org/citation.cfm?id=1401969

[66] T. Franz, A. Schultz, S. Sizov, and S. Staab, “Triplerank: Ranking semantic web data by tensor decomposition,” in *International semantic web conference*. Springer, 2009, pp. 213–228. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-04930-9_14

[67] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 809–816. [Online]. Available: http://machinelearning.wustl.edu/mlpapers/paper_files/ICML2011Nickel_438.pdf

[68] R. Jenatton, N. L. Roux, A. Bordes, and G. R. Obozinski, "A latent factor model for highly multi-relational data," in *Advances in Neural Information Processing Systems*, 2012, pp. 3167–3175. [Online]. Available: http://papers.nips.cc/paper/4744-a-latent-factor-model-for-highly-multi-relational-data

[69] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning With Neural Tensor Networks for Knowledge Base Completion," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 926–934. [Online]. Available: http://papers.nips.cc/paper/5028-reasoning-with-neural-tensor-networks-for-knowledge-base-completion.pdf

[70] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *Proceedings of the IEEE*, 2016.

[71] V. Ng and C. Cardie, "Improving machine learning approaches to coreference resolution," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2002, pp. 104–111.

[72] R. S. S. Prakash, D. Jurafsky, and A. Y. Ng, "Learning to merge word senses," *EMNLP-CoNLL 2007*, vol. 1005, 2007.

[73] A. Singhal, "Introducing the Knowledge Graph: things, not strings." [Online]. Available: https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html

[74] R. Qian, “Understand Your World with Bing.” [Online]. Available: http://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing/

[75] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager *et al.*, “Building watson: An overview of the deepqa project,” *AI magazine*, vol. 31, no. 3, pp. 59–79, 2010.

[76] G. E. Hinton and S. T. Roweis, “Stochastic neighbor embedding,” in *Advances in neural information processing systems*, 2003, pp. 857–864.

[77] I. T. Jolliffe, “Principal component analysis and factor analysis,” in *Principal component analysis*. Springer, 1986, pp. 115–128.

[78] E. Kokiopoulou, J. Chen, and Y. Saad, “Trace optimization and eigenproblems in dimension reduction methods,” *Numerical Linear Algebra with Applications*, vol. 18, no. 3, pp. 565–602, 2011.

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

[80] H. Hotelling, “Relations between two sets of variates,” *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[81] D. R. 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.

[82] R. Kindermann and J. L. Snell, *Markov random fields and their applications*. American Mathematical Society, 1980, vol. 1.

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

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

[85] 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.

[86] 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.

[87] 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.

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

[89] 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.

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

[91] 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.

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

[93] 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. 2536âĂŞ2556, 2007.

[94] 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.

[95] 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.

[96] 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.

[97] 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.

[98] 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.

[99] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[100] D. Bertsekas, "On the goldstein-levitin-polyak gradient projection method," *IEEE Transactions on automatic control*, vol. 21, no. 2, pp. 174–184, 1976.

[101] A. Pothen, H. D. Simon, and K.-P. Liou, "Partitioning sparse matrices with eigenvectors of graphs," *SIAM journal on matrix analysis and applications*, vol. 11, no. 3, pp. 430–452, 1990.

[102] T. F. Cox and M. A. Cox, *Multidimensional scaling*. CRC press, 2000.

[103] P. M. Yelland, “An introduction to correspondence analysis,” *Math. J*, vol. 12, pp. 1–23, 2010.

[104] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” *Journal of the American society for information science*, vol. 41, no. 6, p. 391, 1990.

[105] B. Long, Z. M. Zhang, and P. S. Yu, “Co-clustering by block value decomposition,” in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 635–640.

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

[107] 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.

[108] 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.

[109] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, “Learning entity and relation embeddings for knowledge graph completion,” in *In Proceedings of AAAIÕ15*, 2015.

[110] R. Davis and H. Shrobe, “Mit ai lab and symbolics, inc. peter szolovits mit lab for computer science,” *AI Magazine*, vol. 14, no. 1, pp. 17–33, 1993.

[111] F. M. Suchanek, G. Kasneci, and G. Weikum, “Yago: a core of semantic knowledge,” in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 697–706. [Online]. Available: http://dl.acm.org/citation.cfm?id=1242667

[112] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and others, “DBpediaâĂŞa large-scale, multilingual knowledge base extracted from Wikipedia,” *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015. [Online]. Available: http://content.iospress.com/articles/semantic-web/sw134

[113] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, “Freebase: a collaboratively created graph database for structuring human knowledge,” in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 2008, pp. 1247–1250. [Online]. Available: http://dl.acm.org/citation.cfm?id=1376746

[114] J. Betteridge, A. Carlson, S. A. Hong, E. R. Hruschka Jr, E. L. Law, T. M. Mitchell, and S. H. Wang, “Toward Never Ending Language Learning.” in *AAAI Spring Symposium: Learning by Reading and Learning to Read*, 2009. [Online]. Available: http://www.aaai.org/Papers/Symposia/Spring/2009/SS-09-07/SS09-07-001.pdf

[115] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, “Knowledge vault: A web-scale approach to probabilistic knowledge fusion,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 601–610. [Online]. Available: http://dl.acm.org/citation.cfm?id=2623623

[116] G. Weikum and M. Theobald, “From information to knowledge: harvesting entities and relationships from web sources,” in *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2010, pp. 65–76. [Online]. Available: http://dl.acm.org/citation.cfm?id=1807097

[117] R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, and D. Lin, “Knowledge base completion via search-based question answering,” in *Proceedings of the 23rd international conference on World wide web*. ACM, 2014, pp. 515–526.

[118] J. Pearl and A. Paz, *Graphoids: A graph-based logic for reasoning about relevance relations*. University of California (Los Angeles). Computer Science Department, 1985. [Online]. Available: http://ftp.cs.ucla.edu/pub/stat_ser/r53-L.pdf

[119] R. Fagin, J. Y. Halpern, Y. Moses, and M. Vardi, *Reasoning about knowledge*. MIT press, 2004. [Online]. Available: https://books.google.co.uk/books?hl=en&lr=&id=hvDuCwAAQBAJ&oi=fnd&pg=PR7&dq=reasoning+knowledge+graph&ots=ouTzwA7d5Q&sig=eSfmKml530RDK6VEdvRa98sxHP8

[120] J. F. Sowa, "Conceptual graphs," *Foundations of Artificial Intelligence*, vol. 3, pp. 213–237, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574652607030052

[121] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer, "Learning probabilistic relational models," in *IJCAI*, vol. 99, 1999, pp. 1300–1309. [Online]. Available: http://www.robotics.stanford.edu/~koller/Papers/Friedman+al:IJCAI99.pdf

[122] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda, "Learning systems of concepts with an infinite relational model," in *AAAI*, vol. 3, 2006, p. 5. [Online]. Available: http://www.aaai.org/Papers/AAAI/2006/AAAI06-061.pdf

[123] K. Miller, M. I. Jordan, and T. L. Griffiths, "Nonparametric latent feature models for link prediction," in *Advances in neural information processing systems*, 2009, pp. 1276–1284. [Online]. Available: http://papers.nips.cc/paper/3846-nonparametric-latent-feature-models-for-link-prediction

[124] I. Sutskever, J. B. Tenenbaum, and R. R. Salakhutdinov, "Modelling relational data using bayesian clustered tensor factorization," in *Advances in neural information processing systems*, 2009, pp. 1821–1828. [Online]. Available: http://papers.nips.cc/paper/3863-modelling-relational-data-using-bayesian-clustered-tensor-factorization

[125] M. Richardson and P. Domingos, “Markov logic networks,” *Machine learning*, vol. 62, no. 1-2, pp. 107–136, 2006. [Online]. Available: http://link.springer.com/article/10.1007/s10994-006-5833-1

[126] S. Kok and P. Domingos, “Statistical predicate invention,” in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 433–440. [Online]. Available: http://dl.acm.org/citation.cfm?id=1273551

[127] P. Singla and P. Domingos, “Entity resolution with markov logic,” in *Sixth International Conference on Data Mining (ICDM’06)*. IEEE, 2006, pp. 572–582. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4053083

[128] 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*, 2013, pp. 2787–2795. [Online]. Available: http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data

[129] M. Fan, Q. Zhou, E. Chang, and T. F. Zheng, “Transition-based knowledge graph embedding with relational mapping properties,” in *Proceedings of the 28th Pacific Asia Conference on Language, Information, and Computation*, 2014, pp. 328–337. [Online]. Available: http://anthology.aclweb.org/Y/Y14/Y14-1039.pdf

[130] Z. Wang, J. Zhang, J. Feng, and Z. Chen, “Knowledge graph embedding by translating on hyperplanes,” in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. Citeseer, 2014, pp. 1112–1119. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.486.2800&rep=rep1&type=pdf

[131] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, “Learning entity and relation embeddings for knowledge graph completion,” in *Proceedings of AAAI*, 2015. [Online]. Available: http://166.111.138.24/~lzy/publications/aaai2015_transr.pdf

[132] A. GarcÄśa-DurÃąn, A. Bordes, and N. Usunier, “Composing Relationships with Translations,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 286–290. [Online]. Available: http://www.aclweb.org/anthology/D15-1034.pdf

[133] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, 2013, pp. 3111–3119. [Online]. Available: http://papers.nips.cc/paper/5021-distributed-representations

[134] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *International conference on artificial intelligence and statistics*, 2010, pp. 249–256. [Online]. Available: http://machinelearning.wustl.edu/mlpapers/paper_files/AISTATS2010_GlorotB10.pdf

[135] D. A. Cohn and T. Hofmann, “The missing link-a probabilistic model of document content and hypertext connectivity,” in *Advances in neural information processing systems*, 2001, pp. 430–436.

[136] E. Erosheva, S. Fienberg, and J. Lafferty, “Mixed-membership models of scientific publications,” *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl 1, pp. 5220–5227, 2004.

[137] Q. Mei, D. Cai, D. Zhang, and C. Zhai, “Topic modeling with network regularization,” in *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 101–110.

[138] Y. Sun, J. Han, J. Gao, and Y. Yu, “itopicmodel: Information network-integrated topic modeling,” in *Data Mining, 2009. ICDM’09. Ninth IEEE International Conference on*. IEEE, 2009, pp. 493–502.

[139] J. Yang, J. McAuley, and J. Leskovec, “Community detection in networks with node attributes,” in *Data Mining (ICDM), 2013 IEEE 13th international conference on*. IEEE, 2013, pp. 1151–1156.

[140] L. Dietz, S. Bickel, and T. Scheffer, "Unsupervised prediction of citation influences," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 233–240.

[141] Q. He, B. Chen, J. Pei, B. Qiu, P. Mitra, and L. Giles, "Detecting topic evolution in scientific literature: How can citations help?" in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 957–966.

[142] R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen, "Joint latent topic models for text and citations," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 542–550.

[143] A. Gruber, M. Rosen-zvi, and Y. Weiss, "Latent topic models for hypertext," in *In UAI*. Citeseer, 2008.

[144] Y. Zhu, X. Yan, L. Getoor, and C. Moore, "Scalable text and link analysis with mixed-topic link models," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 473–481.

[145] N. Chen, J. Zhu, F. Xia, and B. Zhang, "Discriminative relational topic models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 5, pp. 973–986, 2015.

[146] J. Lu, J. Xuan, G. Zhang, Y. Da Xu, and X. Luo, "Bayesian nonparametric relational topic model through dependent gamma processes," *IEEE Transactions on Knowledge and Data Engineering*, 2016.

[147] T. Hofmann, "Probabilistic latent semantic analysis," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 289–296.

[148] B. Shaw, B. Huang, and T. Jebara, "Learning a distance metric from a network," in *Advances in Neural Information Processing Systems*, 2011, pp. 1899–1907.

[149] K. Li, J. Gao, S. Guo, N. Du, X. Li, and A. Zhang, “Lrbm: A restricted boltzmann machine based approach for representation learning on linked data,” in *Data Mining (ICDM), 2014 IEEE International Conference on*. IEEE, 2014, pp. 300–309.

[150] H. Wang, X. Shi, and D.-Y. Yeung, “Relational deep learning: A deep latent variable model for link prediction.” in *AAAI*, 2017, pp. 2688–2694.

[151] A. K. Menon and C. Elkan, “Link prediction via matrix factorization,” in *Joint european conference on machine learning and knowledge discovery in databases*. Springer, 2011, pp. 437–452.

[152] R. Nallapati and W. W. Cohen, “Link-plsa-lda: A new unsupervised model for topics and influence of blogs.” in *ICWSM*, 2008.

[153] B. Ball, B. Karrer, and M. E. Newman, “Efficient and principled method for detecting communities in networks,” *Physical Review E*, vol. 84, no. 3, p. 036103, 2011.

[154] B. Shaw and T. Jebara, “Structure preserving embedding,” in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 937–944.

[155] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, “Pegasos: Primal estimated sub-gradient solver for svm,” *Mathematical programming*, vol. 127, no. 1, pp. 3–30, 2011.

[156] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. R. Salakhutdinov, “Neighbourhood components analysis,” in *Advances in neural information processing systems*, 2005, pp. 513–520.

[157] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[158] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore, “Automating the construction of internet portals with machine learning,” *Information Retrieval*, vol. 3, no. 2, pp. 127–163, 2000.

[159] M. Craven, A. McCallum, D. PiPasquo, T. Mitchell, and D. Freitag, “Learning to extract symbolic knowledge from the world wide web,” DTIC Document, Tech. Rep., 1998.

[160] M. Nickel, X. Jiang, and V. Tresp, “Reducing the rank in relational factorization models by including observable patterns,” in *Advances in Neural Information Processing Systems*, 2014, pp. 1179–1187.

[161] K. Toutanova and D. Chen, “Observed versus latent features for knowledge base and text inference,” in *3rd Workshop on Continuous Vector Space Models and Their Compositionality*. ACL Ű Association for Computational Linguistics, July 2015. [Online]. Available: https://www.microsoft.com/en-us/research/publication/observed-versus-latent-features-for-knowledge-base-and-text-inference/

[162] X. Jiang, V. Tresp, Y. Huang, and M. Nickel, “Link prediction in multi-relational graphs using additive models,” in *Proceedings of the 2012 International Conference on Semantic Technologies Meet Recommender Systems & Big Data-Volume 919*. CEUR-WS. org, 2012, pp. 1–12.

[163] S. Riedel, L. Yao, A. McCallum, and B. M. Marlin, “Relation extraction with matrix factorization and universal schemas.” in *HLT-NAACL*, 2013, pp. 74–84.

[164] Y. Lin, Z. Liu, H.-B. Luan, M. Sun, S. Rao, and S. Liu, “Modeling relation paths for representation learning of knowledge bases,” in *EMNLP*, 2015.