DIMENSION REDUCTION FOR NETWORK ANALYSIS WITH AN APPLICATION TO DRUG DISCOVERY

by

HUIYUAN CHEN

Submitted in partial fulfillment of the requirements

for the degree of Doctor of Philosophy

Department of Computer and Data Sciences

CASE WESTERN RESERVE UNIVERSITY

January, 2021

CASE WESTERN RESERVE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

We hereby approve the dissertation of

Huiyuan Chen

candidate for the degree of **Doctor of Philosophy***.

Committee Chair

Dr. Jing Li

Committee Member

Dr. Harold S. Connamacher

Committee Member

Dr. Xusheng Xiao

Committee Member

Dr. Satya S. Sahoo

Date of Defense

March 6, 2020

*We also certify that written approval has been obtained

for any proprietary material contained therein.

Contents

List of Tables

List of Figures

Acknowledgments

First of all, I would like to express my sincere gratitude and appreciation to my advisor Dr. Jing Li for his creative thoughts and supportive comments. He is a great advisor and teaches me extensively about research, presentation, and writing skills over the five years.

I would like to thank my thesis committee members, Dr. Harold S. Connamacher, Dr. Xusheng Xiao, and Dr. Satya S. Sahoo, for going through the entire thesis and giving me critical comments. I am thankful to all of my collaborators, including Dr. Feixiong Cheng and Dr. Sudha K. Iyengar for their many valuable suggestions on my research. I would like to thank Dr. Vincenzo Liberatore for serving on my Ph.D. candidate exam committee.

Great appreciation goes to Dr. Ke Hu for his insightful suggestions and persistent support. I also feel fortunate to have the opportunity to work with my talented lab members during my entire pursuit of Ph.D. degree, including Duan Li, Zheng Wang, Qiwei Lou, Dylan Plummer, Guo Chen, Sunah Song, Yuchen Wang, Waqas Qureshi, Yige Sun, and Dennis Lin. I was so fortunate in Case Western Reserve University and will remember this amazing journey forever.

Finally and most importantly, I would like to thank my family for encouraging me to pursue my dreams. This dissertation would not have been possible without their continuous love and warm support.

Thank you all.

Dimension Reduction for Network Analysis with an Application to Drug Discovery

Abstract

by

HUIYUAN CHEN

Graphs (or networks) naturally represent valuable information for relational data, which are ubiquitous in real-world applications, such as social networks, recommender systems, and biological networks. Statistical learning or machine learning techniques for network analysis, such as random walk with restart, meta-path analysis, network embeddings, and matrix/tensor factorizations, have gained tremendous attentions recently. With rapid growth of data, networks, either homogeneous or heterogeneous, can consist of billions of nodes and edges. How can we find underlying structures within a network? How can we efficiently manage data when multiple sources describing the networks are available? How can we detect the most important relationships among nodes?

To gain insights into these problems, this dissertation investigates the principles and methodologies of dimension reduction techniques that explore the useful latent structures of one or more networks. Our dimension reduction techniques mainly leverage recent developments in linear algebra, graph theory, large-scale optimization, and deep learning. In addition, we also translate our ideas and models to several real-world applications, especially in drug repositioning, drug combinations, and drug-target-disease interactions. For each research problem, we discuss their current challenges, related work, and propose corresponding solutions.

Chapter 1

Introduction

Graphs (or networks) naturally represent valuable information for relational or linked data, which are ubiquitous in our daily life. In e-commerce (e.g. Amazon and eBay), the most prominent usage of networks is to build an user-item bipartite network and recommend new products to potential users [Li and Chen, 2009]. In bioinformatics, the networks have been used in protein-protein interactions (PPIs) prediction [Wang et al., 2013] and drug repositioning [Wang et al., 2014]. In co-authorship networks, the networks can provide good information on the patterns and structures of scientific collaborations [Sun and Han, 2012]. Finally, networks provide an alternative tool to analyze how tendencies spread across the society. For instance, some studies have shown how dynamic networks can be applied to viral marketing in order to develop a better marketing strategy [Richardson and Domingos, 2002]. Because of their prevalence, network mining has become a central topic in research community.

Many biological, social and information systems can be well described by networks, where nodes represent biological entities (e.g. protein, drug), web users, computers, images and items. An edge between two nodes is a relationship in the network. Some networks containing one single type of nodes and links can be represented as homogeneous networks, while some other networks containing abundant types of nodes and

edges, can be denoted as heterogeneous networks [Sun and Han, 2012]. The study of information entities in the networks is of great importance in real-world applications, such as social networks [Kwak et al., 2010], recommender systems [Ricci et al., 2011] and biological networks [Schwikowski et al., 2000], and so on.

For example, Figure 1.1 shows an example of biological network that contains the nodes from three domains: drug, target and disease. Within each domain, its individual network is a homogeneous in which the nodes represent the drugs (or targets and diseases) and the edges represent the interactions between nodes (e.g., drug-drug similarities). Integration of three domains is a heterogeneous network that contains lots of significant information such as the drug-target and drug-disease associations. Comparing to homogeneous networks (e.g., its components), heterogeneous networks incorporate rich semantics and more information in nodes and links and thus appear to be a common phenomenon in practice.

In addition to network representations, statistical learning or machine learning techniques for network analysis, such as random walk with restart, meta-path analysis, network embeddings and matrix/tensor factorizations, have gained a lot of attentions recently [Getoor and Diehl, 2005, Liu et al., 2018]. As the rapid of growing data, the networks, either homogeneous or heterogeneous, often consist of billions of nodes and edges. How can we find the underlying structures within network? How can we efficiently manage the data when multiple sources describing the networks are available? How can we detect the most important relationships among nodes, such as drug-disease associations or user-item interactions?

To gain insights into these problems, this dissertation investigates the principles and methodologies of dimension reduction techniques that explore the useful latent factors of one or more networks. Our dimension techniques mainly leverage techniques from linear algebra, network theory, large-scale optimization and deep learning. In addition, we also translate our ideas and models to several real-world applications,

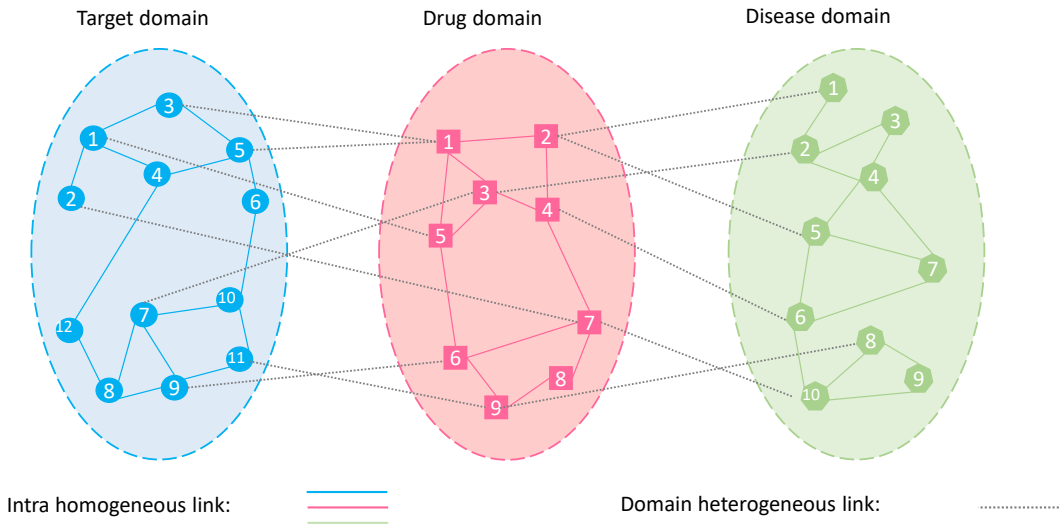

Figure 1.1: An example of a biological network which contains three types of nodes and four types of edges.

especially in drug repositioning, drug combinations and content-aware recommender system. For each research problem, we discuss their current challenges, related work, and propose corresponding solutions. We also exploit the sparsity in the data, show how to use them in large-scale system, including static and dynamic networks (e.g., social network), cross-network analytics (e.g., drug-disease network), classification, and visualization.

The dimension reduction techniques of this thesis is organized into two main parts: (i) matrix factorization, and (ii) tensor factorization. We next introduce some mathematical details about matrix and tensor algebra.

Notations. Following the convention, we denote vectors by boldface lowercase letters (e.g., $\mathbf{a}$), matrices by boldface uppercase letters (e.g., $\mathbf{A}$) and tensors by boldface caligraphic letters (e.g., $\mathcal{X}$). $\mathbf{a}_f$ denotes the f-th column of $\mathbf{A}$.

1.1 Matrix Factorization

In machine learning, encoding rectangular tables of numeric data in the form of matrices are very common, such as user-item rating matrices, user-user adjacency matrices and document-term matrices [Friedman et al., 2001]. These kinds of matrices are often analyzed using dimension reduction techniques like the Singular Value Decomposition (SVD) [De Lathauwer et al., 2000], Principal Component Analysis (PCA) [Wold et al., 1987] and its variant Nonnegative Matrix Factorization (NMF) [Lee and Seung, 2001]. Analysis of matrices is to mainly gain structural understanding of the data. Many dimension reduction techniques make use of low-rank assumption, which in effect highly reduces the dimension of data. The low-rank assumption is essentially useful since we can try to retrieve matrix entities on the basis of the low-rank latent factors. Furthermore, it requires less storage requirements, which becomes more and more important in the era of big data analysis. We next briefly introduce some backgrounds about two dimensionality reduction methods SVD and NMF since we use them frequently through the thesis.

1.1.1 Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is the traditional low-rank approximation method used for performing low-rank matrix approximation. SVD is basically a matrix decomposition method, where in the original matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ $(m \geq n)$ is decomposed into three factor matrices as [De Lathauwer et al., 2000]:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{i=n}^{n} \boldsymbol{\sigma}_i \boldsymbol{u}_i \boldsymbol{v}_i^T \tag{1.1}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices that contain the singular vectors and $\mathbf{\Sigma}$ is a diagonal matrix which contains the singular values in decreasing order in magnitude, i.e., $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$; For any matrix, its singular

values are unique only if they are all distinct, then their corresponding singular vectors are also unique. However, if partial singular values are equal, their corresponding singular vectors can span some subspace. In this case, any set of orthonormal vectors spanning this subspace can be regarded as the singular vectors.

Each singular value component contain different degree of information, of which the uninformative components can be truncated, which is very common in noise data. Assume that data matrix $\mathbf{A}$ is a low-rank matrix with noise: $\mathbf{A} = \mathbf{A}_0 + \mathbf{E}$, where the noise matrix $\mathbf{E}$ is relatively small compared with matrix $\mathbf{A}_0$. In such a situation, if only first r significant singular values are considered, we are able to reconstruct the original data matrix $\mathbf{A}$ as the best rank-r optimization problem:

$$\min \|\mathbf{A} - \mathbf{Z}\|_F^2, \qquad s.t. \quad \text{rank}(\mathbf{Z}) = r \tag{1.2}$$

where $\|\cdot\|_F$ is the Frobenius matrix norm. The above approximation problem has the solution as follow [Horn et al., 1990]:

$$\mathbf{Z} = \mathbf{A}_r := \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^T \tag{1.3}$$

where $\mathbf{U}_r = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_r]$, $\mathbf{V}_r = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_r]$, and $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$

Now the truncated matrix $\mathbf{A}_r$ contains the underlying factors to express the original noise data. Although SVD is a very efficient and simple dimension reduction algorithm, it has negative singular components, which make it difficult to interpret the basis components in real-word applications.

1.1.2 Nonnegative Matrix Factorization (NMF)

In many situations, the data elements are necessarily non-negative (e.g., images data, user-item matrix and matrix of word counts), and so their corresponding latent factors to represent data matrix should arguably be also non-negative. The recent

development of non-negative matrix factorization (NMF) is an attractive alternative to decompose the matrix, which leads to substantial improvements in interpretability of the latent factors. Rather than attempt to perform SVD decomposition of data matrix $\mathbf{A}$ into three matrices in Eq. (1.1), NMF decompose the matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ into two non-negative latent matrices by minimizing the follow objective function [Lee and Seung, 2001]:

$$\min \|\mathbf{A} - \mathbf{WH}\|_F^2, \qquad s.t. \quad \mathbf{W} \geq 0; \mathbf{H} \geq 0; \tag{1.4}$$

where $\mathbf{W} \in \mathbb{R}^{m \times r}$ and $\mathbf{H} \in \mathbb{R}^{r \times n}$ are two nonnegative factor matrices; r is the matrix rank approximation and usually is chosen to be much smaller than m or n. The Eq. (1.4) results in a compressed version of the original data matrix. Although above objective is convex with respect to $\mathbf{W}$ or $\mathbf{H}$ only, it is not jointly convex in both variables. Several optimization algorithm can be used to find local minima, such as project gradient descent (PGD) [Lin, 2007], alternating direction method of multipliers (ADMM) [Boyd et al., 2011] and multiplicative update rules [Lee and Seung, 2001]. Among different optimization algorithm, multiplicative update (MU) rules is perhaps a good compromise between time complexity and easy of implementation for solving Eq. (1.4). MU iteratively minimizes the objective function with respect to a single variable while fixing the remaining variables. This procedure continues until convergence. To be specific, the algorithm update each variable based on follow rules:

$$\mathbf{H}_{ij} \leftarrow \mathbf{H}_{ij} \frac{(\mathbf{W}^T \mathbf{A})_{ij}}{(\mathbf{W}^T \mathbf{WH})_{ij}} \tag{1.5}$$

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij} \frac{(\mathbf{AH}^T)_{ij}}{(\mathbf{WH}^T \mathbf{H})_{ij}} \tag{1.6}$$

The solution of Eq. (1.5) and Eq. (1.6) is mainly derived from the Karush-Kuhn-

Tucker (KKT) complementarity conditions and their correctness and convergence are guaranteed [Boyd and Vandenberghe, 2004].

The advantages of using NMF over SVD are storage requirements for NMF is much small than SVD, since $\mathbf{W}$ and $\mathbf{H}$ have much small size than the orthogonal matrices. In addition, The nonnegativity constraints on factor matrices naturally lead to better interpretation for original non-negative data. Consequently, NMF has been used successfully in many applications, including document clustering [Xu et al., 2003], recommender systems [Chen and Li, 2019e, Xu et al., 2003, Chen and Li, 2017b], community detection [Xie et al., 2013, Chen and Li, 2018b] and image processing in computer vision [Lee and Seung, 1999].

1.2 Tensor Factorization

Tensors, multidimensional extensions of matrices, are very powerful containers to express multi-aspect or multi-modal data [Kolda and Bader, 2009]. For instance, in content-aware recommender system, users can purchase an item, also annotate an text reviews to this item, and so on. The triple relationship of interactions can be modeled as a three-mode tensor $user \times item \times review$. In recent year, tensor factorization has been well applied to many applications, such as computer vision [Shashua and Hazan, 2005], signal processing [Cichocki et al., 2015] and network analysis [Agarwal et al., 2006, Chen and Li, 2018a]. We next give some preliminaries of tensor algebra.

The *order* of a tensor is the number of its dimensions, also known as ways or modes. A *fiber* is a vector extracted from a tensor by fixing every index but one. A *slice* is a matrix extracted from a tensor by fixing all but two indices. Note that an N-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ reduces to a vector when $N = 1$, and a matrix when $N = 2$. The $(i_1, \ldots, i_N)$-th element of $\mathcal{X}$ is denoted as $\mathcal{X}_{i_1,\ldots,i_N}$. *Matricization*, also

known as unfolding or flattening, is the process of reordering the elements of a tensor into a matrix. The mode-n matricization of an N-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is represented as $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 \ldots I_{n-1} I_{n+1} \ldots I_N}$ and is arranging the mode-n fibers of the tensor as columns of the long matrix. We then introduce some tensor operators.

The *norm* of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is the square root of the sum of the square of all its elements as

$$\|\mathcal{X}\| = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \ldots i_N}^2} \tag{1.7}$$

The *inner product* of two tensor with the same size $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ can be expressed as follow

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \ldots i_N} y_{i_1 i_2 \ldots i_N} \tag{1.8}$$

The *n-mode product* of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $\mathbf{A} \in \mathbb{R}^{J \times I_n}$ is denoted as

$$\mathcal{Y}_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_N} = (\mathcal{X} \times_n \mathbf{U})_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} u_{j i_n} \tag{1.9}$$

here $\mathcal{Y}$ is with size $I_1 \times I_2 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$. The product a tensor with a matrix in *n-mode* can change the basis of that mode of tensor. Several matrix product are very important in tensor factorization, so we also define them here.

The *Hadamard product* of two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$ results in a matrix with size

$m \times n$ and is denoted as

$$\mathbf{A} * \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & a_{13}b_{13} & \dots & a_{1n}b_{1n} \\ a_{21}b_{21} & a_{22}b_{22} & a_{23}b_{23} & \dots & a_{2n}b_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1}b_{m1} & a_{m2}b_{m2} & a_{m3}b_{m3} & \dots & a_{mn}b_{mn} \end{bmatrix} \tag{1.10}$$

The *Kronecker product* of matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{k \times l}$ results in a matrix with size $(mk) \times (nl)$ and is represented by

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & a_{13}\mathbf{B} & \dots & a_{1n}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & a_{23}\mathbf{B} & \dots & a_{2n}\mathbf{B} \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1}\mathbf{B} & a_{m2}\mathbf{B} & a_{m3}\mathbf{B} & \dots & a_{mn}\mathbf{B} \end{bmatrix} \tag{1.11}$$

The *Khatri–Rao product* of matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{k \times n}$ results in a matrix with size $(mk) \times (n)$ and is represented by

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{a_1} \otimes \mathbf{b_1} \quad \mathbf{a_2} \otimes \mathbf{b_2} \quad \cdots \quad \mathbf{a_n} \otimes \mathbf{b_n}] \tag{1.12}$$

In addition to tensor-matrix operator, there is a rich variety of tensor factorization in the literature. In the next subsection, we mainly introduce two most widely used tensor factorizations (CANDECOMP/PARAFAC (CP) and Tucker factorization) in dimension reductions, which can be treated as the starting point of many existing variants of tensor completion problem [Kolda and Bader, 2009].

1.2.1 Tensor CP Factorization

The canonical polyadic (CP) factorization (also known as PARAFAC/CANDECOMP) was independently developed by Carroll and Chang [Carroll and Chang, 1970], and Harshman [Harshman et al., 1970]. As shown in Figure 1.2(a), the CP factorization

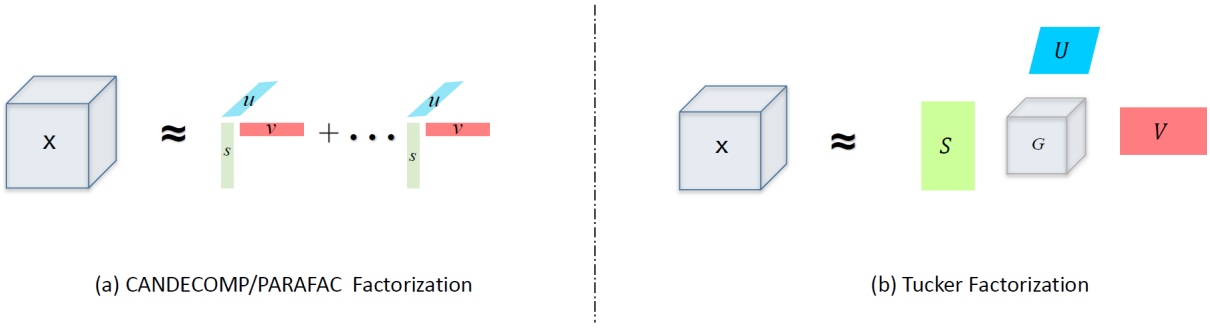

Figure 1.2: (a) CP factorization of a three-way array, and (b) Tucker factorization of a three-way array.

decomposes a tensor into a sum of multiple rank-one tensors. For example, the CP factorization of a third-order tensor $\mathcal{X} \in \mathbb{R}^{M \times N \times L}$ is given as:

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{u_r} \circ \mathbf{s_r} \circ \mathbf{v_r} \tag{1.13}$$

here $\mathbf{u_r} \in \mathbb{R}^M$, $\mathbf{s_r} \in \mathbb{R}^N$, $\mathbf{v_r} \in \mathbb{R}^L$; $\circ$ is the vector outer product. The three-way outer product of vector $\mathbf{u_r}$, $\mathbf{s_r}$ and $\mathbf{v_r}$ is defined by

$$\mathcal{X}_{ijk} \approx (\mathbf{u_r} \circ \mathbf{s_r} \circ \mathbf{v_r})_{ijk} = \mathbf{u_r}(i)\mathbf{s_r}(j)\mathbf{v_r}(k) \tag{1.14}$$

The R in Eq. (1.13) is number of factors and the minimal R that approximate the tensor $\mathcal{X}$ is called the tensor CP rank. The *factor matrices* of tensor $\mathcal{X}$ is given by stack all of the vectors, i.e., $\mathbf{U} = [\mathbf{u_1} \quad \mathbf{u_2} \quad \cdots \quad \mathbf{u_R}]$ and likewise for $\mathbf{S}$ and $\mathbf{V}$.

In order to obtain the factor matrices $\mathbf{U}$, $\mathbf{S}$ and $\mathbf{V}$, one can minimize the following optimization problem:

$$\min_{\mathbf{U},\mathbf{S},\mathbf{V}} \|\mathcal{X} - \sum_{r=1}^{R} \mathbf{u_r} \circ \mathbf{s_r} \circ \mathbf{v_r}\|_F^2 \tag{1.15}$$

The above objective function in not joint convex with respect to $\mathbf{U}, \mathbf{S}$ and $\mathbf{V}$. The Alternating Least Squares (ALS) can be used to split original optimization problem into several linear least square problems for each mode of tensor [Kolda and Bader, 2009].

1.2.2 Tensor Tucker Factorization

Another popular tensor factorization is the Tucker factorization, which is first proposed by Tucker in 1963 [Tucker, 1963] and further develop in his subsequent literature [Tucker, 1966]. As shown in Figure 1.2(b), the Tucker factorization of tensor is to decompose the tensor into a core tensor as well as multiple matrices along each mode. Given an third-order tensor $\mathcal{X} \in \mathbb{R}^{M \times N \times L}$, its Tucker factorization is as:

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{S} \times_3 \mathbf{V} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} \mathbf{u_p} \circ \mathbf{s_q} \circ \mathbf{v_r} \tag{1.16}$$

where $\mathbf{U} \in \mathbb{R}^{M \times P}$, $\mathbf{S} \in \mathbb{R}^{N \times Q}$, $\mathbf{V} \in \mathbb{R}^{L \times R}$ are the factor matrices, which is usually orthogonal; The $\mathcal{G} \in \mathbb{R}^{P \times Q \times R}$ is the core tensor and its elements represent the level of interaction among different modes of tensor.

Similar to CP factorization, we can optimize the follow optimization problem to obtain the factor matrices $\mathbf{U}$, $\mathbf{S}$ and $\mathbf{V}$, and core tensor $\mathcal{G}$:

$$\min_{\mathcal{G},\mathbf{U},\mathbf{S},\mathbf{V}} \|\mathcal{X} - \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{S} \times_3 \mathbf{V}\|_F^2 \qquad s.t. \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}, \mathbf{S}^T\mathbf{S} = \mathbf{I}, \mathbf{V}^T\mathbf{V} = \mathbf{I} \tag{1.17}$$

The above problem can be effectively solve by Higher-Order Orthogonal Iteration (HOOI) or Higher-Order Singular Value Decomposition [Kolda and Bader, 2009].

1.3 Organization

The first chapter presents an overview of the dimension reduction techniques on network analysis and some linear algebra about matrix and tensor factorization. The rest of the dissertation is composed of six parts, each of them attempts to develop an algorithm for a particular mining task, especially in drug discovery and recommender system. In each part, we introduce the challenges and propose our solutions for learning tasks. We summarize the main problems of each part in the form of questions in

Table 1.1: Thesis Overview.

Part	Research Problem	Ch.
I: Matrix Factorization	**Drug repositioning**: How can we model drug-disease relationships into a heterogeneous network and predict the novel edges among drugs and diseases?	2
	Drug combinations: How can we solve drug combinations problem. Unlike like drug repositioning, the drug pairs are considered?	3
II: Tensor Factorization	**Drug combinations**: How to infer more meaningful drug combinations by integrating multiple heterogeneous data sources of drugs and diseases?	4
	Tensor for drug-target-disease discovery: How to infer potential interactions of drug-target-disease in human metabolic system with linear tensor model?	5
	Neural tensor for drug-target-disease discovery: How to infer potential interactions of drug-target-disease with nonlinear tensor model?	6

Table 1.1.

Chapter 2: Drug repositioning, which exploits new therapeutic indications for existing drugs, is a promising strategy in drug discovery. New biomedical insights of drug-target-disease relationships are important in drug repositioning, and such relationships have been intensively studied recently. Most of the existing studies have utilized network-based computational approaches based on drug and disease similarities. However, one common limitation of existing approaches is that both drug similarities and disease similarities are defined based on a single feature of drugs/diseases. In reality, the relationships between drug pairs (or disease pairs) can be characterized based on many different features. With rapid accumulation of such information about drugs and diseases, it is increasingly important to include them in drug repositioning studies. Therefore, in this chapter, we propose a flexible and robust multi-source learning (FRMSL) framework to integrate multiple heterogeneous data sources for drug-disease association predictions. We first construct a two-layer heterogeneous network consisting of drug nodes, disease nodes and known drug-disease relationships. The drug repositioning problem can thus be treated as a missing link

prediction problem on the heterogeneous network and can be solved using Kronecker regularized least square (KronRLS) method. Multiple data sources describing drugs and diseases are incorporated into the framework using similarity-based kernels. In practice, a great challenge in such data integration projects is the data incompleteness problem due to the nature of data generation and collection. To address this issue, we develop a novel multi-view learning algorithm based on symmetric nonnegative matrix factorization (SymNMF).

Chapter 3: Personalized treatments and targeted therapies are the most promising approaches to treat complex human diseases. However, drug resistance is often acquired after treatments. To reduce drug resistance, combinational drug therapies have been considered as a promising strategy in drug discovery. Moreover, the emerging of large-scale genomic, chemical and biomedical data provides new opportunities for drug combinations. Here we propose a network approach, called MCDC, that integrates multiple data sources describing drugs, target proteins, and diseases to predict beneficial drug combination. Specifically, MCDC integrates diverse drug-related information (e.g., chemical structure, target profile), disease-related information (e.g., disease phynotype), together with their interactions to construct a two-layer heterogeneous network. MCDC then predicts drug combinations for each disease using a link prediction algorithm. Our approach has great potential to accelerate the development of drug combination treatments.

Chapter 4: We propose DrugCom, a tensor-based framework for computing drug combinations across different diseases by integrating multiple heterogeneous data sources of drugs and diseases. DrugCom first constructs a primary third-order tensor (i.e., drug×drug×disease) and several similarity matrices from multiple data sources regarding drugs (e.g., chemical structure) and diseases (e.g., disease phenotype). DrugCom then formulates an objective function, which simultaneously factorizes coupled tensor and matrices to reveal the molecular mechanisms of drug synergy. We

adopt the alternating direction method of multipliers algorithm to effectively solve the optimization problem. Extensive experimental studies using real-world datasets demonstrate superior performance of DrugCom. Our comprehensive approach on synergistic effect prediction of drug combinations has great potential to accelerate the development of drug combination treatments for complex diseases.

Chapter 5: The growing availability of new types of data on the internet brings great opportunity of learning a more comprehensive relationship among drugs, targets (druggable proteins), and diseases. However, existing methods often consider drug-target interactions or drug-disease interactions separately, which ignore the dependencies among these three entities. Also, they cannot directly incorporate rich heterogeneous information from diverse resources. Here we investigate the utility of tensor factorization to model the relationships of drug-target-disease, specifically leveraging different types of online data. Our motivation is two-fold. First, drugs interact with targets in cells to modulate target activities, which in turn alter biological pathways to promote healthy functions and to treat diseases. Instead of binary relationships of drug-disease or drug-target, a tighter triple relationships drug-target-disease should be exploited to better understand drug mechanism of actions (MoAs). Second, medical data could be collected from different sources (i.e., drug's chemical structure, target's sequence, or expression measurements). Therefore, exploiting the complementarity effectively among multiple sources is of great importance. We achieve this goal by formulating the problem into a coupled tensor-matrix factorization optimization problem and directly optimize it on the nonlinear manifold.

Chapter 6: Precise medicine recommendations provide more effective treatments and cause fewer drug side effects. A key step is to understand the mechanistic relationships among drugs, targets, and diseases. Tensor-based models have the ability to explore relationships of drug-target-disease based on large amount of labeled data. However, existing tensor models fail to capture complex nonlinear dependen-

cies among tensor data. In addition, rich medical knowledge are far less studied, which may lead to unsatisfied results. Here we propose a Neural Tensor Network (NeurTN) to assist personalized medicine treatments. NeurTN seamlessly combines tensor algebra and deep neural networks, which offers a more powerful way to capture the nonlinear relationships among drugs, targets, and diseases. To leverage medical knowledge, we augment NeurTN with geometric neural networks to capture the structural information of both drugs' chemical structures and targets' sequences. Extensive experiments on real-world datasets demonstrate the effectiveness of the NeurTN model.

Chapter 2

Heterogeneous Network for Drug Repositioning

2.1 Introduction

Traditional drug discovery using high-throughput screening could identify a new drug against a chosen target protein for a special disease. However, even with advanced automotive robotics systems, the whole screening process is still expensive, time consuming, and with high failure rates. A conservative estimate is that it takes on average 13.5 years and $1.8 billion to bring a new drug into the market [Paul et al., 2010]. Drug repositioning, which tries to find new indications of approved drugs, is gaining increasing interests both in academia and pharmaceutical industry recently[Li et al., 2015a]. For example, Avastin was initially used to treat non-small-cell lung cancer, later was also approved for metastatic breast cancer. Minoxidil was originally used for hypertension, but was then approved for hair loss [Dudley et al., 2011]. Drug repositioning provides a promising alternative strategy to reduce the total cost because existing drugs have already been approved by the U.S. Food and Drug Administration (FDA) with known toxicity information.

A recent report has shown that among the 84 new drug products on the market in 2013, new clinical indications of existing drugs accounted for 20% [Graul et al., 2014]. Therefore, drug repositioning has played an important role in current drug development.

Recently, several network-based computational approaches for drug repositioning have been proposed [Chiang and Butte, 2009, Wu et al., 2013a, Wang et al., 2014, Chen et al., 2015, Luo et al., 2016, Martínez et al., 2015]. For example, a drug-disease network was constructed to predict novel indications of drugs and results were highly enriched in clinical trials [Chiang and Butte, 2009]. In another study, a weighted heterogeneous network was applied to identify the connected communities of drugs and diseases using a network clustering approach [Wu et al., 2013a]. Two network topology-based inference methods, ProbS and HeatS, were also introduced to predict new indications for different diseases [Chen et al., 2015]. DrugNet was another network-based prioritization approach simultaneously incorporating information about diseases, drugs and targets to perform drug-disease prioritization [Martínez et al., 2015]. TL_HGBI was a three-layer heterogeneous network model that seamlessly integrated drug repositioning and drug-target into a unified framework [Wang et al., 2014]. A modified random walk algorithm was then developed to rank all candidate drugs for every disease. Similarly, MBiRW used a bi-random walk algorithm on a two-layer network to infer the potential drug-disease associations [Luo et al., 2016]. Most frameworks have utilized the guilt-by-association principle [Altshuler et al., 2000], which states that if two diseases share similar therapeutic profiles, then drugs for one disease could also treat the other with high possibility. However, most previous methods only incorporate one data source in calculating drug-drug similarity (*e.g.*, similarity based on chemical structures) and one data source for disease-disease similarity (*e.g.*, phenotype-based similarity), which may not be able to capture the complex interplay between drugs and diseases. In addition, for those non small molecule drugs (*e.g.*,

Leuprolide), similarities cannot be measured based on chemical structures. In this case, other type of information (*e.g.*, drugs' side effects) may provide valuable insights in learning drug-disease relationships.

To overcome above limitations, we propose a **F**lexible and **R**obust **M**ultiple **S**ources **L**earning (FRMSL) framework to integrate distinct sources of biological datasets to better define similarities among drugs and among diseases. Our model can be viewed as a two-layer heterogeneous graphic model, where the two layers representing drugs and diseases. Similarity kernels are defined for nodes within each layer based on multiple data sources. Links across layers are defined based on known drug indications. The drug repositioning problem is thus the same as the missing link prediction problem on the network. We solve the problem by utilizing Kronecker regularized least squares approach (KronRLS) [van Laarhoven et al., 2011]. It has been shown that integration of large-scale proteomic, genomic, transcriptomic and metabolic data into networks could provide new insights of molecular basis of complex human diseases [Hopkins, 2008]. However, due to the nature of data collection (*i.e.*, data were generated by different labs in different time), it is very unlikely that all datasets will be available for all drugs/diseases. Data incompleteness is always a significant challenge when merging them from different sources. To address this issue, we further develop a multi-view learning algorithm and subsequently solve the problem utilizing symmetric nonnegative matrix factorization (SymNMF). The basic idea is that missing information from one data source/view could be inferred by borrowing information from other sources/views. This can also be viewed as a network alignment algorithm for drug/disease layer, respectively, where each data source adds a sublayer to one of the two layers and the SymNMF forces them to share a common network structure in terms of their weights (*i.e.*, similarities) . FRMSL is flexible because it can easily incorporate more data sources once they are available in the future. Extensive experimental studies have shown the benefits of multi-source learn-

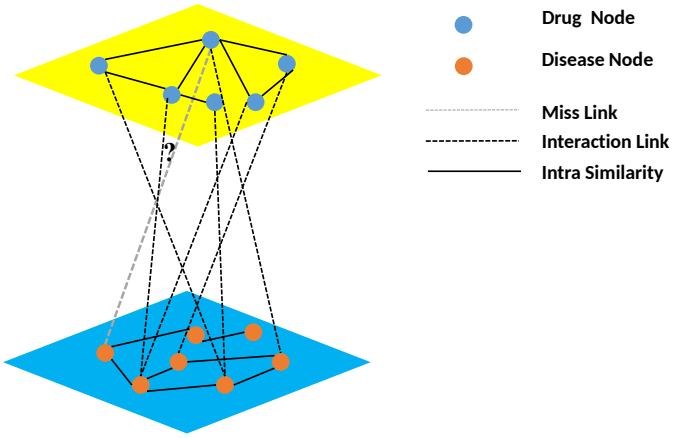

Figure 2.1: A two-layer heterogeneous network consisting of drug-drug similarities (red line), disease-disease similarities (blue line) and known drug-disease interactions (black line). The goal is to predict missing links (black dashed lines) across the drug layer and the disease layer.

ing and multi-view learning, and the proposed FRMSL greatly outperforms several state-of-the-art network-based methods.

2.2 Problem Formulation

In this section, we formulate the drug repositioning task as a missing link prediction problem on a two-layer heterogeneous network as shown in Figure 2.1, consisting of drugs and diseases. The problem will then be solved using the Kronecker regularized least squares approach. Given a set of n drugs $C = \{c_1, c_2, \ldots, c_n\}$, a set of m diseases $D = \{d_1, d_2, \ldots, d_m\}$, and a set of known drug-disease relationships Y (where $y_{ij} = 1$ if drug c_i can treat disease d_j, and $y_{ij} = 0$ otherwise), we construct a two-layer heterogeneous network (Figure 2.1). The network consists of two types of nodes: drug nodes and disease nodes; three types of edges: drug-drug edges, disease-disease edges and drug-disease edges. The edges between the same type of nodes are labeled

Table 2.1: Main Symbols in Chapter 2.

Symbol	Description
$K_c^{(i)}$	the drug kernel matrix for i^{th} view
$K_d^{(j)}$	the disease kernel matrix for j^{th} view
K_c^*	the combined drug kernel
K_d^*	the combined disease kernel
$G^{(i)}$	the latent factor for i^{th} view kernel matrix
G^*	the common latent factor
$W^{(i)}$	the weighted matrix for i^{th} view kernel matrix
n	the number of drugs
m	the number of diseases
ω_c^i	the convex weight parameter for drug $K_c^{(i)}$
ω_d^j	the convex weight parameter for disease $K_d^{(j)}$
λ	the regularization parameter in KronRLS
α_i	the regularization parameter in SymNMF

based on their similarities. Drug-disease edges are defined for all drug-disease pairs (i, j) with $y_{ij} = 1$. The goal of the drug repositioning problem is to predict missing links between the drug-layer and the disease-layer based on the information contained in the two-layer network. Although the two-layer heterogeneous network model is the same structure-wise as the models in some previous studies [Wang et al., 2014, Luo et al., 2016], our unique contribution in this study is the integration of multiple data sources and imputation of missing in one data source by borrowing information from other data sources. Some important notations introduced here and to be used in the paper are listed in Table 2.1.

2.3 FRMSL Framework

In this section, we first solve the missing link problem by utilizing the Kronecker regularized least squares approach (KronRLS) [van Laarhoven et al., 2011]. We include a kernel fusion step to incorporate multiple data sources for each layer. The missing data imputations is then solved based on SymNMF. Finally we give a summary of our computational framework FRMSL.

2.3.1 KronRLS Algorithm

The missing link prediction problem can be solved using the regularized least square (RLS) approach in order to have good generalization power. Given a set of training data $S = \{(\boldsymbol{x_i}, y_i) : \boldsymbol{x_i} = (c_i, d_i); 0 \leq i \leq l\}$, where l is the total number of training data and the goal of the RLS algorithm is to find the mapping function $f : X \rightarrow Y$, by minimizing the following objective function:

$$\underset{f \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{l} \sum_{i=1}^{l} (y_i - f(\boldsymbol{x_i}))^2 + \lambda \|f\|_{\mathcal{H}}^2 \tag{2.1}$$

where $\|f\|_{\mathcal{H}}^2$ is a Hilbert space norm of function f; λ is a trade-off between regression accuracy and the complexity of the function f in the Hilbert space. The representer theorem guarantees Eq. (2.1) has a closed-form solution [Scholkopf and Smola, 2001]:

$$f(\boldsymbol{x}) = \sum_{i=1}^{l} a_i K(\boldsymbol{x}, \boldsymbol{x_i}) \tag{2.2}$$

where $\boldsymbol{a} \in \mathbb{R}^l$ can be obtained by solving the following linear equation: $(\boldsymbol{K} + \lambda \boldsymbol{I})\boldsymbol{a} = \boldsymbol{y}$; and $\boldsymbol{K}$ is a pairwise instance kernel which represents the similarity between any two data points in the Hilbert space. In the current study, we define the pairwise instance kernel as the product of a drug-drug similarity kernel and a disease-disease similarity kernel, to incorporate information from both drugs and diseases. More specifically, for two instances $\boldsymbol{x_i} = (c_i, d_i)$ and $\boldsymbol{x_j} = (c_j, d_j)$, the kernel is defined as:

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = K_c(c_i, c_j) K_d(d_i, d_j) \tag{2.3}$$

where K_c is a kernel to measure the similarity between drug c_i and drug c_j; K_d is a kernel to measure the similarity of diseases d_i and disease d_j. The drug-drug kernel K_c and the disease-disease kernel K_d can then be defined based on multiple sources.

To solve the optimization problem efficiently, we adopt the kernel-based KronRLS

algorithm [van Laarhoven et al., 2011], which further takes advantage of spectral decompositions of kernel matrices to speed up the calculation. Let $K_c = V_c \Lambda_c {V_c}^T$ and $K_d = V_d \Lambda_d {V_d}^T$ be the spectral decomposition of kernel K_c and K_d, respectively. Eq. (2.3) can be rewritten as Kronecker product kernel $K = K_c \otimes K_d = V \Lambda V^T$, where $V = V_c \otimes V_d$, $\Lambda = \Lambda_c \otimes \Lambda_d$, and $A \otimes B$ is the Kronecker product of two matrices A and B. Based on these transformations, one can get the prediction in the form of [van Laarhoven et al., 2011]

$$\hat{Y} = V_c Z^T {V_d}^T \tag{2.4}$$

where

$$vec(Z) = (\Lambda_c \otimes \Lambda_d)(\Lambda_c \otimes \Lambda_d + \lambda I)^{-1} vec({V_d}^T Y^T V_c),$$

and $vec(\cdot)$ is the vectorization operator that stacks all columns of a matrix into a column vector.

2.3.2 Fusion Kernels Across Multiple Heterogeneous Sources

With increasing available biological datasets regarding drugs and diseases, drug-drug similarities and disease-disease similarities can be measured differently using different datasets. For example, one can define drug-drug similarities based on their chemical structures, or based on their side-effect profiles. Different datasets represent different aspects of drugs and/or their interplays with the human body system. Incorporating multiple heterogeneous sources can provide complementary and comprehensive understanding of a drug's mechanism of action (MoA). Furthermore, different datasets serve as different views of the same objects (*e.g.*, drugs or diseases), which can be used in missing data imputation as presented in the next subsection.

For each data source, we will construct a similarity-based kernel (the details of

datasets and kernels associated with them will be discussed in the next section). Given a set of kernels representing drug-drug similarities $k_C = \{K_c^{(1)}, K_c^{(2)}, \ldots, K_c^{(n_c)}\}$, and a set of kernels measuring disease-disease similarities $k_D = \{K_d^{(1)}, K_d^{(2)}, \ldots, K_d^{(n_d)}\}$, where n_c and n_d denote the numbers of kernels for drugs and diseases, we propose to combine them by a weighted sum linear convex function to obtain an overall drug kernel K_c^* and an overall disease kernel K_d^*, respectively [Scholkopf and Smola, 2001]:

$$K_c^* = \sum_{i=1}^{n_c} \omega_c^i K_c^{(i)} \qquad\qquad K_d^* = \sum_{j=1}^{n_d} \omega_d^j K_d^{(j)} \tag{2.5}$$

The ω_c^i and ω_d^j are corresponding convex weight parameters with the constraints $\sum_{i=1}^{n_c} \omega_c^i = 1$ and $\sum_{j=1}^{n_d} \omega_d^j = 1$. These values can be optimally trained using the cross-validation method on a training dataset. Alternatively, they can be assigned based on prior knowledge about relative reliability of different data sources.

2.3.3 Complete Kernels Across Multiple Data Sources

As we discussed earlier, a significant challenge in data integration is data incompleteness. For example, one can define drug-drug similarities using their side-effect profiles. However, many drugs may not have their side-effect profiles recorded in databases (*e.g.*, SIDER [Kuhn et al., 2015]). Missing data points from a data source will lead to an incomplete kernel matrix with rows and columns missing. Previous studies have filled the missing rows and columns with zeros, primarily for computational convenience [van Laarhoven et al., 2011, Wang et al., 2014, Wu et al., 2013a]. However, such a treatment can adversely affect prediction accuracy. Inspired by the idea of multi-view learning [Liu et al., 2013b, Shao et al., 2015, Chen and Li, 2017b], we propose a novel framework to perform missing data imputation of multiple kernels based on symmetric nonnegative matrix factorization (SymNMF), which can also viewed as a network alignment algorithm that tries to align different kernels to

a common clustering structure [Liu et al., 2013b]. The algorithm is rooted in the guilt-by-association principle. For example, the weighted cluster structure of drug layer (Figure 2.1) should be highly aligned from both views of chemical structure and drugs side-effect profiles because similar drugs tend to have similar drugs side-effects [Campillos et al., 2008]. Therefore, the missing components in one dataset can be inferred based on information from other datasets.

Given a set of kernels $\{K^{(i)} : 1 \leq i \leq n_v; K^{(i)} \in \mathbb{R}^{N \times N}\}$ constructed from N instances, we first define an indicator matrix $M \in \mathbb{R}^{n_v \times N}$ with its elements defined as [Shao et al., 2015]:

$$M_{i,j} = \begin{cases} 1 & \text{if } j^{th} \text{ instance exists in view } i \\ 0 & \text{otherwise} \end{cases}$$

As a symmetric nonnegative matrix, a kernel matrix can be factorized into two matrices: $K^{(i)} \approx G^{(i)}{G^{(i)}}^T, 1 \leq i \leq n_v$, where $G^{(i)} \in \mathbb{R}^{N \times r}$ is the base of the kernel matrix $K^{(i)}$. Typically, $r \ll rank(K^{(i)})$. One can show that this factorization is equivalent to K-means-type clustering [Ding et al., 2005]. Our proposed network alignment algorithm is to jointly factorize all kernel matrices and push all the factors $G^{(i)}$ towards a consensus matrix G^* through regularization [Liu et al., 2013b, Chen and Li, 2018b]. The objective function involving multiple kernels can then be written as:

$$\begin{aligned} \min_{\{G^{(i)}\}, G^*} \mathcal{O} &= \sum_{i=1}^{n_v} (\|W^{(i)}(K^{(i)} - G^{(i)}{G^{(i)}}^T)\|_F^2 + \alpha_i \|W^{(i)}(G^{(i)} - G^*)\|_F^2) \\ \text{s.t.} \quad & G^{(i)} \geq 0, G^* \geq 0, i = 1 \ldots n_v \end{aligned} \tag{2.6}$$

where the weighting parameters α_is control the relative alignment agreement between $G^{(i)}$ and the common matrix G^*. $W^{(i)} \in \mathbb{R}^{N \times N}$ is a diagonal matrix for incomplete view i, with the diagonal elements defined as [Shao et al., 2015] :

$$W_{jj}^{(i)} = \begin{cases} 1 & \text{if } i^{th} \text{ view contains } j^{th} \text{ item } (M_{i,j} = 1) \\ \frac{\sum_{j=1}^{N} M_{i,j}}{N} & \text{otherwise} \end{cases}$$

Once the optimization problem is solved, we can obtain a set of matrix factors $\{G^{(1)}, G^{(2)}, \ldots, G^{(n_v)}\}$, we then reconstruct the incomplete kernel matrices by $K^{(i)} \approx G^{(i)}{G^{(i)}}^T, 1 \leq i \leq n_v$.

2.3.4 Optimization

In the following, we solve the optimization problem defined in Eq. (2.6). Because the objective function is not jointly convex for $G^{(i)}$ and G^*, an alternating optimization scheme has to be applied [Liu et al., 2013b]. More specifically, the following two steps are iterated until the solution converges: (1) fixing $G^{(i)}$, minimize $\mathcal{O}$ over G^*; (2) fixing G^*, minimize $\mathcal{O}$ over $G^{(i)}$. We discuss these two steps in sequel.

(1) Fixing $G^{(i)}$, minimize $\mathcal{O}$ over G^*.

The objective function $\mathcal{O}(G^*)$ does not depend on $G^{(i)}$, we can find the optimal solution by taking the derivative $\mathcal{O}(G^*)$ over G^*:

$$\begin{aligned} \frac{\partial \mathcal{O}(G^*)}{\partial G^*} &= \sum_{i=1}^{n_v} \alpha_i(-2{W^{(i)}}^T W^{(i)} G^{(i)} + 2{W^{(i)}}^T W^{(i)} G^*) = 0 \\ \text{s.t.} \quad & G^{(i)} \geq 0, \quad i = 1, \ldots, n_v \end{aligned} \tag{2.7}$$

Solving the equation, we have an exact solution for G^*:

$$G^* = \frac{\sum_{i=1}^{n_v} \alpha_i {W^{(i)}}^T W^{(i)} G^{(i)}}{\sum_{i=1}^{n_v} \alpha_i {W^{(i)}}^T W^{(i)}} \geq 0 \tag{2.8}$$

Since ${W^{(i)}}^T W^{(i)}$ is a positive diagonal matrix and α_i is a positive constant and the constraint $G^{(i)} \geq 0$, the solution above is thus always nonnegative.

(2) Fixing G^*, minimize $\mathcal{O}$ over $G^{(i)}$.

When G^* is fixed, the objective function $\mathcal{O}(G^{(i)})$ becomes a standard SymNMF problem with additional regularization [Ding et al., 2005]. Furthermore, because the calculation of $G^{(i)}$ does not depend on $G^{(i')}$ when $i' \neq i$, we can thus minimize the objective function $\mathcal{O}(G^{(i)})$ with respect to $G^{(i)}$ for each view independently.

$$\begin{aligned}\frac{\partial \mathcal{O}(G^{(i)})}{\partial G^{(i)}} =& -4{W^{(i)}}^T W^{(i)} K^{(i)} G^{(i)} + 4{W^{(i)}}^T W^{(i)} G^{(i)} {G^{(i)}}^T G^{(i)} \\ &+ 2\alpha_i {W^{(i)}}^T W^{(i)} G^{(i)} - 2\alpha_i {W^{(i)}}^T W^{(i)} G^* \\ \text{s.t.} \quad & G^{(i)} \geq 0, , \quad i = 1, \ldots, n_v\end{aligned} \tag{2.9}$$

Using the Karush-Kuhn-Tuvker (KKT) complementary condition, we can get:

$$\begin{aligned}&(-4{W^{(i)}}^T W^{(i)} K^{(i)} G^{(i)} + 4{W^{(i)}}^T W^{(i)} G^{(i)} {G^{(i)}}^T G^{(i)} \\ &+ 2\alpha_i {W^{(i)}}^T W^{(i)} G^{(i)} - 2\alpha_i {W^{(i)}}^T W^{(i)} G^*) * G^{(i)} = 0\end{aligned} \tag{2.10}$$

where $A * B$ denotes element-wise multiplication of two matrices A and B. Based on the fixed point equation above, we can derive the updating rule for $G^{(i)}$ as:

$$G^{(i)}_{j,k} \leftarrow G^{(i)}_{j,k} \sqrt{\frac{(2{W^{(i)}}^T W^{(i)} K^{(i)} G^{(i)} + \alpha_i {W^{(i)}}^T W^{(i)} G^*)_{j,k}}{(2{W^{(i)}}^T W^{(i)} G^{(i)} {G^{(i)}}^T G^{(i)} + \alpha_i {W^{(i)}}^T W^{(i)} G^{(i)})_{j,k}}} \tag{2.11}$$

We will iteratively update variable G^* and $G^{(i)}$ using above updating rules until the objective function Eq.(2.6) converges.

2.3.5 Convergence Analysis

The algorithm converges to a local minimum, which can be proved by using the auxiliary function approach [Lee and Seung, 2001, Liu et al., 2013b]. The update for G^* in Eq. (2.8) gives an exact analytical solution for the minimization of $\mathcal{O}$ when $G^{(i)}$ are fixed. Therefore, we only need to prove that the objective function $\mathcal{O}$ is non-increasing when we update $G^{(i)}$, for $1 \leq i \leq n_v$. We first introduce the definition

of the auxiliary function.

Definition 1. *A function $Z(h, \hat{h})$ is an auxiliary function for a given function $J(h)$ if the conditions*

$$Z(h, \hat{h}) \geq J(h) \quad and \quad Z(h, h) = J(h) \tag{2.12}$$

are satisfied for any given $h, \hat{h}$ [Lee and Seung, 2001].

Lemma 1. *If Z is an auxiliary function for J, then J is non-increasing under the update $h^{(t+1)} = \underset{h}{\operatorname{argmin}} Z(h, h^{(t)})$ [Lee and Seung, 2001].*

Proof. $J(h^{(t+1)}) \leq Z(h^{(t+1)}, h^{(t)}) \leq Z(h^{(t)}, h^{(t)}) = J(h^{(t)})$ □

Theorem 1 gives an auxiliary function for the objective function $\mathcal{O}$ in Eq. (2.6) with respect to $G^{(i)}$.

Theorem 1. *let $J(G^{(i)})$ denote the sum of all terms in Eq. (12) that contain $G^{(i)}$, then the following function*

$$\begin{aligned} Z(G^{(i)}, G'^{(i)}) = & -2 \sum_{pqr} (W^{(i)^T} W^{(i)} K^{(i)})_{pr} {G'^{(i)}}_{rq} {G'^{(i)}}_{pq} \left(1 + log \frac{{G^{(i)}}_{rq} {G^{(i)}}_{pq}}{{G'^{(i)}}_{rq} {G'^{(i)}}_{pq}}\right) \\ & + \sum_{rq} \frac{(W^{(i)^T} W^{(i)} G'^{(i)} G'^{(i)^T})_{rq} (G^{(i)} G^{(i)^T})^2_{rq}}{(G'^{(i)} G'^{(i)^T})_{rq}} + \alpha_i \sum_{rq} \frac{(W^{(i)^T} W^{(i)} G'^{(i)})_{rq} (G^{(i)})^2_{rq}}{(G'^{(i)})_{rq}} \\ & - 2\alpha_i \sum_{pqr} (G^* W^{(i)} W^{(i)})_{pr} {G'^{(i)}}_{rq} \left(1 + log \frac{{G^{(i)}}_{rq}}{{G'^{(i)}}_{rq}}\right) \end{aligned} \tag{2.13}$$

is an auxiliary function for $J(G^{(i)})$. Furthermore, it is a convex function w.r.t. $G^{(i)}$ and has a global minimum.

The formal proof are similar to [Lee and Seung, 2001] and the proof can be found in Appendix.

2.4 Experiments and Results

In this section, we first define similarity measures/kernels from multiple data sources for both drugs and diseases. We take those measures from recent literatures that have been shown effective. We then evaluate the proposed FRMSL algorithm on a gold standard dataset and compare its performance with two state-of-the-art network-based methods published recently, as well as the KronRLS algorithm (without missing value imputation). Furthermore, we also investigate the performance of FRMSL on an independent test dataset. Finally, we present case study results on five complex human diseases to illustrate the practical usefulness of FRMSL in drug discovery.

2.4.1 Datasets

The gold standard dataset with known drug-disease associations used in this study is obtained from Gottlieb *et al.* [Gottlieb et al., 2011], which consists of comprehensive drug-disease relationships from multiple datasets. There are in total 1933 known drug-disease interactions involving 593 drugs assembled from DrugBank[1] and 313 human diseases listed in the Online Mendelian Inheritance in Man (OMIM) database[2]. Additional information about drugs, including drug side-effects, drug target profiles, and drug 3D structures for calculating binding site similarities, are extracted from SIDER[3], Uniprot[4], and PDB[5], respectively.

Based on different types of features, we define four distinct kernels to measure drug-drug similarities and three distinct kernels for disease-disease similarities. Drug similarities are defined based on drug chemical structures, drug protein target amino acid sequences, drug side-effects, and drug target binding site information. Disease similarities are defined primarily based on various disease phenotype descriptions

[1]https://www.drugbank.ca/
[2]https://www.omim.org/
[3]http://sideeffects.embl.de/
[4]https://www.uniprot.org/
[5]https://www.rcsb.org/

and disease ontologies. All of the similarity measures are finally normalized in the range of $[0, 1]$. Some of the similarity matrices are already positive definite when constructed and are obviously valid kernels (*e.g.*, the drug chemical structure similarity [Mahé et al., 2006]). Even when a similarity matrix (symmetric, non-negative) is not strictly positive definite, one can add a small strictly positive definite kernel (e.g. $k'(x_i, x_j) = \delta_{ij}$) to obtain a positive definite matrix [Scholkopf and Smola, 2001]. The four drug-drug similarities are constructed as follows.

(1) Chemical structure similarity: Chemical properties of a drug are evidently related to its ultimate therapeutic effects [Dudley et al., 2011]. In this article, chemical structures of drugs in Canonical SMILES (Simplified Molecular Input Line Entry Specification) form are downloaded from DrugBank. The Chemical Development Kit is then applied to compute the Tanimoto similarity score of two drugs using their corresponding 2D chemical fingerprints [Steinbeck et al., 2006].

(2) Drug side-effect similarity: Drug side-effects, representing unintended consequences of drug activities, provide another measure of drug relationships in terms of their therapeutic effects. It has been shown that drug side-effect is an important factor and contributes in revealing novel drug-target interactions [Campillos et al., 2008]. In this work, drug side-effects are obtained from SIDER database. Each drug is then represented by a binary profile, in which the existence or absence of corresponding side effect keyword is encoded as 1 or 0. Finally, a weighted cosine correlation coefficient is used to calculate the similarity score [Takarabe et al., 2012].

(3) Drug-target sequence similarity: Protein target information of drugs is another important feature that can be used to characterize drug relationships. For drugs with known protein targets, their similarities are measured based on the Smith-Waterman alignment scores of the amino acid sequences of their corresponding targets [Smith and Waterman, 1981]. The similarity scores are further normalized based on the method in [Bleakley and Yamanishi, 2009]. For drugs with more than one tar-

gets, the similarity score is the average of the scores computed from all the target proteins.

(4) Drug binding site similarity: Ligand binding sites are critical in defining relationships between drugs and their targets [Capra et al., 2009]. We thus define a score to capture the binding site similarities among targets. Target protein structures are first downloaded from PDB. Then a binding similarity score is calculated based on the local structural alignment of two proteins using SMAP software [Xie et al., 2009]. Similarly, for drugs with multiple targets, the final similarity score is the average of the scores computed using all targets.

The three disease-disease similarities are constructed as follows.

(1) Phenotype similarity: Disease phenotypes are a reflection of interacting gene products and it has been shown that their similarities are positively correlated with gene functions [Brunner and Van Driel, 2004]. In this work, we adopt the phenotype similarity constructed by a text mining method [Van Driel et al., 2006], which are calculated based on the number of shared MeSH terms appearing in the medical description of diseases in OMIM database.

(2) Disease HPO similarity: The Human Phenotype Ontology (HPO) offers a standardized and controlled vocabulary reflecting phenotypic abnormalities of human diseases, and becomes a comprehensive resource for analyzing human disease phenotypes [Robinson et al., 2008]. The similarity between two diseases is computed based on the two HPO-term sets annotating the two diseases [Deng et al., 2015].

(3) Disease DO similarity: Disease Ontology is another comprehensive ontology representing disease classifications based on their etiology. A directed acyclic graph is first constructed for each disease based on their DO terms and the similarity score between the two diseases is measured based on their graph structures [Li et al., 2011].

As discussed in Sec 2.1, those similarity matrices are incomplete with different

Table 2.2: The statistics of each view of drugs/diseases in dataset.

Drug View	Number	Disease View	Number
Chemical Structure	585	Phenotype	313
Side Effects	412	HPO term	310
Target Profiles	536	DO term	149
Binding Site	467		

degree. For example, although the drug Leuprolide (DrugBank ID: DB00007) is approved by FAD, it is a biotech-type compound and does not have the chemical structure. Similarly, no all of drugs' side-effect are recorded in SIDER dataset. The statistics of each view of drugs and diseases are shown in Table 2.2

2.4.2 Analysis of Multi-view Data

We first investigated the strength of associations among different sources of drugs and diseases, respectively. For drugs, previous studies have shown that similar chemical structures tend to have similar drug-target profiles and drug side-effect profiles [Campillos et al., 2008, Takarabe et al., 2012]. The relationship of chemical structures and ligand binding sites have also been investigated recently [Xie et al., 2009]. We evaluated all pairwise relationships among the four features of drugs: Chemical Structure (CS), Side-Effect (SE), Drug-Target profiles (DT), and Drug-ligand Binding sites (DB). As discussed in Sec 2.4.3, all of the drug features have considerable missing values. We thus first selected instances with complete data for all features, which resulted in total $23,887$ datapoints.

Table 2.3 shows the pairwise correlation coefficients among four drug features. All of the correlation coefficients are positive and statistically significant (t-test, significance level 0.00001). Similarities defined based on drug-target profiles and drug-ligand binding sites have extreme high correlations, which makes intuitive sense. The result about the relationships between drug structures and their target information is consistent with many previous studies. For example, a recent study [Haupt et al., 2013]

Table 2.3: Pairwise relationships among four drug features.

	CS	**SE**	**DT**	**DB**
CS	1.00	0.158	0.361	0.325
SE	0.158	1.00	0.140	0.129
DT	0.361	0.140	1.00	0.796
DB	0.325	0.129	0.796	1.00

Table 2.4: Pairwise relationships for disease features.

	PH	**HPO**	**DO**
PH	1.00	0.427	0.368
HPO	0.427	1.00	0.344
DO	0.368	0.344	1.00

has found that more than 70% of a set of 164 known promiscuous drugs have at least two shared targets with similar binding sites. The same analysis can be also applied to three disease features: Phenotype (PH), HPO terms and DO terms. As shown in Table 2.4, all of the coefficients are also positive and statistically significant. Therefore, it's reasonable for FRMSL to impute missing values by exploiting compatible and complementary information across multiple data sources.

2.4.3 Experimental Results

To systematically evaluate the performance of the proposed FRMSL algorithm, we first performed a five-fold cross-validation experiment on the gold standard dataset mentioned earlier, and compared its results with two recent network based algorithms, as well as the KronRLS algorithm (without missing data imputation). Basically, all known drug-disease interactions in gold standard dataset are randomly divided into five groups in equal size. In each cross validation trial, one group is selected in turn as the test set, while the remaining four groups regards as the training set. After performing by different methods on the test sets, the overall predicted value of test dataset are then compared with its original true labels. Given a specified threshold, true positive rate (TPR) and false positive rate (FPR) are computed respectively to

draw We used the receiver operating curve (ROC) and the area under the ROC curve (AUC) to evaluate the overall performance of different algorithms. Furthermore, using the trained models based on the gold standard dataset, we tested the performance of the four algorithms on an independent dataset. Finally, we showed that FRMSL could be used to detect novel drug-disease associations by utilizing all existing data.

Comparison with Other Methods

To see any improvement due to multi-view learning, we directly compared FRMSL with the KronRLS algorithm. For KronRLS, as a common practice, all the missing columns and rows in drug and disease kernels were filled with zeros. There are many parameters in FRMSL (KronRLS as well). In theory, one may find an optimal set of parameters using the grid search algorithm based on cross-validation experiments. But in this study, we chose these parameters based on prior experience, prior domain knowledge, or some uninformative priors. Although the final results of FRMSL may not be optimal, it still outperforms other algorithms (to be shown), which demonstrates its robustness and generalization power. In the experiments, we set $\lambda = 1$ in Eq. (2.1). For the parameters to fuse different kernels into one in Eq. (2.5), we simply chose the same weight for each kernel (*i.e.*, $W_c^i = 0.25$ for each drug kernel and $w_d^j = 0.33$ for each disease kernel). The same set of parameters were used for KronRLS for a fair comparison. For FRMSL, the regularization parameters $\{\alpha_i\}$ controls the level of consistency between each individual view/kernel and the common structure across all views. We chose them based on domain knowledge. Among the four drug kernels, the chemical structure, drug side-effect and drug binding sites similarity are all drug-centric (either drugs' intrinsic properties or drug-target local interaction property). On the other hand, the target sequence similarities based on global alignments of drug targets (e.g. K_{target}) may only be weakly associated with drugs' therapeutic effects. Therefore, we chose a higher penalty coefficient for G_{target}

($\alpha = 0.2$ for target sequence kernel and $\alpha = 0.1$ for the rest of the three kernels). This effectively forced the network represented by K_{target} to align more closely to other three kernels. For disease kernels, we used the same parameter ($\alpha = 0.1$) for all of them because no additional information about the relationships among phenotype, HPO and DO similarities was available.

Apparently, FRMSL performed much better than KronRLS (Figure 2.2, ROC 0.9295 vs 0.8356). Because the only difference between the two is the data imputation step, our results shows the effectiveness of the multi-view kernel completion approach. We compared with two recent network-based approaches: TL_HGBI [Wang et al., 2014] and MBiRW [Luo et al., 2016]. As mentioned earlier, TL_HGBI is a three-layer heterogeneous network that captures inter- and intra-relationships among drugs, targets and diseases. The target layer serves as a transfer layer to further improve the performance of drug repositioning with additional information on drug-target relationships. MBiRW is a more recent work that uses a two-layer network for drugs and diseases and adopts a Bi-Random walk algorithm. The parameters of TL_HGBI and MBiRW are set as their original paper to obtain the optimal performance. The five-fold cross-validation results (ROC curves and AUC values in Figure 2.2) show that all three algorithms perform well on this dataset. The newly proposed approach FRMSL (AUC: 0.9295) outperforms both TL_HGBI (AUC: 0.8981) and MBiRW (AUC: 0.9031), while the later two have almost identical performance.

Previous studies have found that the links with low drug-drug or disease-disease similarities provide little information for network inferences [van Laarhoven et al., 2011]. All three methods have tried to handle this issue to some extend, but in different ways. For example, TL_HGBI tried to incorporate more information by adding an additional drug-target layer. It was shown that by randomly removing some known drug-target links in the graph, the performance gradually decreased [Wang et al., 2014], which supported the belief that drug-target information contributed to drug-disease predic-

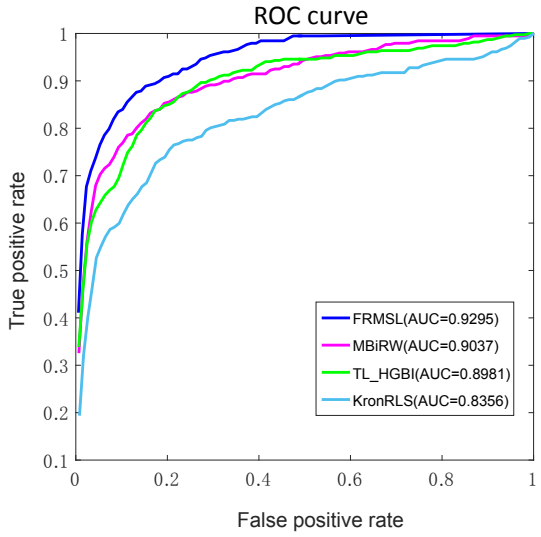

Figure 2.2: The ROC curves and AUC values in predicting drug-disease interactions by FRMSL, MBiRW, TL_HGBI, KronRLS.

tion. However, it is hard to extend the model to include more data sources related to diseases or drugs. MBiRW handled this issue by first clustering drugs and diseases based on common drug indications. And similarities for drugs and diseases were adjusted if they were in the same cluster [Luo et al., 2016]. However, those clusters highly depended on the known drug-disease interactions. Also, although the original paper [Luo et al., 2016] has shown that MBiRW outperformed HGBI, results in the current study have shown that MBiRW and TL_HGBI had almost identical performance. FRMSL addressed this issue by incorporating multiple data sources and by imputing missing values using multi-view learning. It provides a better approach not only because it has better performance, but also because the framework can be easily extended to include more data sources.

Parameter Studies

There are two types of parameters in the proposed FRMSL algorithm: λ and $\{\alpha_i\}$, where λ (in Eq. (2.1)) controls the complexity of the predict function $f(\cdot)$ in the Hilbert space, and α_i (in Eq. (2.6)) controls the disagreement between i-th view's latent factor and the consensus latent factor. Generally speaking, if the i-th view of data is incomplete with high noise, a relative small α_i is then preferred. In the study of these regularization parameters, we set α_i to be the same for all views of data sources for computational convenience. We then vary α_i and λ from $\{0.001, 0.01, 0.1, 1, 10\}$. Figure 2.3 shows the AUC value by using a grid-based search algorithm. FRMSL is relatively stable over a wide range of both λ and α_i, and a relatively high AUC can be achieved when λ is within $[0.1, 1]$ and α_i is inside $[0.01, 0.1]$.

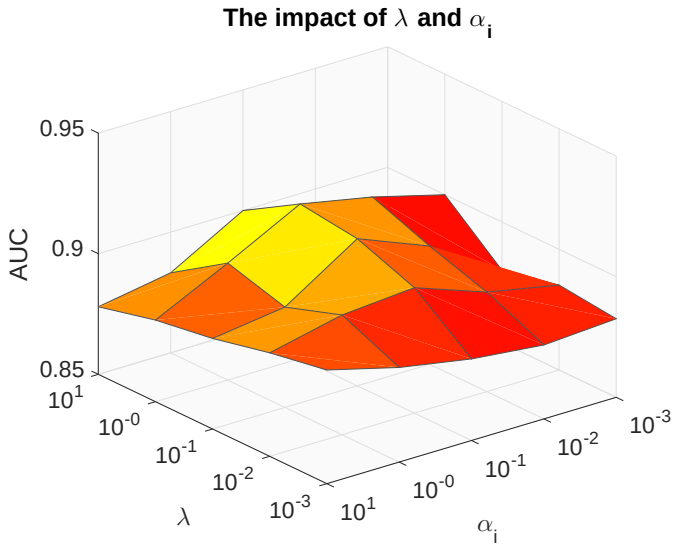

Figure 2.3: Parameter studies by grid-based search algorithm.

2.4.4 Results on an Independent Test Dataset

We also tested the four approaches for drug repositioning on an independent test set released recently [Martínez et al., 2015]. The drug-disease relationships in the new dataset were obtained mainly from recent literatures and KEGG database. After re-

moving the repetitive drug-disease associations that were already in the gold standard dataset, 89 new associations were finally retained. We then utilized them to evaluate the performance of the four different algorithms. The overall results are presented in Table 2.5. Overall, the results on this independent dataset are not as good as the results obtained from cross-validation experiments using the gold standard data set. Nevertheless, FRMSL still outperforms all other methods, which is consistent with ealier cross-validation results. On the other hand, TH_HGBI achieves slightly better results than MBiRW. The KronRLS without imputation still performs the worst. Overall, results on the independent dataset has demonstrated that FRMSL is a promising method for novel drug-disease predictions.

Table 2.5: The AUC values on an independent dataset for FRMSL, MBiRW, TH_HGBI, KronRLS.

Framework	AUC value
FRMSL	0.8526
MBiRW	0.8317
TH_HGBI	0.8405
KronRLS	0.7810

2.4.5 Case Studies

In addition to previous experiments, we further applied FRMSL on all collected data to identify novel drugs for some complex human diseases [Wang et al., 2014]. In this study, we focused our analysis on the five diseases: Non-small-cell lung cancer (NSCLC, OMIM:211980), Alcohol dependence (AD, OMIM:103780), Small-cell lung cancer (SCLC, OMIM: 182280), Human immunodeficiency virus type 1 (HIV-1, OMIM:609423) and Leukemia (LA, OMIM: 608232). Among the five diseases, two of them, HIV-1 and LA had no known drugs in the training data. For NSCLC disease, only one known drug, Doxorubicin (DB00997), had been recorded in training dataset. There were six known drugs (Disulfiram (DB00822), Ondansetron (DB00904), Citalopram (DB00215), Chlordiazepoxide (DB00475), Acamprosate (DB00659), Naltrexone

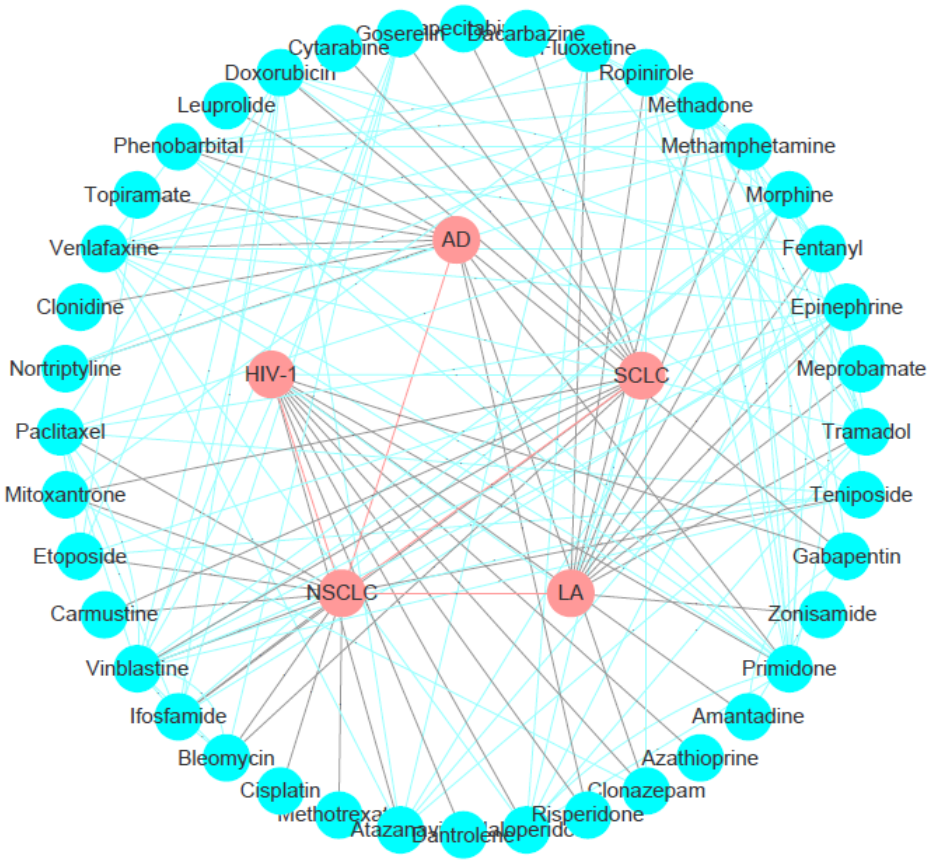

Figure 2.4: Case study results: a subnetwork consists of five diseases and the top 10 predicted drugs for each of the diseases. Drug nodes (blue) and disease nodes (red) are connected by different types of edges (drug-drug edges: blue; disease-disease edges: red; drug-disease edges: black).

Table 2.6: The top 10 novel predictions for NSCLC

Rank	**Drug**	**Drugbank ID**
1	Methotrexate	DB00563
2	Cisplatin	DB00515
3	Bleomycin	DB00290
4	Vincristine	DB00541
5	Teniposide	DB00444
6	Vinblastine	DB00570
7	Carmustine	DB00262
8	Etoposide	DB00773
9	Mitoxantrone	DB01204
10	Paclitaxel	DB01229

(DB00704)) that can treat AD disease and five known drugs (Cisplatin (DB00515), Topotecan (DB01030), Teniposide (DB00444), Methotrexate (DB00563), Etoposide (DB00773))for SCLC disease. We then applied our FRMSL framework to predict novel drug-disease associations that were not recorded in the training dataset. For each disease, the top 10 ranked predictions are presented in Tables 2.6-2.10. The corresponding drug-disease sub-network consisting of the five diseases and their top-10 drugs is shown in Figure 2.4. We further manually searched PubMed to see if any literature supporting our top-10 predictions. Interestingly, many of the predictions can be found in the literature. For example, three of ten drugs predicted for NSCLC can be found in the literature: Cisplatin (PMID: 23349823), Etoposide(PMID: 28137739) and Paclitaxel (PMID: 28304139). For Alcohol dependence, Topiramate (PMID: 28319159) and Gabapentin (PMID: 25969570) have been tested in clinical trials. For SCLC, Vinorelbine and Ifosfamide have been compared with an exist drug (Cisplatin) in terms of safety and efficacy (PMID: 23167406). For the diseases with no known drugs in the training data LA and HIV-1, Methadone (PMID: 23633472) and Atazanavir (PMID: 18722869) were used to treat the two diseases, respectively. The results further demonstrate the effectiveness of FRMSL in predicting novel indications of drugs.

Table 2.7: The top 10 novel predictions for AD

Rank	**Drug**	**Drugbank ID**
1	Primidone	DB00794
2	Gabapentin	DB00996
3	Nortriptyline	DB00540
4	Clonidine	DB00575
5	Venlafaxine	DB00285
6	Clonazepam	DB01068
7	Topiramate	DB00273
8	Phenobarbital	DB01174
9	Risperidone	DB00734
10	Leuprolide	DB00007

Table 2.8: The top 10 novel predictions for SCLC

Rank	**Drug**	**Drugbank ID**
1	Ifosfamide	DB01181
2	Doxorubicin	DB00997
3	Bleomycin	DB00290
4	Vinblastine	DB00570
5	Cytarabine	DB00987
6	Carmustine	DB00262
7	Mitoxantrone	DB01204
8	Goserelin	DB00014
9	Capecitabine	DB01101
10	Dacarbazine	DB00851

Table 2.9: The top 10 novel predictions for LA

Rank	**Drug**	**Drugbank ID**
1	Zonisamide	DB00909
2	Fluoxetine	DB00472
3	Ropinirole	DB00268
4	Methadone	DB00333
5	Methamphetamine	DB01577
6	Morphine	DB00295
7	Fentanyl	DB00813
8	Epinephrine	DB00668
9	Meprobamate	DB00371
10	Tramadol	DB00193

Table 2.10: The top 10 novel predictions for HIV-1

Rank	Drug	Drugbank ID
1	Gabapentin	DB00996
2	Zonisamide	DB00909
3	Primidone	DB00794
4	Amantadine	DB00915
5	Azathioprine	DB00993
6	Clonazepam	DB01068
7	Risperidone	DB00734
8	Haloperidol	DB00502
9	Dantrolene	DB01219
10	Atazanavir	DB01072

2.5 Discussion

Drug repositioning provides a promising alternative for drug discovery. In this article, we have proposed a novel flexible and robust algorithm based on multi-source learning to identify new indications for existing drugs. Distinct kernels are constructed first from multiple data sources to measure the similarities among drugs and among diseases. Furthermore, we have developed a multi-kernel completion algorithm to impute missing values by borrowing information from other data sources. We have also investigated the performance of the proposed algorithm FRMSL by comparing it with several other network-based drug repositioning methods. Both cross-validation experiments and the independent test have shown that FRMSL performs the best among all the tested approaches. Case studies on five human diseases have further demonstrated its effectiveness in practice.

While the results of FRMSL are promising, the current framework can only assess the relationship between a single drug and a single disease. However, many human diseases (especially cancers) are extremely complex, involving multiple metabolic pathways. Therefore, the use of an individual drug, which usually binds to a single target protein, may not be always effective, for example, due to drug resistance. Recently, combinatorial drug therapy has been widely used to overcome drug resistance and to treat complex human diseases such as cancers [Greco and Vicent, 2009].

Drug combinations, consisting of multiple agents, can possibly modulate activities of distinct signaling pathways simultaneously thus maximizing the therapeutic effect. For example, chlorpromazine and pentamidine do not show any anti-tumor activities when used separately in clinical trials, but their combination successfully inhibits the growth of tumor that is more effective than paclitaxel, a common anti-cancer chemotherapy drug [Borisy et al., 2003].

We plan to address the drug combination prediction problem by extending our multiple sources learning framework in the future. One challenge is how to capture the relationships between multiple drugs and a disease in a graphic model. One possible solution is to first cluster drug compounds into different communities according to drugs' properties alone. These communities will be treated as one unit in evaluating their relationships with diseases. Another challenge in drug combination prediction is the administration of optimal doses of different components in treating diseases. It is known that different drug ratios can highly influence the kinetics of drugs [Borisy et al., 2003] and limited training data is available with such information. Therefore, wet lab experimental test is an essential step to validate computational predictions, which cannot be achieved without collaborations with investigators with expertise in clinical medicine and drug discovery.

Appendix

In this part, we provide the detailed proof of Theorem 1 in Sec 2.3.5, following similar ideas in previous work [Lee and Seung, 2001, Liu et al., 2013b, Ding et al., 2006]. We first introduce a matrix inequality proposed in [Ding et al., 2006].

Lemma 2. *For any symmetric matrices* $A \in {R_+}^{n\times n}$, $B \in {R_+}^{k\times k}$, $S \in {R_+}^{n\times k}$ *and*

$S' \in {R_+}^{n\times k}$, the following inequality holds [Ding et al., 2006]:

$$\sum_{ik} \frac{(AS'B)_{ik}S_{ik}^2}{S'_{ik}} \geq Tr(S^T ASB)$$

Proof of Theorem 1 in Sec 2.3.5.

Proof. We define $J(G^{(i)})$ as the sum of all the terms in the objective function in Eq. (2.6) that consist of $G^{(i)}$.

$$\begin{aligned} J(G^{(i)}) = &- 2Tr(W^{(i)^T}W^{(i)}K^{(i)}G^{(i)}G^{(i)^T}) + Tr(W^{(i)}G^{(i)}G^{(i)^T}G^{(i)}G^{(i)^T}W^{(i)^T}) \\ &+ \alpha_i Tr(W^{(i)}G^{(i)}G^{(i)^T}W^{(i)^T}) - 2\alpha_i Tr(G^{*T}W^{(i)^T}W^{(i)}G^{(i)}) \end{aligned} \tag{2.14}$$

We need to find an auxiliary function Z for $J(G^{(i)})$. In other word, $Z(G^{(i)}, G'^{(i)})$ needs to satisfy the following conditions $Z(G^{(i)}, G'^{(i)}) \geq J(G^{(i)})$, $Z(G^{(i)}, G^{(i)}) = J(G^{(i)})$ and $Z(G^{(i)}, G'^{(i)})$ is a convex function of $G^{(i)}$. Therefore, the local optima will become the global optima.

According the inequality $z \geq 1 + logz$, we have

$$\begin{aligned} &Tr(W^{(i)^T}W^{(i)}K^{(i)}G^{(i)}G^{(i)^T}) = \sum_{pqr}(W^{(i)^T}W^{(i)}K^{(i)})_{pr}{G^{(i)}}_{rq}{G^{(i)}}_{pq} \\ &\geq \sum_{pqr}(W^{(i)^T}W^{(i)}K^{(i)})_{pr}{G'^{(i)}}_{rq}{G'^{(i)}}_{pq}\left(1 + log\frac{{G^{(i)}}_{rq}{G^{(i)}}_{pq}}{{G'^{(i)}}_{rq}{G'^{(i)}}_{pq}}\right) \end{aligned} \tag{2.15}$$

Similarly,

$$\begin{aligned} &Tr(G^{*T}W^{(i)^T}W^{(i)}G^{(i)}) = \sum_{pqr}(G^{*T}W^{(i)^T}W^{(i)})_{pr}{G^{(i)}}_{rq} \\ &\geq \sum_{pqr}(G^{*}W^{(i)}W^{(i)})_{pr}{G'^{(i)}}_{rq}\left(1 + log\frac{{G^{(i)}}_{rq}}{{G'^{(i)}}_{rq}}\right) \end{aligned} \tag{2.16}$$

According to the inequality introduced in Lemma 2, we have

$$\begin{aligned} &Tr(W^{(i)}G^{(i)}G^{(i)^T}G^{(i)}G^{(i)^T}W^{(i)^T}) = Tr(G^{(i)}G^{(i)^T}W^{(i)^T}W^{(i)}G^{(i)}G^{(i)^T}) \\ &\leq \sum_{rq} \frac{(W^{(i)^T}W^{(i)}G'^{(i)}G'^{(i)^T})_{rq}(G^{(i)}G^{(i)^T})^2_{rq}}{(G'^{(i)}G'^{(i)^T})_{rq}} \end{aligned} \tag{2.17}$$

and

$$Tr(W^{(i)}G^{(i)}G^{(i)^T}W^{(i)^T}) = Tr(G^{(i)^T}W^{(i)^T}W^{(i)}G^{(i)}) \leq \sum_{rq} \frac{(W^{(i)^T}W^{(i)}G'^{(i)})_{rq}(G^{(i)})^2_{rq}}{(G'^{(i)})_{rq}} \tag{2.18}$$

With the inequalities from Eq. (2.15-2.18). We can define an auxiliary function $Z(G^{(i)}, G'^{(i)})$ for $J(G^{(i)})$ as following

$$\begin{aligned} &Z(G^{(i)}, G'^{(i)}) = -2\sum_{pqr}(W^{(i)^T}W^{(i)}K^{(i)})_{pr}G'^{(i)}{}_{rq}G'^{(i)}{}_{pq}\left(1 + log\frac{G^{(i)}{}_{rq}G^{(i)}{}_{pq}}{G'^{(i)}{}_{rq}G'^{(i)}{}_{pq}}\right) \\ &+\sum_{rq}\frac{(W^{(i)^T}W^{(i)}G'^{(i)}G'^{(i)^T})_{rq}(G^{(i)}G^{(i)^T})^2_{rq}}{(G'^{(i)}G'^{(i)^T})_{rq}} + \alpha_i\sum_{rq}\frac{(W^{(i)^T}W^{(i)}G'^{(i)})_{rq}(G^{(i)})^2_{rq}}{(G'^{(i)})_{rq}} \\ &-2\alpha_i\sum_{pqr}(G^*W^{(i)}W^{(i)})_{pr}G'^{(i)}{}_{rq}\left(1 + log\frac{G^{(i)}{}_{rq}}{G'^{(i)}{}_{rq}}\right) \end{aligned} \tag{2.19}$$

As we can see $Z(G^{(i)}, G'^{(i)}) \geq J(G^{(i)})$ and $Z(G^{(i)}, G^{(i)}) = J(G^{(i)})$. We can then obtain the Hessian matrix $\nabla^2_{G^{(i)}}Z(G^{(i)}, G'^{(i)}) \succ 0$ using a similar idea in [Ding et al., 2006]. We omit the details.

□

Chapter 3

Heterogeneous Network for Drug Combinations

3.1 Introduction

Complex human diseases, such as neurological disorders and cancer, usually involve multiple genes, diverse metabolic pathways, and complex biological processes. Traditional single-drug based treatments are not capable of effectively treating complex human diseases because the majority of drugs are developed to target specific proteins [Barretina et al., 2012, LoRusso et al., 2012, Foucquier and Guedj, 2015, Hu, 2018, Hu et al., 2015]. As an alternative approach, multi-target treatments such as drug combinations, which refer to the use of multiple medications simultaneously to treat a disease, can potentially improve therapeutic efficacy due to drug synergistic effect, an exaggerated response over and beyond additive effects [Lehár et al., 2009, Tsigelny, 2018]. There exist many successful combinatorial therapies in treating complex diseases. For example, Pentamidine and Chlorpromazine show no anti-tumor activities when uses individually, but their combination inhibits tumor growth more effectively than Paclitaxel, an anti-cancer chemotherapy drug [Borisy et al., 2003].

Metformin and Glyburide are both indicated for type 2 diabetes but with different mechanisms: Metformin increases insulin secretion while Glyburide reduces insulin resistance. Their combination can thus improve therapeutic efficacy due to their compensatory mechanisms [Bokhari et al., 2003]. In addition, drug combinations often use FDA-approved drugs and their toxic properties and adverse side effects are thus well studied, and they could be directly used safely by patients without significant risks [Borisy et al., 2003, Foucquier and Guedj, 2015]. Synergistic therapeutic effect, toxicity reduction, and insusceptible to resistance make combinatorial therapies appealing, especially for the development of personalized cancer medicines [Tsigelny, 2018].

Despite the beneficialness of combinatorial therapies, many effective drug combinations are found based on clinical experiences or the test-and-trial strategy. The underlying molecular mechanisms of drug synergies are mostly unclear. Traditional high-throughput screening can be performed to identify potential drug combinations [Lehár et al., 2009], but systematic combinatorial *in vitro* screening remains time-consuming and expensive due to the large combinatorial space of potential combinations. Consequently, researchers have attempted to develop machine learning methods for systematic identification of drug combinations [Tsigelny, 2018]. Existing work mainly collects rich drug-related data (e.g., chemical structures and drugs' target profiles) or disease-related data (disease-specific pathways and disease genes), and discovers drug synergy through supervised learning methods [Pang et al., 2014], network-based models [Huang et al., 2014], and deep learning techniques [Zitnik et al., 2018] (see related work in Section 3.2). However, many successful models have studied drug-related information or disease-related information independently, very few methods attempt to combine these two sources of information. Indeed, the goal of understanding drug synergy is to treat diseases effectively. To model how drug combinations work in disease pathway, we should consider both information of drugs and diseases at the

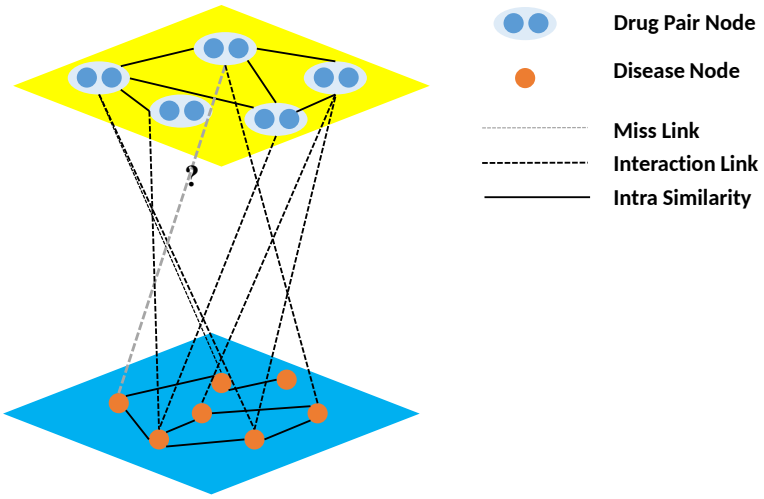

Figure 3.1: A two-layer heterogeneous network for drug combinations, which consisting of drug-pair nodes, disease nodes and edges within and between the two layers. The goal of M CDC is to predict the missing link across the network.

same time.

In this work, we propose a novel **M**ulti-source learning algorithm for **C**omputational **D**rug **C**ombination (**MCDC**) that seamlessly integrates drug-related and disease-related information into a unified framework. Our goal aims to answer the clinical question: which drug pairs could treat which diseases effectively. We investigate the utility of heterogeneous networks to achieve our goal. Our two-layer heterogeneous network consists of two different types of nodes: drug-pair nodes and disease nodes (Figure 3.1). The edges between the same type of nodes are constructed in the form of kernel functions that represent similarity measurements. The edges between different types of nodes across the two layers are constructed based on existing knowledge of drug combinations and diseases. Unlike most existing heterogeneous networks [Martínez et al., 2017], the drug layer of our network is unique since it contains pairs of drugs.

In addition to the network topology, there exist many valuable data sources describing both drugs and diseases, such as drug chemical structures, drugs' target proteins, and disease phenotypes. We further define novel interactions or similarities

within and across layers (e.g., edge weights) according to these rich data sources. By doing so, our network is able to identify more meaningful drug combinations that align more closely with existing medical knowledge. With these definitions, it is not surprising that many current heterogeneous network algorithms [Martínez et al., 2017] can be directly applied to our network. The prediction of drug combinations thus can be formulated as a missing link prediction problem across this two-layer network. A unique characteristic of this framework is that it automatically incorporates both drug information and disease context simultaneously. To address the problem of missing data, we further adopt the idea of multi-view learning [Liu et al., 2013b] by enforcing a common low-rank representation for all views, e.g., drugs' views through collective matrix factorization. Intuitively, missing entities in one view can be inferred by borrowing information from other views. The experiment results demonstrate MCDC can successfully predict drug synergies for different diseases with reasonable accuracy. In a nutshell, MCDC provides a feasible way to integrate multiple omics data to analyze drug synergies in large-scale.

3.2 Related work

Current computational approaches for drug combinations can be roughly categorized into two groups: drug-oriented and disease-oriented [Foucquier and Guedj, 2015]. Drug-oriented approaches, which mostly only utilize drug-related data, could predict potential drug combinations in large-scale [Zhao et al., 2011, Iwata et al., 2015, Chen et al., 2016, Li et al., 2015b, Zitnik et al., 2018, Pang et al., 2014, Xu et al., 2017, Chen and Li, 2019f]. For example, Xu et al. extracted features involving biological and chemical information for each drug combination, and then predicted drug synergistic scores by using a stochastic gradient boosting algorithm [Xu et al., 2017]. Iwata et al. applied a logistic regression to predict beneficial drug combinations by using

drug efficacy and target profiles [Iwata et al., 2015]. Li et al. proposed a probability ensemble approach for the analysis of both the efficacy and adverse effects of drug combinations [Li et al., 2015b]. Recently, Zitnik et al. developed a graph convolutional neural network for drug combinations by incorporating a multi-modal graph of protein-protein interactions, drug-protein target interactions, and the polypharmacy side effects [Zitnik et al., 2018]. However, one drawback of drug-oriented approaches is that the disease contexts are not considered in their frameworks. In precision medicine, it is particularly critical to know which disease(s) the drug combinations can treat accurately.

Disease-oriented approaches, on the other hand, infer drug combinations for a specific disease, relying on disease-related genes or their targets in disease pathways [Huang et al., 2014, Jaeger et al., 2017, Iadevaia et al., 2010, Sun et al., 2015, Preuer et al., 2017]. For instance, Huang et al. prioritized synergistic drug combinations by integrating drug functional networks and disease-specific signaling networks [Huang et al., 2014]. Iadevaia et al. identified optimal drug combinations for breast cancer by utilizing cellular networks [Iadevaia et al., 2010]. Sun et al. combined features of targeting networks and transcriptomic profiles, and validated it on three types of cancers: human b-cell lymphoma, the breast cancer, and lung adenocarcinoma [Sun et al., 2015]. DeepSynergy, a recent deep learning method, used both chemical and genomic information to predict synergy scores of drug combinations for cancer cell lines [Preuer et al., 2017]. However, disease-oriented approaches are applicable only to a specific disease, which can not be easily extended to large-scale discovery studies across different diseases. Finally, existing methods often suffer from the problem of missing data, which is very common in drug combination prediction studies when utilizing large-scale genomic data such as gene expression profiles [Iwata et al., 2015]. Recently, DrugCom[Chen and Li, 2018a], a tensor factorization model, integrated diverse information of drug pairs and disease to infer

drug combinations. Nevertheless, it is not applicable to novel drug pairs since an observed tensor needs to be constructed in advance.

The most closely related work to ours includes two recent DREAM challenges that were launched by the research community to accelerate the understanding of drug synergy. The first DREAM challenge only focused on a single cancer cell line (OCI-LY3) and 14 different compounds. Participants were asked to rank the 91 compound pairs from the most synergistic to the most antagonistic [Bansal et al., 2014]. More recently, the second DREAM challenge released around 11.5k experimentally tested drug combinations measuring cell viability across 85 cancer cell lines for 118 drugs. They also provided both monotherapy and combination therapy drug-response data, which allowed participants to quantify the degree of drug synergy at an unprecedented level [Menden et al., 2019]. Nevertheless, the number of drugs in both DREAM challenges was limited and drugs' names were anonymous, which prevented researchers from integrating existing knowledge of drugs' properties. Moreover, both challenges relied on drug combination screening on a panel of cancer cell lines, which might not easily translate into patients' treatments in the clinic.

3.3 Methods

In this section, we first formulate the computational drug combinations via a two-layer network. We then introduce a kernel-based algorithm to predict missing links across two layers of the network. Furthermore, we develop a multi-view learning framework to impute missing values by integrating multiple data sources.

3.3.1 Problem Formulation

Although drug combinations can consist of more than two drugs leading to a large combinatorial space, only drug pairs are studied in this work. The drug combinations

problem can be viewed as a link prediction problem across a two-layer heterogeneous network. The network consists of two types of nodes (Figure 3.1): drug-pair nodes and disease nodes. A drug-pair node represents a pair of drugs and a disease node represents a single disease. In addition, there are three types of edges: edges between drug-pair nodes, edges between disease nodes, and edges between drug-pair nodes and disease nodes. The edge weights among the same type of nodes represent their similarities. An edge between a drug-pair and a disease represents prior knowledge about the drug-pair in treating the disease: whether it is synergistic or not.

Formally, let $P = \{(p_1^1, p_1^2), (p_2^1, p_2^2), \ldots, (p_n^1, p_n^2)\}$ denote the n drug pair nodes, and $D = \{d_1, d_2, \ldots, d_m\}$ denote the m disease nodes. Let $\mathbf{K_P}((p_i^1, p_i^2), (p_j^1, p_j^2))$ denote the edge weight between (p_i^1, p_i^2) and (p_j^1, p_j^2), and $\mathbf{K_D}(d_i, d_j)$ denote the edge weight between d_i and d_j. The goal is to predict the edge weights between drug-pairs and diseases based on existing training data.

Our network structure is an extension of the structures defined in previous studies [van Laarhoven et al., 2011, Luo et al., 2016, Chen and Li, 2017a], in which the drug layer here consists of drug-pair nodes instead of single drug nodes. The heterogeneous network enables us to tackle the drug combination prediction problem.

3.3.2 Kernel-based Algorithm and Kernel Definitions

Among many link prediction algorithms, we adopt Kernel Regularized Least Squares (KRLS) algorithm to solve the link prediction problem due to its easy implementation [van Laarhoven et al., 2011]. Mathematically, the training data set can be rewritten as $S = \{(\mathbf{x_i}, y_i) : \mathbf{x_i} = \{(p_i^1, p_i^2), d_i\}\}_{i=1}^{l}$. The goal of the KRLS algorithm is to find a decision function $f : X \to Y$, which can predict the label $f(\mathbf{x})$ for a given new sample $\mathbf{x}$, by minimizing the objective function:

$$\underset{f \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{l} \sum_{i=1}^{l} (y_i - f(\mathbf{x_i}))^2 + \lambda \|f\|_{\mathcal{H}}^2 \tag{3.1}$$

where $\|f\|^2_{\mathcal{H}}$ is the norm of the decision function on the Hilbert space; l is the number of training data and λ controls the trade-off between prediction errors and the complexity of the function. The representer theorem guarantees that Eq. (3.1) has a closed-form solution [Scholkopf and Smola, 2001]:

$$f(\mathbf{x}) = \sum_{i=1}^{l} c_i \mathbf{K}(\mathbf{x}, \mathbf{x_i}) \tag{3.2}$$

where $\mathbf{c} \in \mathbb{R}^l$ is a column vector that can be obtained by solving: $(\mathbf{K} + \lambda \mathbf{I})\mathbf{c} = \mathbf{y}$; $\mathbf{I}$ is the identity matrix and $\mathbf{K}(\cdot, \cdot)$ is a pairwise instance kernel matrix, which measures the similarity between any two instances.

We define the pairwise instance kernel $\mathbf{K}(\cdot, \cdot)$ based on the *guilt-by-association* principle [Altshuler et al., 2000]. We assume that when drug-pairs are similar, they tend to have similar synergistic on similar diseases with high probability. To be specific, two instances $\mathbf{x_i} = \{(p_i^1, p_i^2), d_i\}$ and $\mathbf{x_j} = \{(p_j^1, p_j^2), d_j\}$ are similar if their components are similar, *i.e.*, the drug-pair (p_i^1, p_i^2) is similar to (p_j^1, p_j^2) and the disease d_i is similar to d_j at the same time (Figure 3.2). We thus define pairwise instance kernel $\mathbf{K}(\mathbf{x_i}, \mathbf{x_j})$ that directly combines information from both drug-pairs and diseases:

$$\mathbf{K}(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{K_P}((p_i^1, p_i^2), (p_j^1, p_j^2))\mathbf{K_D}(d_i, d_j) \tag{3.3}$$

where $\mathbf{K_P}$ measures the similarity of drug pairs (p_i^1, p_i^2) and (p_j^1, p_j^2), and $\mathbf{K_D}$ measures the similarity of disease d_i and disease d_j. As long as $\mathbf{K_P}$ and $\mathbf{K_D}$ are kernels, the composing kernel $\mathbf{K}$ is also a valid kernel. As illustrated in Figure 3.2, we define $\mathbf{K_P}$ according to its components:

$$\begin{aligned}\mathbf{K_P}((p_i^1, p_i^2), (p_j^1, p_j^2)) =& \mathbf{K_{drug}}(p_i^1, p_j^1) + \mathbf{K_{drug}}(p_i^1, p_j^2) \\ &+ \mathbf{K_{drug}}(p_i^2, p_j^1) + \mathbf{K_{drug}}(p_i^2, p_j^2)\end{aligned} \tag{3.4}$$

where $\mathbf{K_{drug}}(\cdot, \cdot)$ measures the similarities of two individual drugs.

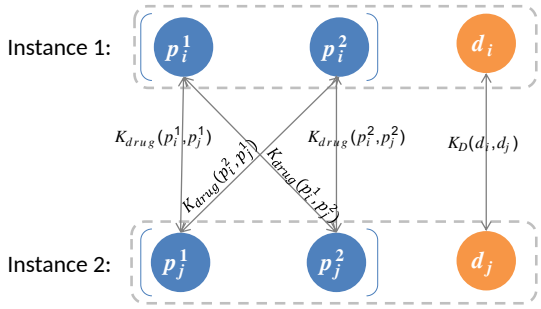

Figure 3.2: A model to construct a pairwise instance kernel between $\{(p_i^1, p_i^2), d_i\}$ and $\{(p_j^1, p_j^2), d_j\}$ based on the components of the two instances. The composing kernel $K(\boldsymbol{x_i}, \boldsymbol{x_j})$ incorporates information from both drugs and diseases.

3.3.3 Kernels Incorporating Multiple Data Sources

The drug kernel $\mathbf{K_{drug}}$ and disease kernel $\mathbf{K_D}$ can be defined in many different ways. For example, drug-drug similarities can be defined based on their chemical structures or their target profiles. Exploiting the complementarity among multi-view data is thus of great importance to understand effect of drug combinations on diseases. In addition, integration of multi-view data provides a powerful way for missing data imputation as discussed later.

From multi-view data, we have a set of drug kernels: $k_{drug} = \{\mathbf{K_{drug}}^{(1)}, \ldots, \mathbf{K_{drug}}^{(n_d)}\}$, and a set of disease kernels: $k_D = \{\mathbf{K_D}^{(1)}, \mathbf{K_D}^{(2)}, \ldots, \mathbf{K_D}^{(m_d)}\}$, where n_d and m_d indicate the numbers of kernels for drugs and diseases, respectively. The optimal drug kernel $\mathbf{K^*_{drug}}$ and disease kernel $\mathbf{K^*_D}$ are defined by simple linear combinations of the individual kernels [Scholkopf and Smola, 2001]:

$$\mathbf{K^*_{drug}} = \sum_{i=1}^{n_d} \omega^i \mathbf{K_{drug}}^{(i)} \qquad \mathbf{K^*_D} = \sum_{j=1}^{m_d} \theta^j \mathbf{K_D}^{(j)} \tag{3.5}$$

The weight ω^i and θ^j can be selected using cross-validation experiments. Alternatively, they can be determined based on prior knowledge of relative importance/reliability of different data sources.

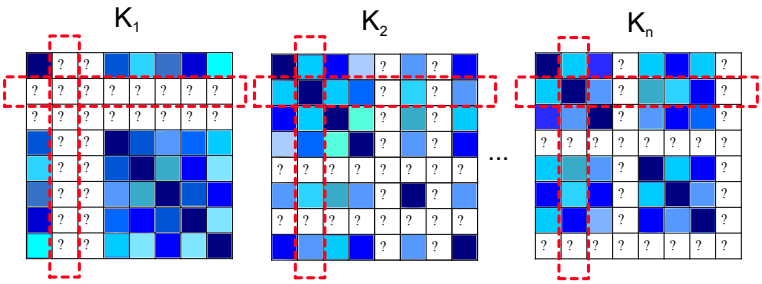

Figure 3.3: N kernel matrices $\{K_1, K_2, \ldots, K_n\}$ constructed from N drug's (or disease's) views. The missing columns or rows in one view could be inferred from other views that contain relevant information.

3.3.4 Multi-view Kernel Completion

As mentioned earlier, missing data is ubiquitous in large-scale integration analysis. For example, one can define drug-drug similarities based on their target protein profiles. However, not all drugs have target information. When a data point has a missing at one attribute, the kernel defined based on that attribute will have a complete row and column missing (Figure 3.3). Replacing missing using zeros in kernel matrices provides computational convenience, but may lead to incorrect predictions [Iwata et al., 2015, van Laarhoven et al., 2011, Nascimento et al., 2016]. Inspired by multi-view learning [Bhadra et al., 2017, Liu et al., 2013b, Chen and Li, 2017b], we propose a collective matrix factorization to impute missing entries by utilizing the hidden complementary relationships among views.

Given a set of drug (or disease) kernels $\{\mathbf{K}^{(i)} \in \mathbb{R}^{N\times N} : 1 \leq i \leq n_v\}$ as shown in Figure 3.3. The idea of MCDC is to jointly factorize all kernel matrices and push all the low-dimensional representations $\mathbf{F}^{(i)}$ towards a consensus matrix $\mathbf{F}^*$ through regularization [Liu et al., 2013b, Chen and Li, 2017b]. The co-training objective function can be written as:

$$\min_{\mathbf{F}^{(i)}\geq 0, \mathbf{F}^*\geq 0} \sum_{i=1}^{n_v} \|\mathbf{K}^{(i)} - \mathbf{F}^{(i)}{\mathbf{F}^{(i)}}^T\|_F^2 + \sum_{i=1}^{n_v} \alpha_i \|\mathbf{F}^{(i)}\mathbf{P}^{(i)} - \mathbf{F}^*\|_F^2 \tag{3.6}$$

where $\mathbf{F}^{(i)} \in \mathbb{R}^{N\times r}$ is the low-dimensional representation for $\mathbf{K}^{(i)}$. $\mathbf{F}^* \in \mathbb{R}^{N\times r}$

is the consensus latent representation shared by all views. The second term is a measure of inconsistency between $\mathbf{F}^{(i)}$ and $\mathbf{F}^*$, and α_i controls its relative degree of inconsistency. $\mathbf{P}^{(i)}$ can be regarded as a scale matrix for $\mathbf{F}^{(i)}$ since different views might not be comparable at the same scale. It can be defined as: $\mathbf{P}^{(i)} = Diag(\sum_u \mathbf{F}^{(i)}_{u,1}, \sum_u \mathbf{F}^{(i)}_{u,2}, \ldots, \sum_u \mathbf{F}^{(i)}_{u,t})$ as suggested by [Liu et al., 2013b]. Moreover, the non-negativity on $\mathbf{F}^{(i)}$ and $\mathbf{F}^*$ improves interpretability of data [Lee and Seung, 2001].

We solve the optimization problem with the objective function in Eq. (3.6) using an alternating scheme, which iteratively minimizes the objective function with respect to one variable while fixing the remaining variables [Lee and Seung, 2001]. The solution of Eq. (3.6) is given as:

$$\mathbf{F}^{(i)}_{pq} \leftarrow \mathbf{F}^{(i)}_{pq} \sqrt{\frac{(2\mathbf{K}^{(i)}\mathbf{F}^{(i)} + \alpha_i \mathbf{F}\mathbf{P}^{(i)^T})_{pq}}{(2\mathbf{F}^{(i)}\mathbf{F}^{(i)^T}\mathbf{F}^{(i)} + \alpha_i \mathbf{F}^{(i)}\mathbf{P}^{(i)}\mathbf{P}^{(i)^T})_{pq}}} \tag{3.7}$$

$$\mathbf{F}^*_{pq} \leftarrow \mathbf{F}^*_{pq} \sqrt{\frac{\left(\sum_{i=1}^{n_v} \alpha_i \mathbf{F}^{(i)}\mathbf{P}^{(i)}\right)_{pq}}{n_v \alpha_i \mathbf{F}^*_{pq}}} \tag{3.8}$$

We alternatively update $\mathbf{F}^{(i)}$ and $\mathbf{F}^*$ until convergence. The convergence can be proved by using the auxiliary function approach [Lee and Seung, 2001]. The optimization algorithm as well as proof details of convergence can be derived like Appendix in Chapter 2. After obtaining the $\mathbf{F}^{(i)}$, the kernel $\mathbf{K}^{(i)}$ can be completed by $\mathbf{K}^{(i)} = \mathbf{F}^{(i)}\mathbf{F}^{(i)^T}$. The complete kernel matrices can then be used in the composite kernels in Eq. (3.3) and Eq. (3.4).

3.4 Experiments and Results

3.4.1 Dataset

To evaluate the proposed MCDC model on real-world datasets, we first download the known drug combinations and their effects on diseases from a comprehensive drug combination database called DCDB[1]. In this study, we mainly focus on drug combinations that only consist of two drugs. In total, 786 pairs of drugs are labeled as "Efficacious", each pair treating with one or more diseases. Diseases in DCDB are encoded by the international classification of diseases system (ICD-10-CM). The disease IDs coded in ICD-10-CM can be mapped to their corresponding OMIM[2] phenotype terms [Garcia-Albornoz and Nielsen, 2015]. Finally, 3556 positive samples in schema $\langle(drug1, drug2), disease\rangle$ are obtained in the experiments, which involves 759 individual candidate drugs and 751 diseases.

Additional information about drugs including drug chemical structures, drug target information (*e.g.*, target protein names, sequences, structures, and annotations), and drug-ligand binding sites are collected from multiple public databases including DrugBank [3], Uniprot [4], and PDB [5]. Information of disease genes was then obtained from the OMIM database.

Following several previous studies [Nascimento et al., 2016, Gottlieb et al., 2011, Chen and Li, 2017a, Yu et al., 2014, Zheng et al., 2013], we define four drug-drug similarity kernels and three disease-disease similarity kernels. The four drug-drug kernels are following: *Drug chemical structures*, *Drug-target protein sequences*, *Target GO annotations*, *Drug-target binding sites*. For diseases, we include three types of disease-disease similarity scores that are based on *Phenotype similarity*, *HPO*

[1] http://www.cls.zju.edu.cn/dcdb/
[2] https://www.omim.org/
[3] https://www.drugbank.ca/
[4] https://www.uniprot.org/
[5] https://www.rcsb.org/

similarity, *DO similarity*.

The statistics of each view of drugs and diseases are list in Table 3.1. From Table 3.1, we observe that each view suffers from different level of missing data. Incorporating multiple sources thus provides compatible and complementary information.

Table 3.1: The statistics of each different view of drugs (number: 779) and diseases (number:751).

Drug View	**Number**	**Disease View**	**Number**
Drug Chemical Structure	665	Disease Phenotype	418
Ligand Binding Site	444	HPO term	621
Target GO term	518	DO term	384
Target Sequences Profiles	518		

3.4.2 Correlations among Different Data Sources

We next assess the correlations among different heterogeneous sources of drugs (or diseases) to have a better understanding on the validity of imputing missing data in one source by borrowing information from other sources. For drugs, previous studies have shown that drugs with similar chemical structures tend to have similar target profiles [Bleakley and Yamanishi, 2009, Huang et al., 2014, Nascimento et al., 2016]. The relationships between drugs' chemical structures and their ligand binding sites have also been investigated [Xie et al., 2009]. Here, we evaluate pairwise relationships among drugs' chemical structures, target proteins (both protein sequences and GO terms), and drug-ligand binding sites. We first choose these instances from the DCDB dataset with complete information from all the four features, which results in $73,827$ data points. Figure 3.4 shows the scatter plots of pairwise relationships of different features of drugs (first 6 subplots).

The pairwise correlation coefficients are also calculated as follows: 0.346 (chemical structures vs. target sequences), 0.304 (chemical structures vs. drug-ligand binding

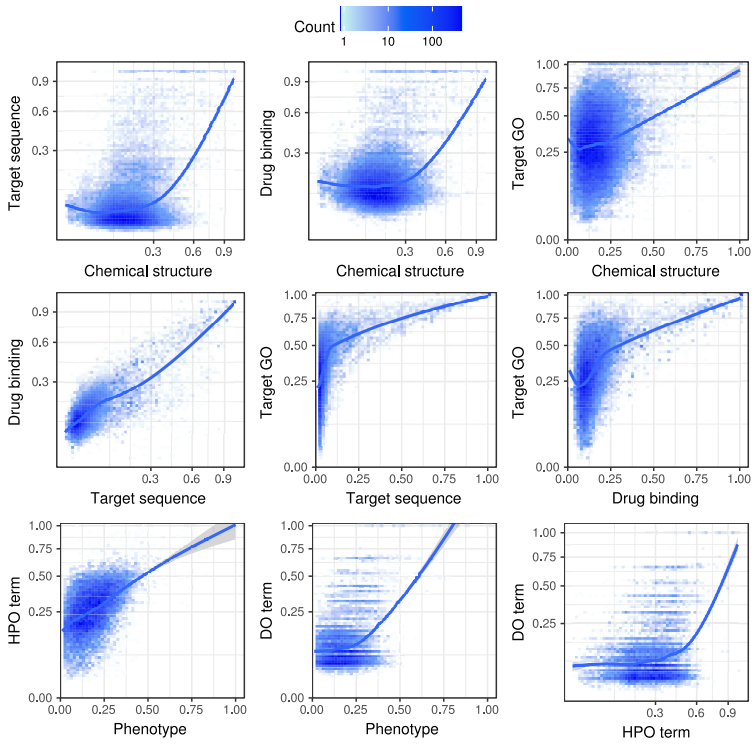

Figure 3.4: The pairwise scatter plots showing correlations among different views of drugs (top-6 subplots) and diseases (bottom-3 subplots).

sites), 0.245 (chemical structures vs. target GO terms), 0.876 (target sequences vs. drug-ligand binding sites), 0.633 (target GO terms vs. target sequences) and 0.599 (target GO terms vs. drug-ligand binding sites). All of the correlation coefficients are positive and statistically significant (t-test, p-value less than 0.0001), indicating the strong correlations among those multiple heterogeneous sources. Similar analysis can be applied to diseases' data sources. Their correlation coefficients are: 0.509 (disease phenotypes vs. HPO terms), 0.351 (disease phenotypes vs. DO terms), and 0.295 (HPO terms vs. DO terms), respectively. The encouraging results suggest that one can impute missing values from different data sources, as proposed in MCDC model.

3.4.3 Experimental Design

To our knowledge, very few existing methods can directly predict the drug combinations for different diseases in a large-scale. However, with our new definition of network topology (Figure 3.1) and its interactions in E.q (3.3)-(3.5), many link prediction algorithms [Martínez et al., 2017, Shi et al., 2017] can be seamlessly mapped to solve drug combination problem. We mainly compared the proposed MCDC model with other four similarity-based methods:

- KRLS, a kernel-based algorithm that is similar to MCDC but without missing data imputation [van Laarhoven et al., 2011].
- Support Vector Machine (SVM), a popular kernel-based supervised learning models for classification [Scholkopf and Smola, 2001].
- MBiRW, a random walk based algorithm that is used to infer missing links on a bipartite network [Luo et al., 2016].
- Graph Regularized Matrix Factorization (GRMF), matrix factorization methods that uses graph regularization to learn low-rank representations for drugs and targets [Ezzat et al., 2017].

The area under the receiver operating characteristic curve (AUC) and the area under the precision recall curve (AUPR) are used to assess the performance of different methods.

We set the regularized parameter $\lambda = 1$ in Eq. (3.1) for both MCDC and KRLS. For the kernel weight parameters in Eq. (3.5), we choose their values based on prior knowledge. Among the four features defining drug similarities, we give higher weights for similarities defined based on drug chemical structures and drugs' target binding sites. It is well known that the chemical structure of a drug is critical in determining the binding of the drug with its targets. The importance of drug chemical

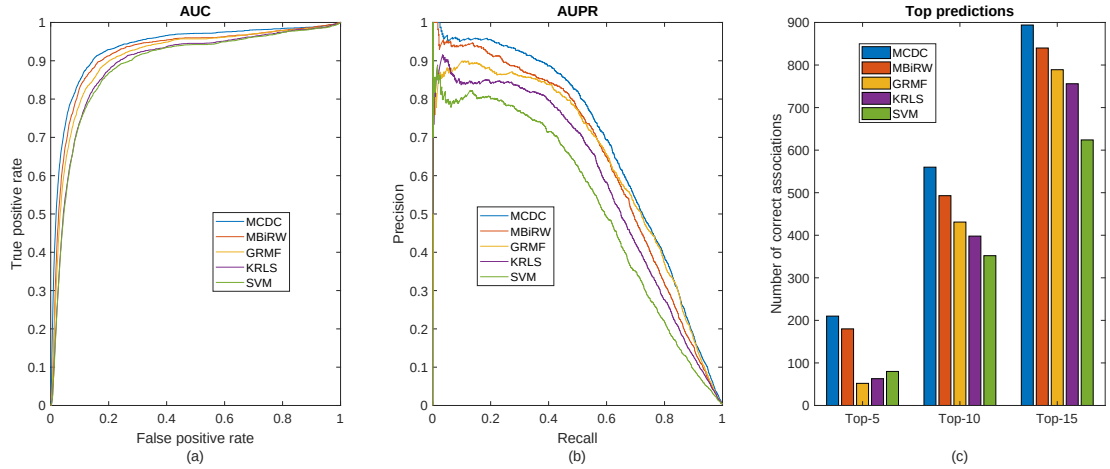

Figure 3.5: (a) AUC (a) and (b) AUPR curves of drug combinations associated with diseases predicted by five different approaches; (c) Number of correctly retrieved known drug pairs for disease associations with various rank thresholds.

structures has long been established in drug design and development (e.g., in quantitative structure-activity relationship (QSAR) studies). Likewise, the local structures of target protein binding sites are also very important because small drug molecules primarily bind and interact with their targets locally. Furthermore, a recent study [Haupt et al., 2013] has found a strong correlation between drug promiscuity and binding site similarities of their targets. We therefore set the weights for similarities based on drug chemical structures and target binding sites to be 0.3, and the weights for similarities based on target protein sequences and GO terms to be 0.2. With respect to disease-disease similarities, no special reference is given to any of the three and they all have the same weight of 0.33.

For the proposed MCDC , we assume missing data in different views are random and utilize the same regularization parameters for each view's kernel matrix ($\{\alpha_i\} = 0.01$ for both drugs and diseases). The impacts of these parameters are studied using the grid search algorithm later. For the other three methods, they can only incorporate information from a single data source, and obviously, cannot impute missing by borrowing information from other data sources. We thus only include the

drug chemical structure and disease HPO terms for those four methods respectively because of least missing ratio in these views. We tune the regularized parameters of all methods by using cross-validation approach and their optimal performances are reported in the experiments. To minimize randomness, we randomly select 20% positive samples as well as an equal amount of unobserved elements as negative samples to be the test dataset, the rest are served as training dataset. The experiments run ten times independently and the average results are reported for all methods.

3.4.4 Experimental Results

Figure 3.5 shows the results of different approaches in terms of AUC and AUPR values. The proposed algorithm MCDC (AUC: 0.931, AUPR: 0.652) consistently outperforms all other approaches including SVM (AUC: 0.871, AUPR: 0.521), KRLS (AUC: 0.894, AUPR: 0.583), GRMF (AUC: 0.901, AUPR: 0.622), and MBiRW (AUC: 0.914, AUPR: 0.631). We have the following observations. First, comparing MCDC with KRLS, the primary difference between the two is that KRLS replaces missing values using zeros for computational convenience, which leads to suboptimal results by KRLS because zeros in similarity matrices indicating two instances are dissimilar. The gain of MCDC implies that the missing information in one view can be inferred effectively using other views. Second, SVM's performance is worse than MCDC, but comparable to KRLS, further confirming that the gain by MCDC is primarily attributable to the missing data imputation step using multi-view learning. Third, MCDC also performs better than MBiRW and GRMF frameworks. The primary difference between the three is that MCDC considers multiple types of similarities for both drugs and diseases while MBiRW and GRMF only include one type. The results demonstrate the benefits of incorporating multiple data sources of drugs and diseases. Previous studies have found that weak drug-drug similarities and disease-disease similarities provide little information for the network inferences [Campillos et al., 2008, Yamanishi, 2014].

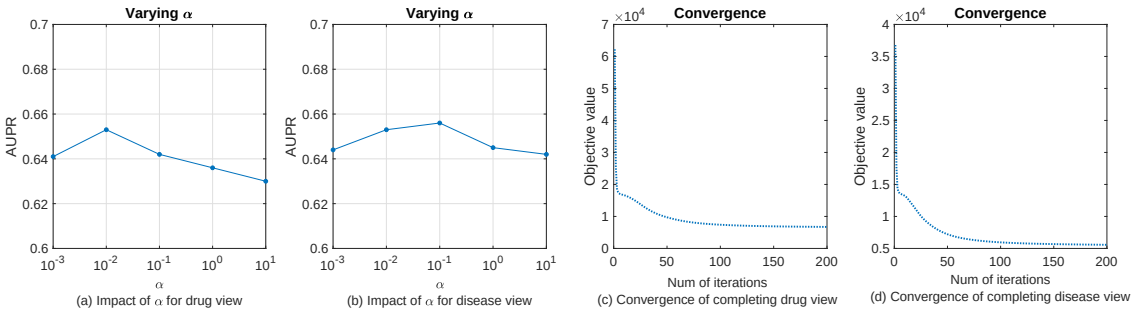

Figure 3.6: (a) The impact of α in drug view; (b) The impact of α in disease view; (c) and (d) Convergence of Eq. (6) in completing drug and disease kernels, respectively.

The final predictions may thus be incorrect when only including single view of drugs and diseases. Another noticeable characteristics of MCDC is that on the very left side of PR curves, MCDC performes much better than other methods, indicating it achieves much higher precisions for same recall rates.

In addition, we test the top-k candidates by different methods similarly to a recent study [Luo et al., 2016]. The number of correctly retrieved drug pairs for disease associations is shown in Figure 3.5(c). For a specified top-ranked threshold, a true drug pair for a disease association is considered as correctly retrieved if the predicted ranking of this association is higher than the specified top-rank threshold. Obviously, MCDC outperforms the other four methods in the top-k prediction.

In summary, with our novel network definition, existing link prediction algorithms can successfully predict the drug synergy for different diseases with a reasonable accuracy. Also, the performance can be improved by incorporating multi-view auxiliary information of drugs and diseases, as shown in the proposed MCDC model.

3.4.5 Impact of Parameters

We then study the sensitivity of MCDC for different regularization parameters $\{\alpha_i\}$ in the data imputation stage. There are two sets of parameters $\{\alpha_i\}$ for drugs and diseases, respectively. Briefly, parameter α_i reflects disagreement between the latent

factor from view i of drugs (or diseases) and the consensus latent factor. A larger α_i will force this two latent factors closer. We first consider the impact of two sets of parameters for drugs while fixing the parameters of diseases ($\{\alpha_i\} = 0.1$). We further limit the search space by setting α_i to be the same for all drug's views. We then varied $\{\alpha_i\} = \alpha$ from $\{0.001, 0.01, 0.1, 1, 10\}$. As shown in Figure 3.6(a-b), MCDC is generally stable over a wide range of α for both drug and disease scenarios. Specifically, a relatively high AUPR can be achieved when α_i is between 0.01 and 0.1.

In addition, we study the convergence of Eq. (3.6) in terms of the number of iterations while completing the kernel matrices. Figure 3.6(c) and 3.6(d) shows the value of the objective function value with respect to the number of iterations. From the figure, we observe the objective function value decreases steadily with more iterations. Usually, less than 80 iterations are sufficient for convergence.

3.5 Conclusion

Drug combinations provide a promising strategy for overcoming drug resistance and can be a great alternative to treat complex human diseases. In this study, we propose a novel network-based model for systematic prediction of possible drug combinations for diseases based on multiple incomplete data sources. The proposed framework not only can take both drug and disease information into account, it also addresses the data incompleteness in large-scale integration analysis. Extensive experiments demonstrate that our method has achieved great prediction performance compared with other similarity-based methods.

A future direction of our work is to include more heterogeneous data, such as drug-side effect information, gene-expression data before and after drug treatments, protein-protein interaction networks, and signaling pathways of diseases. We can easily incorporate such information into our model by introducing a similarity matrix

for each data source, which may help to uncover the comprehensive mechanisms of drug synergy. Another future task is to consider the sequence of drug combinations. Drug synergy may change when two drugs are administrated in different orders. For example, it was shown that applying drug Cisplatin before Taxol was less effective *in vitro* comparing to the alternative order [Rowinsky et al., 1991]. In this case, we can extend the undirected network to a directed network to address this issue. Finally, further wet lab experimental testings are needed to validate the computational predictions.

Chapter 4

Multi-view Tensor Completion for Drug Combinations

4.1 Introduction

Although monotherapy has been successful in treating many human diseases, it suffers from some obvious limitations, such as acquired resistance and/or poor efficiency [Lehár et al., 2009, Borisy et al., 2003]. One of the reasons is that many human diseases are complex, caused by interplay of many factors and regulated by multiple pathways. Therefore, the use of a single drug, which usually targets a single protein, is not capable of treating a complex disease effectively. As an alternative approach, combinatorial drug therapy, which refers to the use of two or more compounds simultaneously to treat a disease, could potentially improve therapeutic efficacy due to its synergistic effect. For example, pentamidine and chlorpromazine show no anti-tumor activities when being administrated individually, but their combination inhibits tumor growth more effectively than paclitaxel, an anti-cancer chemotherapy drug. Moreover, drug combinations often use existing drugs that have been approved by the Food and Drug Administration (FDA). Therefore, their toxic properties and side effects are

usually well studied, and their combination could be directly used safely by patients [Borisy et al., 2003]. Combination therapies, which achieve better efficacy, decrease toxicity, and reduce drug resistance, have thus become a standard for the treatment of several complex disease. Despite the beneficialness of drug combination therapy, drug pairs were often found in clinic empirically or by coincidence. Traditional high-throughput screening was useful to identify some drug pairs [Lehár et al., 2009]. However, it is unrealistic to screen all possible drug combinations due to the large combinatorial space of candidate drug combinations. Therefore, there is a clear need to develop effective computational approaches for systematic identification of drug combinations.

Current computational approaches for drug combinations can be roughly categorized into two groups: drug-oriented and disease-oriented. Drug-oriented approaches, which mostly only utilized drug-related data, could predict potential drug combinations [Zhao et al., 2011, Iwata et al., 2015, Li et al., 2015b, Chen et al., 2019]. For example, Zhao et al. [Zhao et al., 2011] explored various pharmacological features of drug pairs, and then ranked their synergistic scores based on these features. Iwata et al. [Iwata et al., 2015] applied a logistic regression to predict beneficial drug combinations by using drug efficacy and target profiles. However, one limitation of drug-oriented approaches was that the disease context was not considered. In clinics, it is critical to know which disease(s) the drug combinations can treat. Disease-oriented approaches, on the other hand, inferred drug combinations for a specific disease relying on disease-related genes and targets in pathways [Huang et al., 2014, Iadevaia et al., 2010]. Huang et al. [Huang et al., 2014] prioritized synergistic drug combinations by integrating drug functional networks and disease-specific signaling networks. Iadevaia et al. [Iadevaia et al., 2010] identified optimal drug combinations for breast cancer by utilizing cellular networks. However, disease-oriented approaches were applicable only for a specific disease, which could not be easily extended to

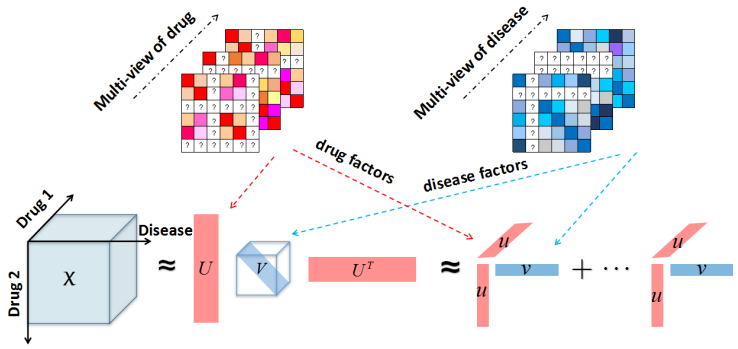

Figure 4.1: Overview of the DrugCom, which utilizes coupled tensor-matrix decomposition to learn the hidden structure of drug×drug×disease relationship and auxiliary information.

large-scale discovery studies across different diseases. Finally, both drug-oriented and disease-oriented methods often suffered from the problem of missing data, which was very common in drug combination prediction studies utilizing large scale genomic data such as gene expression profiles [Iwata et al., 2015]. To overcome these limitations, we propose an effective framework DrugCom, which seamlessly integrates drug-oriented and disease-oriented information into one unified framework. DrugCom (Figure 4.1) first constructs a primary third-order tensor (i.e., drug×drug×disease) and several similarity matrices from multiple data sources of drugs (e.g., chemical structure) and diseases (e.g., disease phenotype). DrugCom then formulates an objective function that simultaneously factorizes coupled tensor and matrices to reveal underlying structures of drug-drug-disease interactions. Doing so allows DrugCom to alleviate the noise and bias contained in each individual data source, therefore to enhance its overall performance.

4.2 Problem Definition

Notations. We denote matrices by boldface uppercase letters (e.g., $\mathbf{A}$) and tensors by boldface caligraphic letters (e.g., $\mathcal{X}$). Let $\mathbf{a}_f$ denote the f-th column of $\mathbf{A}$. More notations of tensor can be found in [Kolda and Bader, 2009].

Although drug combinations can consist of more than two drugs leading to a large combinatorial space, only **drug pairs** are studied in this work. To be specific, let $C = \{c_1, c_2, \ldots, c_N\}$ be a set of N candidate drugs and $D = \{d_1, d_2, \ldots, d_M\}$ be a set of M diseases. The input can be organized as a third-order tensor $\mathcal{X}$ of size $N \times N \times M$ (i.e., drug×drug×disease), where the entries on first two modes of the tensor correspond to the $N \times N$ drug pairs, and the third mode holds M different diseases (Figure 4.1). An entry $\mathcal{X}_{ijk} = 1$ denotes the fact that drug pair (c_i, c_j) can treat disease d_k. Otherwise, the entries are set to 0. In practice, we can only observe parts of the tensor $\mathcal{X}$. We denote this partially observed tensor as $\mathcal{O}$.

We define the *drug combination problem* as follows: given a set of diseases M and a set of drug pairs $N \times N$ with partially observed interactions between diseases and drug pairs in $\mathcal{O}$, our purpose is to predict whether a potential drug pair can treat a certain disease d_i or not. In other words, the goal of drug combination prediction is to solve the Full Tensor Recovery problem to obtain $\mathcal{X}$.

Problem 1. ***Full Tensor Recovery.*** *Given a set of drugs and diseases with only partial observed drug combination treatments (i.e., $\mathcal{O}$), how to recover the full tensor $\mathcal{X}$ in order to obtain new drug combination indications?*

In general, tensor $\mathcal{O}$ is very sparse with a large number of unknown entries. Many studies have shown that in such a case, drug combination predictions can be improved by incorporating additional information from drugs and diseases [Narita et al., 2012, Chen and Li, 2017b, Cao et al., 2017, Ge et al., 2016]. In addition to the main tensor $\mathcal{O}$, there exist many additional data sources for both drugs and diseases, such as

drug chemical structures, drugs' target proteins, and disease phenotypes, representing different views of drugs and diseases. In most cases, one can further summarize such data as drug-drug and disease-disease similarity/kernel matrices to be included in an analysis. Let $\mathscr{A} = \{\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(n_c)}\}$ denote the similarity matrices constructed from n_c views of drugs, and $\mathscr{B} = \{\mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \ldots, \mathbf{B}^{(n_d)}\}$ denote the similarity matrices from n_d views of diseases, both of which are assumed symmetric and non-negative. One great challenge in incorporating multiple heterogeneous data from different sources is data missingness. Because of the nature of data collection (e.g., different datasets were generated by different labs for different purposes at different time), it is very unlikely that all datasets are available for all drugs/diseases. For example, to define drug-drug similarities based on their target protein profiles, not all drugs have target protein information. When a data point has a missing at one attribute, the similarity matrix (or kernel) defined based on this attribute will have a complete row and column missing (Figure 1).The proposed DrugCom will also solve the second problem.

Problem 2. ***Incorporating Existing Knowledge.*** *How existing knowledge of drugs and diseases (i.e., several incomplete similarity matrices from different data sources) can be used in a principled way to alleviate the noise and missing information?*

DrugCom solves above two problems simultaneously by factorizing the main tensor together with multi-view side information in a unified framework. By incorporating existing knowledge of drugs and diseases, DrugCom can effectively identify top drug pairs for each disease.

4.3 Our DrugCom

4.3.1 Recover the Main Tensor

We use tensor factorization to solve the first problem. Let $\mathcal{X}$ denote the drug-drug-disease tensor of size $N \times N \times M$ described in before. DrugCom simultaneously decomposes all the disease slice $\mathcal{X}_k$ (shorthanded of $\mathcal{X}_{(:,:,k)}$) of $\mathcal{X}$ (the third mode) using a rank-r factorization:

$$\mathcal{X}_k \approx \mathbf{U}\widetilde{\mathbf{V}}_k\mathbf{U}^T, \quad \text{for } k = 1, \ldots, M \tag{4.1}$$

where $\mathcal{X}_k \in \mathbb{R}^{N\times N}$ is symmetric and contains all drug pair for disease d_k, $\mathbf{U} \in \mathbb{R}^{N\times R}$ contains the latent-component representation of drugs shared by all diseases, $\widetilde{\mathbf{V}}_k \in \mathbb{R}^{R\times R}$ is diagonal and denotes the latent factor only for disease d_k, R is a user-defined parameter that specifies the number of latent factors. Eq. (4.1) can be interpreted as network clustering of drug pairs in the disease domain by the symmetric matrix factorization approach [Ding et al., 2005]. In particular, by requring the same drug latent factor $\mathbf{U}$, DrugCom simultaneously decomposes the drug-drug networks across all diseases. We further point out that there is a connection between Eq. (4.1) and INDSCAL factorization [Kolda and Bader, 2009], a special case of CP model for third-order tensors that are symmetric in two modes, which is shown in Lemma 3 (the proof is omitted).

Lemma 3. *Equation (4.1) is equivalent to the INDSCAL factorization* $\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{u}_r \circ \mathbf{u}_r \circ \mathbf{v}_r = [\![\mathbf{U}, \mathbf{U}, \mathbf{V}]\!]$ *with factor matrix* $\mathbf{V}(k,:) = diag(\widetilde{\mathbf{V}}_k)$*, where* $[\![\cdot]\!]$ *is a shorthand notation of the sum of rank-one tensors.*

Lemma 3 establishes the equivalence of the objective function in Eq. (4.1) with the CP "slice-wise" model. This equivalence implies that minimizing the objective function in Eq. (4.1) can be achieved by executing the CP decomposition on the tensor $\mathcal{X}$ without optimizing the k independent objective functions in Eq. (4.1), which provides an efficient algorithm to handle large datasets.

4.3.2 Model Side Information

To utilize existing knowledge of drugs and diseases, we use matrix factorization to learn multiple incomplete kernel matrices from multiple datasets [Chen and Li, 2017b]. The key idea is to factorize all views of drugs (diseases) into similar latent factors. Because all the views representing the same entities, they should share some common latent structures. Doing so allows DrugCom to alleviate the noise and missing data contained in each individual data source. Formally, to learn a common underlying latent structure from multiple views of drugs $\mathscr{A} = \{\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(n_c)}\}$, the objective function is

$$\mathcal{J}(\mathbf{U}^{(i)}, \mathbf{U}^*) = \sum_{i=1}^{n_c} (\|\mathbf{A}^{(i)} - \mathbf{U}^{(i)}\mathbf{U}^{(i)^T}\|_F^2 + \|\mathbf{U}^{(i)}\mathbf{Q}^{(i)} - \mathbf{U}^*\|_F^2) \tag{4.2}$$

where $\mathbf{U}^{(i)} \in \mathbb{R}^{N \times R}$ is the drug latent factor for i^{th} view, and $\mathbf{U}^* \in \mathbb{R}^{N \times R}$ is the consensus drug latent structure shared by all views. The first term is the decomposition of each view and the second term is a measure of inconsistency between $\mathbf{U}^{(i)}$ and $\mathbf{U}^*$. The matrix $\mathbf{Q}^{(i)} = \text{diag}\left(\sum_\tau \mathbf{U}^{(i)}_{\tau,1}, \sum_\tau \mathbf{U}^{(i)}_{\tau,2}, \ldots, \sum_\tau \mathbf{U}^{(i)}_{\tau,R}\right)$ is a scale matrix for $\mathbf{U}^{(i)}$ because different views might not be comparable at the same scale when factorizing together [Chen and Li, 2017b]. A joint factorization for all views of diseases can be defined similarly. We leave the details in the unified model.

4.3.3 DrugCom: Optimization Formulation

To combine the tensor decomposition model in Eq. (4.1) with the matrix factorization model in Eq. (4.2), we need to understand the relationship between the drug latent factor $\mathbf{U}$ in Eq. (4.1) and the consensus drug latent factor $\mathbf{U}^*$ in Eq. (4.2) first. In general, the drug pairs in tensor $\mathcal{X}$ can also be regarded as one view of drugs' characteristics, indicating drugs' mechanism of action when treating different diseases. Therefore, it's expected that $\mathbf{U}$ and $\mathbf{U}^*$ should be close to each other because both of them reflect properties of the same drugs. To combine the objective functions of

Eq. (4.1) and Eq. (4.2), we require the two drug factor matrices to be the same. By incorporating disease information as well, we obtain a unified framework, DrugCom, for simultaneous learning of the tensor and similarity matrices from multiple data sources:

$$\min_{\mathcal{X},\mathbf{U},\mathbf{V},\mathscr{U},\mathscr{V}} \{\mathbf{\Phi}(\mathcal{X},\mathbf{U},\mathbf{V},\mathscr{U},\mathscr{V})\}, \qquad s.t. \quad \mathcal{W} * \mathcal{X} = \mathcal{O} \tag{4.3}$$

where $\mathcal{W}$ is a weight tensor with same size as $\mathcal{X}$ and $\mathcal{W}_{ijk} = 1$ if $\mathcal{X}_{ijk}$ is observed; otherwise 0. And

$$\mathbf{\Phi} = \underbrace{\|\mathcal{X} - \mathcal{C}\|_F^2}_{\text{Factorization error}} + \underbrace{\sum_{i=1}^{n_c} \alpha_i (\|\mathbf{A}^{(i)} - \mathbf{U}^{(i)}\mathbf{U}^{(i)^T}\|_F^2 + \|\mathbf{U}^{(i)}\mathbf{Q}^{(i)} - \mathbf{U}\|_F^2)}_{\text{Multiple side information of drugs}}$$
$$+ \underbrace{\sum_{j=1}^{n_d} \beta_j (\|\mathbf{B}^{(j)} - \mathbf{V}^{(j)}\mathbf{V}^{(j)^T}\|_F^2 + \|\mathbf{V}^{(j)}\mathbf{P}^{(j)} - \mathbf{V}\|_F^2)}_{\text{Multiple side information of diseases}}$$

$$\underbrace{\mathcal{C} = [\![\mathbf{U},\mathbf{U},\mathbf{V}]\!] \in \Omega_{\mathcal{C}}}_{\text{INDSCAL factorization}}, \quad \Omega_{\mathcal{C}} = \Omega_{\mathbf{U}} \times \Omega_{\mathbf{U}} \times \Omega_{\mathbf{V}}$$

$$\underbrace{\mathbf{U}, \mathbf{U^{(i)}} \in [0,+\infty)^{N\times R}}_{\text{Nonnegative}}; \qquad \mathbf{V}, \mathbf{V^{(j)}} \in [0,+\infty)^{M\times R}$$

The scale matrix $\mathbf{P}^{(j)}$ for $\mathbf{V}^{(j)}$ can be defined in the same way as $\mathbf{Q}^{(i)}$ before and Ω. denotes the variable domains. The weight parameter α_i and β_j represent the relative strength of i-th view of drugs and j-th view of diseases, i.e., how important $\mathbf{A}^{(i)}$ and $\mathbf{B}^{(j)}$ are in shaping the decomposition of the tensor. The non-negativity constraints on the drug's and disease's latent factors can lead to more interpretable and intuitive results [Kolda and Bader, 2009].

4.3.4 DrugCom: Optimization Algorithm

Many techniques have been proposed to solve the nonnegative tensor optimization problem by the multiplicative update rules [Welling and Weber, 2001]. However, the non-negativity constraints may bring significant computational burden with slow con-

vergence and complicate the development of parallel algorithms to compute large datasets. Also the INDSCAL model in Eq. (4.3) is not as well understood since its factorization is symmetric [Kolda and Bader, 2009]. In this paper, we propose two optimization algorithms. The first one is based on the scaling multiplicative update rules [Welling and Weber, 2001], denoted as DrugCom-MU (The solution is omitted).

A known issue with DrugCom-MU is its low convergence rate. We therefore further develop a more efficient algorithm based on Alternating Direction Method of Multipliers (ADMM), a distributed optimization algorithm that has been shown to perform very well for large-scale tasks [Boyd et al., 2011].

Re-formulation

The objective function $\mathbf{\Phi}$ in Eq. (4.3) is non-convex w.r.t. $\mathbf{U}$, $\mathbf{V}$, $\mathbf{U}^{(i)}$, and $\mathbf{V}^{(j)}$ all together and it involves fourth-order term w.r.t $\mathbf{U}$, $\mathbf{U}^{(i)}$, $\mathbf{V}^{(j)}$, which is hard to optimize directly. Using the variable splitting technique, let $\mathbf{S} = \mathbf{U}$, $\mathbf{C}^{(i)} = \mathbf{U}^{(i)}$ and $\mathbf{D}^{(j)} = \mathbf{V}^{(j)}$, we obtain an equivalent form of $\mathbf{\Phi}$ as follows:

$$\begin{aligned}\mathbf{\Psi} = \|\mathcal{X} - [\![\mathbf{U}, \mathbf{S}, \mathbf{V}]\!]\|_F^2 + \sum_{i=1}^{n_c} \alpha_i (\|\mathbf{A}^{(i)} - \mathbf{C}^{(i)} {\mathbf{U}^{(i)}}^T\|_F^2 + \|\mathbf{U}^{(i)}\mathbf{Q}^{(i)} - \mathbf{U}\|_F^2) \\ + \sum_{j=1}^{n_d} \beta_j (\|\mathbf{B}^{(j)} - \mathbf{D}^{(j)} {\mathbf{V}^{(j)}}^T\|_F^2 + \|\mathbf{V}^{(j)}\mathbf{P}^{(j)} - \mathbf{V}\|_F^2)\end{aligned}$$

The optimization function in Eq. (4.3) can then be reformulated:

$$\begin{aligned} & \min \{\mathbf{\Psi}(\mathcal{X}, \mathbf{U}, \mathbf{S}, \mathbf{V}, \mathscr{U}, \mathscr{V})\} \\ s.t. \quad & \mathbf{S} = \mathbf{U}, \mathbf{S} \in \Omega_{\mathbf{U}},\ \mathcal{W} * \mathcal{X} = \mathcal{O} \\ & \mathbf{C^{(i)}} = \mathbf{U^{(i)}}, i = 1, \ldots, n_c \\ & \mathbf{D^{(j)}} = \mathbf{V^{(j)}}, j = 1, \ldots, n_d \end{aligned} \tag{4.4}$$

where $\mathbf{S}$, $\mathscr{C} = \{\mathbf{C}^{(1)}, \ldots, \mathbf{C}^{(n_c)}\}$ and $\mathscr{D} = \{\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(n_d)}\}$ are the auxiliary variables with respect to $\mathbf{U}$, $\mathscr{U} = \{\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(n_c)}\}$ and $\mathscr{V} = \{\mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(n_d)}\}$.

Optimization Scheme

The partial augmented Lagrangian function for Eq. (4.4) is

$$\begin{aligned}\mathcal{L} = \mathbf{\Psi} + \langle \mathbf{F}, \mathbf{S} - \mathbf{U} \rangle + \frac{\rho}{2}\|\mathbf{S} - \mathbf{U}\|_F^2 + \sum_{i=1}^{n_c}(\langle \mathbf{Y}^{(i)}, \mathbf{C}^{(i)} - \mathbf{U}^{(i)} \rangle + \frac{\eta}{2}\|\mathbf{C}^{(i)} - \mathbf{U}^{(i)}\|_F^2) \\ + \sum_{j=1}^{n_d}(\langle \mathbf{Z}^{(j)}, \mathbf{D}^{(j)} - \mathbf{V}^{(j)} \rangle + \frac{\mu}{2}\|\mathbf{D}^{(j)} - \mathbf{V}^{(j)}\|_F^2)\end{aligned} \tag{4.5}$$

where $\mathbf{F}$, $\mathscr{Y} = \{\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(n_c)}\}$ and $\mathscr{Z} = \{\mathbf{Z}^{(1)}, \ldots, \mathbf{Z}^{(n_d)}\}$ are the Lagrange multipliers. $\langle \cdot, \cdot \rangle$ denotes the inner product of two matrices. $\{\rho, \eta, \mu\}$ are penalty parameters and can be adjusted efficiently according to [Lin et al., 2011]. We solve Eq. (4.4) by successively updating one variable while fixing others until convergence.

Update drug factor U. The terms in the objective function involving $\mathbf{U}$ can be rewritten as

$$\min_{\mathbf{U} \geq \mathbf{0}} \|\mathbf{U}\mathbf{\Pi}_t^T - \mathbf{X}_{(\mathbf{1})}^{\mathbf{t}}\|_F^2 + \sum_{i=1}^{n_c} \alpha_i \|\mathbf{U}_t^{(i)}\mathbf{Q}_t^{(i)} - \mathbf{U}\|_F^2 + \frac{\rho_t}{2}\|\mathbf{U} - \mathbf{S}_t - \frac{\mathbf{F}_t}{\rho_t}\|_F^2 \tag{4.6}$$

where $\mathbf{\Pi}_t = \mathbf{V}_t \odot \mathbf{S}_t$, $\mathbf{X}_{(\mathbf{1})}^{\mathbf{t}}$ is the mode-1 matricization of tensor $\mathcal{X}_t$. Setting the derivatives of Eq. (4.6) w.r.t. $\mathbf{U}$ to zero:

$$\mathbf{U}_{t+1} = (2\mathbf{X}_{(1)}^t\mathbf{\Pi}_t + 2\sum_{i=1}^{n_c} \alpha_i \mathbf{U}^{(i)}\mathbf{Q}^{(i)} + \rho_t\mathbf{S}_t + \mathbf{F}_t)(\widetilde{\mathbf{\Pi}}_t)^{-1} \tag{4.7}$$

where $\widetilde{\mathbf{\Pi}}_t = 2\mathbf{\Pi}_t^T\mathbf{\Pi}_t + 2\sum_{i=1}^{n_c} \alpha_i\mathbf{I} + \rho_t\mathbf{I}$. To implement the non-negative condition in Eq. (4.3), we set the elements of $\mathbf{U}_{t+1}$ to 0 when they are negative. Similarly, we can solve the auxiliary variable $\mathbf{S}$ by minimizing

$$\min_{\mathbf{S} \in \Omega_{\mathbf{U}}} \|\mathbf{S}\mathbf{\Theta}_t^T - \mathbf{X}_{(\mathbf{2})}^{\mathbf{t}}\|_F^2 + \frac{\rho_t}{2}\|\mathbf{S} + \mathbf{F}_t/\rho_t - \mathbf{U}_{t+1}\|_F^2 \tag{4.8}$$

where $\mathbf{\Theta}_t = \mathbf{V}_t \odot \mathbf{U}_{t+1}$, and its closed-form solution is

$$\mathbf{S}_{t+1} = (2\mathbf{X}^{\mathbf{t}}{}_{(2)}\mathbf{\Theta}_t + \rho_t\mathbf{U}_{t+1} - \mathbf{F}_t)(2\mathbf{\Theta}_t^T\mathbf{\Theta}_t + \rho_t\mathbf{I})^{-1} \tag{4.9}$$

Update disease factor V. The terms in the objective function involving $\mathbf{V}$ can be rewritten as

$$\min_{\mathbf{V}\geq\mathbf{0}} \|\mathbf{V}\Xi_t^T - \mathbf{X}^{\mathbf{t}}_{(\mathbf{3})}\|_F^2 + \sum_{j=1}^{n_d} \beta_j \|\mathbf{V}_t^{(j)}\mathbf{P}_t^{(j)} - \mathbf{V}\|_F^2 \tag{4.10}$$

where $\Xi_t = \mathbf{S}_{t+1} \odot \mathbf{U}_{t+1}$. Its close-form solution is

$$\mathbf{V}_{t+1} = (\mathbf{X}^{\mathbf{t}}{}_{(3)}\Xi_t + \sum_{j=1}^{n_d} \beta_j \mathbf{V}_t^{(j)}\mathbf{P}_t^{(j)})(\Xi_t^T\Xi_t + \sum_{j=1}^{n_d} \beta_j \mathbf{I})^{-1} \tag{4.11}$$

Update drug side information factors $\mathbf{U}^{(i)}$. The optimization function for $\mathbf{U}^{(i)}$:

$$\min_{\mathbf{U^{(i)}}\geq\mathbf{0}} \|\mathbf{A}^{(i)} - \mathbf{C}_t^{(i)}\mathbf{U}^{(i)^T}\|_F^2 + \|\mathbf{U}^{(i)}\mathbf{Q}_t^{(i)} - \mathbf{U_{t+1}}\|_F^2 + \frac{\eta_t}{2}\|\mathbf{U}^{(i)} - \mathbf{C}_t^{(i)} - \mathbf{Y}_t^{(i)}/\eta_t\|_F^2 \tag{4.12}$$

Letting the derivatives to be 0, we can get the updating rule

$$\mathbf{U}_{t+1}^{(i)} = (2\mathbf{A}^{(i)^T}\mathbf{C}_t^{(i)} + 2\mathbf{U}_{t+1}\mathbf{Q}_t^{(i)^T} + \eta_t\mathbf{C}_t^{(i)} + \mathbf{Y}_t^{(i)})(2\mathbf{C}_t^{(i)^T}\mathbf{C}_t^{(i)} + 2\mathbf{Q}_t^{(i)}\mathbf{Q}_t^{(i)^T} + \eta_t\mathbf{I})^{-1} \tag{4.13}$$

And the updating rule for its auxiliary variable $\mathbf{C}^{(i)}$ is

$$\mathbf{C}_{t+1}^{(i)} = 2\mathbf{A}^{(i)}\mathbf{U}_{t+1}^{(i)} + \eta_t\mathbf{U}_{t+1}^{(i)} - \mathbf{Y}_t^{(i)})(2\mathbf{U}_{t+1}^{(i)^T}\mathbf{U}_{t+1}^{(i)} + \eta_t\mathbf{I})^{-1} \tag{4.14}$$

Update disease side information factors $\mathbf{V}^{(j)}$. Similarly,

$$\mathbf{V}_{t+1}^{(j)} = (2\mathbf{B}^{(j)^T}\mathbf{D}_t^{(j)} + 2\mathbf{V}_{t+1}\mathbf{P}_t^{(j)^T} + \mu_t\mathbf{D}_t^{(j)} + \mathbf{Z}_t^{(j)})(2\mathbf{D}_t^{(j)^T}\mathbf{D}_t^{(j)} + 2\mathbf{P}_t^{(j)}\mathbf{P}_t^{(j)^T} + \mu_t\mathbf{I})^{-1} \tag{4.15}$$

And the updating equation for its auxiliary variable $\mathbf{D}^{(j)}$ is

$$\mathbf{D}_{t+1}^{(j)} = (2\mathbf{B}^{(j)}\mathbf{V}_{t+1}^{(j)} + \mu_t\mathbf{V}_{t+1}^{(j)} - \mathbf{Z}_t^{(j)})(2\mathbf{V}_{t+1}^{(j)^T}\mathbf{V}_{t+1}^{(j)} + \mu_t\mathbf{I})^{-1} \tag{4.16}$$

Update the Lagrange multipliers F, $\mathbf{Y}^{(i)}$ and $\mathbf{Z}^{(j)}$. We optimize the Lagrange

multipliers using gradient ascent as:

$$\begin{aligned}\mathbf{F}_{t+1} &= \mathbf{F}_t + \rho_t(\mathbf{S}_{t+1} - \mathbf{U}_{t+1}) \\ \mathbf{Y}^{(i)}_{t+1} &= \mathbf{Y}^{(i)}_t + \eta_t(\mathbf{C}^{(i)}_{t+1} - \mathbf{U}^{(i)}_{t+1}) \\ \mathbf{Z}^{(j)}_{t+1} &= \mathbf{Z}^{(j)}_t + \mu_t(\mathbf{D}^{(j)}_{t+1} - \mathbf{V}^{(j)}_{t+1})\end{aligned} \tag{4.17}$$

Update the full tensor $\mathcal{X}$. To perform predictions, $\mathcal{X}_{t+1}$ is computed iteratively as follows:

$$\mathcal{X}_{t+1} = \mathcal{O} + \mathcal{W}^c * [\![\mathbf{U}_{t+1}, \mathbf{S}_{t+1}, \mathbf{V}_{t+1}]\!] \tag{4.18}$$

where $\mathcal{W}^c$ is the complement of $\mathcal{W}$, which is equal to $\mathbf{1} - \mathcal{W}$. The overall procedure is summarized in Algorithm 1.

Complexity Analysis

In Eq. (4.7) , the complexity for updating $\mathbf{U}$ mainly comes from the following matrix operations: $O(MN^2R)$ for computing $\mathbf{X}_{(1)}\mathbf{\Pi}_t$; $O((N+M)R^2)$ for computing $\mathbf{\Pi}_t^T\mathbf{\Pi}_t$ by using property of the Khatri-Rao product $(\mathbf{V} \odot \mathbf{S})^T(\mathbf{V} \odot \mathbf{S}) = \mathbf{V}^T\mathbf{V} * \mathbf{S}^T\mathbf{S}$; $O(R^3)$ for Cholesky decomposition of $\widetilde{\mathbf{\Pi}}_t$ and $O(NR^2)$ for solving the system equation. An analogous estimate can be derived for updating other variables. Overall, the complexity of DrugCom is $O(MN^2R + (N+M)R^2 + R^3)$ in total, noting that $R \ll \min(N, M)$.

4.4 Experiments

4.4.1 Datasets

We first download a dataset from a comprehensive drug combinations database DCDB[1] with data schema (*drug1*, *drug2*, *disease*), which reveals that *drug1* and *drug2* can be combined together to treat a *disease*. In total, there are 759 individual drugs, 751 diseases, and 786 pairs of drugs that are labeled as "Efficacious" for one or

[1]http://www.cls.zju.edu.cn/dcdb/faq1.jsf

Algorithm 1: DrugCom

Input: $\mathcal{O}$, Ω, R, $\{\mathbf{A}^{(i)}\}_{i=1}^{n_c}$, $\{\mathbf{B}^{(j)}\}_{j=1}^{n_d}$, $\{\alpha_i\}$, $\{\beta_j\}$, tol.

Initialize $\mathbf{U}_0$, $\mathbf{V}_0$, $\mathbf{U}_0^{(i)}$, $\mathbf{V}_0^{(j)}$ randomly, set $\mathcal{X}_0 = \mathcal{O}$; $\mathbf{F}_0 = \mathbf{Y}_0^{(i)} = \mathbf{Z}_0^{(j)} = \mathbf{0}$; $\rho_0 = \eta_0 = \mu_0 = 10^{-7}$, $\tau_{max} = 10^{12}$, $a = 1.15$

repeat

  Update $\mathbf{U}_{t+1}$ and $\mathbf{S}_{t+1}$ by Eq. (4.7) and Eq. (4.9)

  Update $\mathbf{V}_{t+1}$ and $\mathbf{F}_{t+1}$ by Eq. (4.11) and in Eq. (4.17)

  for $i \leftarrow 1$ ***to*** n_c **do**

    Update $\mathbf{U}_{t+1}^{(i)}$ and $\mathbf{C}_{t+1}^{(i)}$ by Eq. (4.13) and Eq. (4.14)

    Update $\mathbf{Y}_{t+1}^{(i)}$ in Eq. (4.17)

  end

  for $j \leftarrow 1$ ***to*** n_d **do**

    Update $\mathbf{V}_{t+1}^{(j)}$ and $\mathbf{D}_{t+1}^{(j)}$ by Eq. (4.15) and Eq. (4.16)

    Update $\mathbf{Z}_{t+1}^{(j)}$ in Eq. (4.17)

  end

  Update $\mathcal{X}_{t+1}$ by Eq. (4.18)

  Update parameter $\rho_{t+1} = \min(a * \rho_t, \tau_{max})$ (speed up [Lin et al., 2011]).

  Update parameter $\eta_{t+1} = \min(a * \eta_t, \tau_{max})$.

  Update parameter $\mu_{t+1} = \min(a * \mu_t, \tau_{max})$.

until $\frac{\|\mathcal{X}_{t+1} - \mathcal{X}_t\|_F}{\mathcal{X}_t} \leq tol$

return $\mathcal{X}$

more diseases. Diseases in DCDB are coded using the international classification of disease system (ICD-10-CM), which can be mapped directly to the OMIM[2] disease phenotypes [Garcia-Albornoz and Nielsen, 2015]. Finally, we can construct a binary $759 \times 759 \times 751$ tensor with 3556 positive samples. Existing knowledge about drugs and diseases are collected from multiple online databases including DrugBank[3] for drugs' chemical structures, SIDER[4] for drugs' side effects, and OMIM for disease phenotypes. Following several previous studies [Chen and Li, 2017a, Zheng et al., 2013], we define four drug-drug similarities based on drug's chemical structures, drug's side effects, drug-target profiles, and drug-ligand binding site; three disease-disease similarities based on disease's phenotypes, disease Human Phenotype Ontology (HPO) terms, and Disease Ontology (DO). The statistics of each view of drugs and diseases

[2]https://www.omim.org/

[3]https://www.drugbank.ca/

[4]http://sideeffects.embl.de/

are shown in Table 4.1. The data source about drugs' side effects has the largest number of missing, with 432 (56.9%) out of the 759 drugs having no side effect information in SIDER. For other views of drugs and diseases, the missing rate ranges from 12.3% and 48.8%.

Table 4.1: The statistics of each view of drugs/diseases.

Drug View	**Number**	**Disease View**	**Number**
Chemical Structure	665	Phenotype	418
Side Effects	327	HPO term	621
Binding Site	444	DO term	384
Target Profiles	518		

4.4.2 Experiment Design

We compare DrugCom and DrugCom-MU with several state-of-the-art models that are also based on tensor completion, including (1) CANDECOMP/PARAFAC (CP): a baseline tensor factorization model without considering any auxiliary information of drugs or diseases [Carroll and Chang, 1970]; (2) CMTF: a gradient based tensor completion method by coupling matrix and tensor factorizations [Acar et al., 2011]; (3) TFAI: a tensor analysis method that integrates auxiliary information by within-mode regularization [Narita et al., 2012]; (4) AirCP: another tensor model that integrates auxiliary information using Laplacian regularization [Ge et al., 2016], and (5) t-BNE: a symmetric tensor factorization with orthogonality constraints and Laplacian regularization[Cao et al., 2017].

Notice that the tensor methods CMTF, TFAI, AirCP, and t-BNE can only incorporate information from one data source for each entity, and obviously, cannot impute missing by borrowing information from other data sources. One has to either only include one data source for each entity, or somehow to combine different data sources. We have tried both approaches: 1) using one data source with least missing (chemical structures for drugs and HPO terms for drugs), and 2) using the average

values from all similarity matrices of each entity. The performances of the four models using two different ways of incorporating auxiliary information do not show much differences. Results are only presented for the four models using single data source with least missing.

Another characteristic of the data is that the original observed tensor is very sparse with many unobserved elements. With no doubt, a significant majority of these unobserved elements are actually negative samples. To handle the imbalance of the positive and negative samples, we randomly select an equal number of unobserved elements and treat them as negative samples. Therefore we have the same number of positive and negative samples in the experiments to evaluate the performance of all the approaches. We then randomly select 80% of total data instances as the training dataset and the rest of 20% as the test dataset, and evaluate the performance of all the approaches using Area Under the Precision-Recall curve (AUPR). The dimensionality of the latent factor in all models is set to 30 (i.e., $R = 30$) to have a better tradeoff between computational cost and accuracy. For all the approaches, the regularization parameters are tuned using cross-validation using the training data alone. For t-BNE, the parameters β and γ are both set to 0 because the classification information of drugs or diseases are unknown. The Laplacian regularized terms are both added for drugs and diseases. For DrugCom and DrugCom-MU, the parameters $\{\alpha_i\}$ and $\{\beta_j\}$ are both set to 0.1. The experiments are carried out five times independently and their average results are reported.

4.4.3 Performance Results

Figure 4.2 shows the Precision-Recall curves of all approaches on the DCDB dataset. DrugCom (AUPR: 0.821), as well as DrugCom-MU (0.815), clearly outperforms all other methods (CP: 0.651, CMFT: 0.711, TFAI: 0.704, AirCP: 0.726 and t-BNE: 0.738), showing an improvement of 26.2% to 16.7%. There are several interesting

observations. First, all methods (CMTF, TFAI, AirCP, t-BNE and DrugCom) that consider any auxiliary information consistently perform better than CP model, implying the importance of auxiliary information when data (i.e., the drug×drug×disease tensor) is sparse. Second, both AirCP and t-BNE have similar objective functions with Laplacian regularization. The main difference is that t-BNE utilizes a symmetric factorization of the tensor. For the drug combination prediction problem, t-BNF performs better than AirCP, suggesting that symmetric factorization makes more sense for the drug×drug×disease tensor because the first two modes are symmetric. DrugCom also uses a symmetric factorization, which may also contribute to its superior performance. Third, both DrugCom and DrugCom-MU constantly gain better performance than CMTF, TFAI, AirCP and t-BNE. One of the main reasons is that DrugCom and DrugCom-MU incorporate multiple more compatible and complementary information from multiple data sources, while the others only use one data source. Finally, the performance of DrugCom and DrugCom-MU is similar, implying that different optimization algorithms (e.g., ADMM and MU) for solving Eq. (4.5) are generally consistent with each other. However, as shown in later sections, ADMM is much more efficient than MU. In summary, DrugCom can effectively predict potential drug combinations based on multiple incomplete data sources and has great potential to accelerate the development of compound combinations in drug discovery.

Robustness to Missing Data

As shown in Table 4.2, data missing occurs frequently when integrating multiple data sources. By utilizing multi-view learning, DrugCom can effectively handle missing data.To further test the robustness of DrugCom, we randomly remove some additional data points from each individual view of drugs/diseases by 20%, 30% and 40% (but requiring that information from at least one view is available for each entity when discarding data). We then perform the tensor methods with ability of incor-

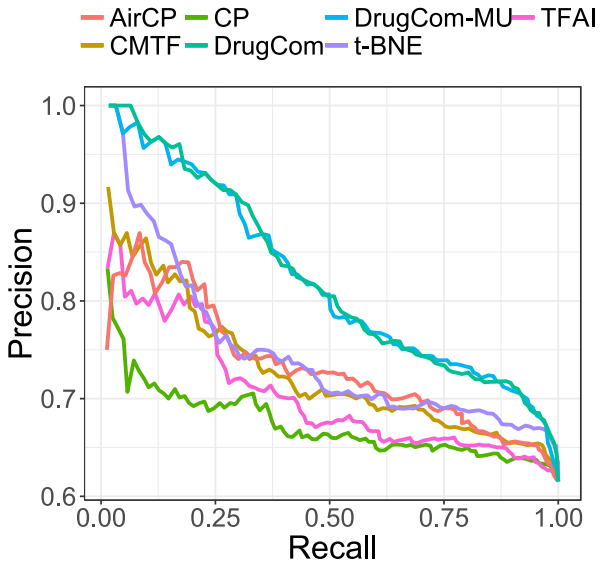

Figure 4.2: The Precision-Recall curves on DCDB dataset.

porating auxiliary information on these datasets. The results are averaged over five independent tests. Table III shows that DrugCom performs well even when the missing rate is as high as 40%. One intuitive interpretation is that the common consensus latent features of drugs/diseases are stable even with a very high missing rate. On the contrary, for tensor methods that can only include one data source for each entity, high missing rates have greater impact on their performance. For example, with 40% missing ratio, all four approaches incorporating a single auxiliary information source actually perform worse than the one without using auxiliary information (CP: 0.651). In summary, DrugCom is resilient to missing data. Therefore it is more generalizable for real-world applications.

Table 4.2: The AUPR values for all approaches when removing additional data at rate of 20%,30% and 40%.

Ratio	CMTF	TFAI	AirCP	t-BNE	DrugCom
20%	0.701	0.672	0.683	0.714	**0.801**
30%	0.643	0.646	0.675	0.676	**0.768**
40%	0.617	0.608	0.617	0.624	**0.759**

4.5 Conclusion

Drug combination therapies are a promising strategy for overcoming drug resistance and can be a great alternative to treat complex human diseases. In this paper, we propose a new tensor model, DrugCom, which can infer beneficial drug combinations for diseases based on multiple incomplete data sources. We formulate the problem as an optimization problem and develop an efficient algorithm. Experimental results on real-world datasets demonstrate the effectiveness and efficiency of the proposed method.

Chapter 5

Learning Drug-Target-Disease Interactions via Tensor Factorization

5.1 Introduction

Targeted therapies and personalized treatments are the most promising approaches to treat complex human diseases such as cancer. Clear understanding of drugs' mechanism of actions (MoAs) is a critical step in drug discovery [Hauser et al., 2017]. Traditional high-throughput screening methods are desirable to identify a new drug against a chosen target (most time a druggable protein) for a special disease. However, the process has been costly and lengthy. A conservative estimate is that it takes $ 2.87 billion and more than 10 years to bring a new drug into market [Lindsley, 2014]. On the other hand, statistical and machine learning models provide an alternative way to accelerate the process of understanding drugs' MoAs by mining bioinformatics, cheminformatics, and Web data sources [Paul and Dredze, 2013, Li et al., 2015a, Campillos et al., 2008, Santos et al., 2017, Althouse et al., 2015, Ezzat et al., 2018,

Araujo et al., 2017]. Learning tasks have been proposed from different angles, such as exploring drug behaviors [Zitnik et al., 2018, Eguale et al., 2016, Sarker and Gonzalez, 2015], assessing target activities [Santos et al., 2017, Gonzalez et al., 2007, Radivojac et al., 2013], and understanding disease models [Zou et al., 2018, Zhang et al., 2017, Perra et al., 2011]. Two of the most prominent formulations among these recent advancements are the drug-target [Ezzat et al., 2018] prediction and drug-disease prediction, also known as drug repositioning [Li et al., 2015a]. In these two tasks, researchers have attempted to collect a variety of omics data from scientific literature and online Web sources, and discover new interactions between drugs and targets/diseases through different statistical models [Zitnik et al., 2018, Lewis et al., 2011, Chen and Li, 2017a, Ray et al., 2016]. For example, a network-based inference method was proposed to infer new targets for known drugs by leveraging the drug-target bipartite network [Cheng et al., 2012]. The method Decagon modeled drug polypharmacy side-effect via graph convolutional networks [Zitnik et al., 2018]. Another approach, PREDICT, integrated multiple drug-drug and disease-disease similarities to infer potential drug-disease interactions [Gottlieb et al., 2011]. Approaches based on multiple kernel learning were developed to predict drug-target [Nascimento et al., 2016] or drug-disease [Chen and Li, 2017a] relationships, by integrating additional heterogeneous information sources.

However, despite the obvious connections, most existing work treat drug-disease and drug-target predictions as two independent tasks, which is viewed as a major shortcoming. Indeed, the therapeutic effect of drugs on a disease is through their abilities to bind and to modulate biological targets that involve the disease pathways, which in turn promote healthy function of the metabolic system and cure the disease [Hauser et al., 2017]. Therefore, instead of a binary relationship such as drug-disease or drug-target, a strong triple relationship drug-target-disease should be considered to better model their interplay. In addition, the growing availability of new types of data on the web (Figure 5.1) brings new opportunities to learn a more com-

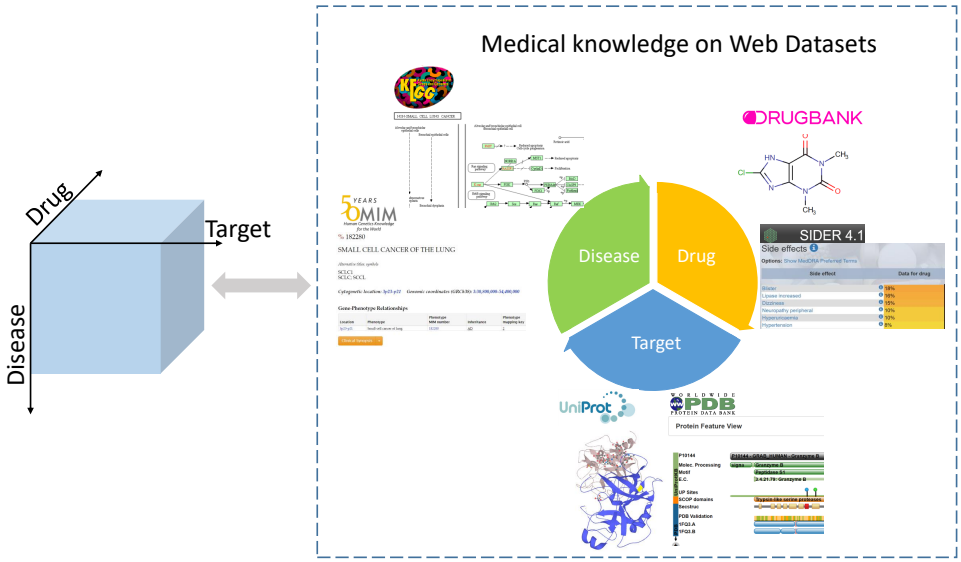

Figure 5.1: Illustration of DTD. Our model jointly explores the $drug \times target \times disease$ tensor along with rich existing medical knowledge on the Web.

prehensive relationship among drugs, targets, and diseases [Paul and Dredze, 2013, Zheng et al., 2013, Nascimento et al., 2016, Chen and Li, 2018a, Althouse et al., 2015, Araujo et al., 2017]. Incorporating such heterogeneous information can significantly improve our understanding of the underlying biological processes. For example, joint analysis of drugs' chemical structures, drugs' side-effects, and protein-protein networks can improve success rates of finding novel drug-target interactions [Gottlieb et al., 2011, Zheng et al., 2013, Nascimento et al., 2016].

In this work, we propose a tensor completion method, termed **DTD**, to model **D**rug-**T**arget-**D**isease interactions, with the help of existing data from the Web. DTD explicitly learns a third-order $drug \times target \times disease$ tensor using Tucker Decomposition [Kolda and Bader, 2009]. In addition, to alleviate the data sparsity issue, DTD incorporates multiple auxiliary information sources of drugs, targets, and diseases. For example, in Figure 1, in addition to $drug \times target \times disease$ tensor, there are rich data on the Web describing drugs (e.g., chemical structures in DrugBank

and side-effects in SIDER database), targets (e.g., protein sequence in Uniprot), and diseases (phenotype description in OMIM or KEGG). The tensor and multiple feature matrices are coupled in the "drug", "target", and "disease" mode, respectively. These feature matrices from Web sources are very useful to learn extra static information about each mode of the tensor. Fusing those datasets together can lead to better interpretations of the complex biological processes. To achieve this goal, DTD further explores the correlations among the latent matrices from the tensor and these multiple data sources via a coupled tensor-matrix factorization. This ensures that knowledge in the tensor aligns more closely with existing medical knowledge of each of the entities.

Another critical challenge is how to solve the tensor completion problem effectively. Because of the nonlinear and non-convex orthogonality constraints in the tensor Tucker model, the solutions of the popular HOSVD and HOOI Euclidean solvers are not unique, which makes it hard to couple with auxiliary information. Motivated by the fast growing Riemannian optimization [Absil et al., 2009, Zhang et al., 2016], we cast the coupled tensor-matrix factorization problem as a nonlinear program with the factor matrices constrained to the Grassmann manifold. Rather than a special non-uniqueness solution in Euclidean solvers, DTD obtains an equivalence class of matrices on Grassmann manifold, which leads to more meaningful subspace representations of factor matrices and can be well coupled with auxiliary information. Moreover, empirical studies have shown that nonlinear Riemannian solvers are significantly faster comparing to the Euclidean solvers [Kasai and Mishra, 2016, Zhang et al., 2016].

5.2 Background and Task Description

5.2.1 Tensor Algebra

We follow the notations introduced by Kolda and Bader [Kolda and Bader, 2009]. *Tensors* are multidimensional arrays that extend the concept of matrices. The *order* of a tensor is the number of its dimensions, also known as ways or modes. A *fiber* is a vector extracted from a tensor by fixing every index but one. A *slice* is a matrix extracted from a tensor by fixing all but two indices. Note that an N-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ reduces to a vector when $N = 1$, and a matrix when $N = 2$. The $(i_1, \ldots, i_N)$-th element of $\boldsymbol{\mathcal{X}}$ is denoted as $\boldsymbol{\mathcal{X}}_{i_1,\ldots,i_N}$. *Matricization*, also known as unfolding or flattening, is the process of reordering the elements of a tensor into a matrix. The mode-n matricization of an N-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is represented as $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 \ldots I_{n-1} \times I_{n+1} \ldots I_N}$ and is arranging the mode-n fibers of the tensor as columns of the long matrix. The *n-mode matrix product* of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ is denoted as $\boldsymbol{\mathcal{X}} \times_n \mathbf{U}$ with size of $I_1 \cdots \times I_{n-1} \times J \times I_{n+1} \cdots \times I_N$. We also give the definition of Tucker decomposition and coupled tensors for third-order tensors.

Definition 1. ***(Tucker Decomposition)****. The Tucker decomposition is a form of higher-order PCA. It decomposes a tensor into a core tensor multiplied by a matrix along each mode. For a three-order tensor* $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$*, its Tucker decomposition is*

$$\boldsymbol{\mathcal{X}} \approx \boldsymbol{\mathcal{G}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$$

where $\mathbf{A} \in \mathbb{R}^{I \times R_1}$*,* $\mathbf{B} \in \mathbb{R}^{J \times R_2}$ *and* $\mathbf{C} \in \mathbb{R}^{K \times R_3}$ *are the factor matrices (which are usually orthogonal) and can be regarded as the principal components in each mode. The tensor* $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ *is the core tensor and captures interactions between factor matrices.*

Definition 2. ***(Coupled Tensors)***. *If a tensor shares one or more modes with other matrices or other tensors, then they can be coupled with one another. For example, in a recommender system, a triple relationship user* $\times$ *movie* $\times$ *review tensor and a user* $\times$ *movie rating matrix can be coupled on the shared user and movie modes.*

5.2.2 Task Description

Our aim is to infer potential drug-target-disease interactions for rational drug repositioning. We formulate the task as a tensor completion problem. To be specific, the input can be organized as a three-way *drug* $\times$ *target* $\times$ *disease* tensor $\boldsymbol{\mathcal{X}}$ of size $n_1 \times n_2 \times n_3$, in which n_1, n_2, and n_3 denote the number of drugs, targets, and diseases, respectively. An entry $\boldsymbol{\mathcal{X}}_{ijk} = 1$ if drug i binds to target j and treats disease k. Otherwise, the entries are set to 0. In practice, we can only observe part of the tensor $\boldsymbol{\mathcal{X}}$ and this partially observed tensor is denoted as $\boldsymbol{\mathcal{T}}$. Our goal is to predict the potential concurrences of drug-target-disease, which can be achieved by completing tensor $\boldsymbol{\mathcal{X}}$ given the incomplete tensor $\boldsymbol{\mathcal{T}}$.

In real-world applications, tensor $\boldsymbol{\mathcal{T}}$ is often sparse with a large number of unknown entries. Recovering tensor $\boldsymbol{\mathcal{X}}$ is challenging when relying only on the observed tensor $\boldsymbol{\mathcal{T}}$. Fortunately, for *drug* $\times$ *target* $\times$ *disease*, there exist many additional data sources on the Web to describe drugs, targets, and diseases. For example, for most drugs, their chemical structures and side-effects can be obtained from online databases, which represent different views of drugs. Datasets regarding targets and diseases are also available online. In most cases, one can further summarize such auxiliary data as drug-drug, target-target, and disease-disease similarity/kernel matrices, which can be incorporated in the learning process. Formally, let $\mathscr{S}_A = \{\mathbf{S}_A^{(1)}, \ldots, \mathbf{S}_A^{(n_a)}\}$ denote the similarity matrices constructed from n_a views of drugs. $\mathscr{S}_B = \{\mathbf{S}_B^{(1)}, \ldots, \mathbf{S}_B^{(n_b)}\}$ and $\mathscr{S}_C = \{\mathbf{S}_C^{(1)}, \ldots, \mathbf{S}_C^{(n_c)}\}$ are defined similarly from n_b views of targets and n_c views of diseases, respectively. All of them are assumed to

be symmetric and non-negative. The details of their constructions will be presented in Sec. 5.5. The proposed DTD framework jointly explores the main tensor $\boldsymbol{\mathcal{X}}$ together with multi-view auxiliary information $\mathscr{S}_A$, $\mathscr{S}_B$ and $\mathscr{S}_C$ to predict more meaningful interactions of drug-target-disease.

5.3 The DTD Model

In this section, we develop the novel DTD model, a simple but effective coupled tensor-matrix factorization to show how different data sources can be included in a principled way. Notation used throughout the paper is provided in Table 5.1.

Table 5.1: Main Notation

Symbol	Description
$\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{T}}$	Full recovery tensor and observed tensor
$\boldsymbol{\mathcal{G}}$	Core tensor in Tucker model
$\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$	Drug, target, disease factor matrix
$\mathbf{L}_A^{(i)}$, $\mathbf{A}^{(i)}$	Laplacian matrix and its spectral embedding from i-th view of drug auxiliary information
$\mathbf{L}_B^{(j)}$, $\mathbf{B}^{(j)}$	Laplacian matrix and its spectral embedding from j-th view of target auxiliary information
$\mathbf{L}_C^{(k)}$, $\mathbf{C}^{(k)}$	Laplacian matrix and its spectral embedding from k-th view of disease auxiliary information

5.3.1 Recover the Main Tensor

To complete the tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we adopt the Tucker Decomposition (Definition 1), which can be represented by the following optimization problem:

$$\min_{\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{G}},\mathbf{A},\mathbf{B},\mathbf{C}} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}\|_F^2 \quad s.t. \quad \mathcal{P}_\Omega(\boldsymbol{\mathcal{X}}) = \mathcal{P}_\Omega(\boldsymbol{\mathcal{T}}) \tag{5.1}$$

where $\|\cdot\|_F$ is the Frobenius norm. Recall that $\boldsymbol{\mathcal{G}}$ is the core tensor; $\mathbf{A} \in \mathbb{R}^{n_1 \times r_1}$, $\mathbf{B} \in \mathbb{R}^{n_2 \times r_2}$ and $\mathbf{C} \in \mathbb{R}^{n_3 \times r_3}$ are the factor matrices with respect to drug, target and disease mode. Ω is the set which contains the indices of observed entities and $\mathcal{P}_\Omega(\cdot)$ keeps the

entries in Ω and zeros out others [Wang et al., 2015, Kasai and Mishra, 2016]. The equality constraint ensures that the corresponding elements of the recovering tensor $\boldsymbol{\mathcal{X}}$ should match with these observed elements in tensor $\boldsymbol{\mathcal{T}}$.

In addition, the factor matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are orthogonal matrices, i.e., $\mathbf{A}^T\mathbf{A} = \mathbf{I}$ and $\mathbf{I}$ is the identity matrix. The orthogonal constraints indicate that the matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are well defined on the so-called *Stiefel manifold* [Absil et al., 2009], which is defined as

$$St(m, n) = \{\mathbf{U} \in \mathbb{R}^{n\times m} | \mathbf{U}^T\mathbf{U} = \mathbf{I}_m\}$$

which contains a set of $n \times m$ orthonormal matrices. We will see the advantage of Stiefel manifold when coupled with the auxiliary information.

5.3.2 Coupled with Auxiliary Information

To incorporate multi-view auxiliary information, we adopt the idea of spectral clustering, due to its flexibility and ease of implementation [Ng et al., 2002, Lu et al., 2016]. Before going on, we give a brief introduction of spectral clustering. Suppose $\mathbf{S} \in \mathbb{R}^{N\times N}$ is the similarity/affinity matrix for N objects where $\mathbf{S}_{ij}$ measures the similarity between object i and object j. One can then compute the normalized Laplacian matrix: $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}$ in which $\mathbf{D}$ is the diagonal matrix with $\mathbf{D}_{ii} = \sum_{j=1}^{N} \mathbf{S}_{ij}$. The spectral clustering is to solve the following optimization:

$$\min_{\mathbf{U}\in\mathbb{R}^{N\times k}} \quad \langle \mathbf{U}\mathbf{U}^T, \mathbf{L}\rangle \quad s.t. \quad \mathbf{U}^T\mathbf{U} = \mathbf{I} \tag{5.2}$$

where $\langle\cdot,\cdot\rangle$ denotes the matrix inner product. $\mathbf{U}$ can be regarded as the low-dimensional spectral embedding of N objects. For clustering task, the k-means algorithm can be then applied to $\mathbf{U}$ to get the clustering indicators. Furthermore, the orthogonal constraint indicates that the spectral embedding $\mathbf{U}$ is also well defined on the Stiefel manifold. Therefore, it is reasonable to jointly consider the factor matrices

in Eq. (5.1) together with the spectral embedding in Eq. (5.2) when performing the coupled tensor-matrix factorization in the sense that both of embeddings are on the Stiefel manifold.

For drug mode of tensor $\boldsymbol{\mathcal{X}}$, considering drug factor matrix $\mathbf{A} \in St(r_1, n_1)$ and its multi-view auxiliary information $\mathscr{S}_A = \{\mathbf{S}_A^{(1)}, \ldots, \mathbf{S}_A^{(n_a)}\}$, we extend the single-view spectral embedding to the follow multi-view co-training optimization function as:

$$\min_{\mathbf{A}, \mathbf{A}^{(i)} \in St(r_1, n_1)} \sum_{i=1}^{n_a} (\langle \mathbf{A}^{(i)} \mathbf{A}^{(i)^T}, \mathbf{L}_A^{(i)} \rangle + \|\mathbf{A}^{(i)} \mathbf{A}^{(i)^T} - \mathbf{A}\mathbf{A}^T\|_F^2) \tag{5.3}$$

where $\mathbf{L}_A^{(i)}$ is the normalized Laplacian matrix of $\mathbf{S}_A^{(i)}$ and $\mathbf{A}^{(i)}$ is the corresponding spectral embedding. The key idea behind Eq. (5.3) is that all the spectral embeddings $\mathbf{A}^{(i)}$ $(0 \leq i \leq n_a)$ from auxiliary information should be close to the drug factor $\mathbf{A}$ since they all represent the same drugs. We achieve this by minimizing the disagreements $d(\mathbf{A}^{(i)}, \mathbf{A}) = \|\mathbf{A}^{(i)} \mathbf{A}^{(i)^T} - \mathbf{A}\mathbf{A}^T\|_F^2$. The reason for choosing $d(\mathbf{A}^{(i)}, \mathbf{A})$ is two-fold: (i) the reconstruction of similarity/kernel matrix $\mathbf{A}^{(i)} \mathbf{A}^{(i)^T}$ from spectral embedding is expected to be consistent with similarity $\mathbf{A}\mathbf{A}^T$ from tensor factor matrix, which is our assumption. (ii) we later show that the joint optimization is further defined on the Grassmann manifold, the quotient space of Stiefel manifold [Absil et al., 2009]. $d(\mathbf{A}^{(i)}, \mathbf{A})$ exactly measures the *geodesic distance* between two Grassmannian points $\mathbf{A}^{(i)}$ and $\mathbf{A}$ [Edelman et al., 1998].

The same analysis can be applied to the target and disease factor matrices $\mathbf{B}$ and $\mathbf{C}$ with auxiliary information $\mathscr{S}_B$ and $\mathscr{S}_C$. We leave the details in the unified model in Eq. (5.4) later.

5.3.3 The Overall Model

Now, we propose the coupled tensor-matrix factorization model by combining the loss function in Eq. (5.1) and Eq. (5.3). Moreover, as pointed out by recent work [Ng et al., 2002,

Lu et al., 2016], in the ideal case, the new affinity/similarity matrix $\mathbf{A}^{(i)}\mathbf{A}^{(i)^T}$ implies the true membership of data cluster and it is naturally *sparse*. The sparse property also holds for $\mathbf{A}\mathbf{A}^T$ in tensor.

Our DTD model seeks for a better representation of drug by further adding l_1-norm on $\mathbf{A}\mathbf{A}^T$ [Lu et al., 2016]. As the matrix $\mathbf{A}\mathbf{A}^T$ is sparse, the spectral embeddings $\mathbf{A}^{(i)}\mathbf{A}^{(i)^T}$ $(0 \leq i \leq n_a)$ from auxiliary information will also encourage to be sparse because of the geodesic distance measurement $d(\mathbf{A}^{(i)}, \mathbf{A})$. The same process holds for target's and disease's factor matrices. Therefore, DTD formulates a joint optimization problem as:

$$
\begin{aligned}
\min f = & \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}\|_F^2 + \rho \cdot (\|\mathbf{A}\mathbf{A}^T\|_1 + \|\mathbf{B}\mathbf{B}^T\|_1 + \|\mathbf{C}\mathbf{C}^T\|_1) \\
& + \alpha \cdot \sum_{i=1}^{n_a} (\langle \mathbf{A}^{(i)}\mathbf{A}^{(i)^T}, \mathbf{L}_A^{(i)} \rangle + \|\mathbf{A}^{(i)}\mathbf{A}^{(i)^T} - \mathbf{A}\mathbf{A}^T\|_F^2) \\
& + \beta \cdot \sum_{j=1}^{n_b} (\langle \mathbf{B}^{(j)}\mathbf{B}^{(j)^T}, \mathbf{L}_B^{(j)} \rangle + \|\mathbf{B}^{(j)}\mathbf{B}^{(j)^T} - \mathbf{B}\mathbf{B}^T\|_F^2) \\
& + \gamma \cdot \sum_{k=1}^{n_c} (\langle \mathbf{C}^{(k)}\mathbf{C}^{(k)^T}, \mathbf{L}_C^{(k)} \rangle + \|\mathbf{C}^{(k)}\mathbf{C}^{(k)^T} - \mathbf{C}\mathbf{C}^T\|_F^2) \\
s.t. \quad & \mathbf{A}, \mathbf{A}^{(i)} \in St(r_1, n_1);\ \mathbf{B}, \mathbf{B}^{(j)} \in St(r_2, n_2);\ \mathbf{C}, \mathbf{C}^{(k)} \in St(r_3, n_3) \\
& \mathcal{P}_\Omega(\boldsymbol{\mathcal{X}}) = \mathcal{P}_\Omega(\boldsymbol{\mathcal{T}});\ \boldsymbol{\mathcal{G}} \in \mathbb{R}^{r_1 \times r_2 \times r_3}
\end{aligned}
\tag{5.4}
$$

where parameter ρ controls the sparsity of factor matrices. And α, β and γ represent the impact of auxiliary information on each mode of tensor, i.e., how important such knowledge is to improve the performance. At first glance, the objective function f is complicated with nonlinear and non-convex orthogonality constraints, we next provide a simple but effective algorithm to solve our problem.

5.4 Optimization Algorithm

In this section, we develop an alternating minimization algorithm to optimize the objective function in Eq. (5.4). To be specific, the objective function f is successively minimized with respect to one variable while fixing others until convergence. To deal with orthogonality constraints, we directly optimize on Grassmann manifolds, an emerging topic in nonlinear programming, to leverage the smooth geometry of the search space and its convergence is guaranteed [Absil et al., 2009, Kasai and Mishra, 2016, Wang et al., 2017, Zhang et al., 2016].

Updating core tensor $\mathcal{G}$: The objective with respect to $\mathcal{G}$ is:

$$\min f(\mathcal{G}) = \|\mathcal{X} - \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}\|_F^2$$

The core tensor $\mathcal{G}$ is obtained as the closed form solution

$$\mathcal{G} = \mathcal{X} \times_1 \mathbf{A}^T \times_2 \mathbf{B}^T \times_3 \mathbf{C}^T \tag{5.5}$$

Updating factor matrices A, B and C: For matrix $\mathbf{A}$, the objective $f(\mathbf{A})$ can be regarded as an unconstrained manifold optimization problem on the Stiefel manifold:

$$\begin{aligned}\min_{\mathbf{A}\in St(r_1,n_1)} f(\mathbf{A}) = \|\mathcal{X} - \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}\|_F^2 + \rho\|\mathbf{A}\mathbf{A}^T\|_1 \\ + \alpha \sum_{i=1}^{n_a} \|\mathbf{A}^{(i)}\mathbf{A}^{(i)^T} - \mathbf{A}\mathbf{A}^T\|_F^2\end{aligned}$$

Above optimization can be further converted into follow [Kolda and Bader, 2009]:

$$\min_{\mathbf{A}\in St(r_1,n_1)} f(\mathbf{A}) = -\|\mathbf{A}^T\mathbf{W}\|_F^2 + \alpha \sum_{i=1}^{n_a} \|\mathbf{A}^{(i)}\mathbf{A}^{(i)^T} - \mathbf{A}\mathbf{A}^T\|_F^2 + \rho\|\mathbf{A}\mathbf{A}^T\|_1 \tag{5.6}$$

where $\mathbf{W} = \mathbf{X}_{(1)}(\mathbf{C} \otimes \mathbf{B})$, in which $\mathbf{X}_{(1)}$ is the mode-1 matricization of tensor $\mathcal{X}$ and $\otimes$ is the Kronecker product. However, simply optimizing problem (5.6) on Stiefel manifold may result in identifiability issue [Absil et al., 2009]. We next analyze

problem (5.6) more deeply.

Consider the r_1-order group $\mathcal{S}(r_1) = \{\mathbf{Q} \in \mathbb{R}^{r_1 \times r_1} | \mathbf{Q}^T\mathbf{Q} = \mathbf{I}\}$ that contains all the $r_1 \times r_1$ orthogonal matrices. With $\mathcal{S}(r_1)$, we can define an equivalent relation $\sim$ on the Stiefel manifold $St(r_1, n_1)$ in the sense that $\mathbf{A} \sim \mathbf{A}'$ indicates that there exists a $\mathbf{Q} \in \mathcal{S}(r_1)$ such that $\mathbf{A} = \mathbf{A}'\mathbf{Q}$. The quotient space of Stiefel manifold $St(r_1, n_1)$ under this equivalence relation is exactly the *Grassmann manifold* $Gr(r_1, n_1)$ [Absil et al., 2009], which consists of all the r_1-dimensional subspaces in n_1-dimensional Euclidean space $\mathbb{R}^{n_1}$ $(0 \leq r_1 \leq n_1)$. Moreover, it is interesting to observe that for any $\mathbf{Q} \in \mathcal{S}(r_1)$, we have the following invariance property:

$$f(\mathbf{A}) = f(\mathbf{AQ})$$

Above equivalence indicates that the function $f(\mathbf{A})$ is independent from the choice of basis spanned by $\mathbf{A}$ and it is thus well defined on the Grassmann manifolds. Instead of optimizing $f(\mathbf{A})$ on Stiefel manifold, a better strategy is thus to regard the problem (5.6) as an unconstrained Grassmann manifold optimization problem:

$$\min_{\mathbf{A} \in Gr(r_1, n_1)} f(\mathbf{A}) = \underbrace{-\|\mathbf{A}^T\mathbf{W}\|_F^2 + \alpha \sum_{i=1}^{n_a} \|\mathbf{A}^{(i)}\mathbf{A}^{(i)^T} - \mathbf{A}\mathbf{A}^T\|_F^2}_{\mathcal{J}_1(\mathbf{A})} + \rho\|\mathbf{A}\mathbf{A}^T\|_1 \tag{5.7}$$

Problem (5.7) can be then efficiently solved by standard gradient descent on the Grassmann manifold such as Riemannian conjugate gradient descent algorithm or trust region algorithm [Absil et al., 2009]. For Grassmann manifold, its Riemannian gradient is the projection of Euclidean gradient into relevant tangent space of the manifold. We next compute the Euclidean gradient of function $f(\mathbf{A})$.

For the first two terms in the problem (5.7), we have:

$$\nabla\mathcal{J}_1(\mathbf{A}) = -2\mathbf{W}\mathbf{W}^T\mathbf{A} + \alpha \sum_{i=1}^{n_a} (4\mathbf{A}\mathbf{A}^T\mathbf{A} - 4\mathbf{A}^{(i)}\mathbf{A}^{(i)^T}\mathbf{A}) \tag{5.8}$$

The third term $\|\mathbf{A}\mathbf{A}^T\|_1$ in objective function (5.7) is not differentiable when the elements of $\mathbf{A}\mathbf{A}^T$ are zeros, we consider the sub-differential. According to the chain rule:

$$\operatorname{vec}\left(\frac{\partial\|\mathbf{A}\mathbf{A}^T\|_1}{\partial(\mathbf{A})}\right)^T = \operatorname{vec}(\operatorname{sgn}(\mathbf{A}\mathbf{A}^T))^T\frac{\partial\mathbf{A}\mathbf{A}^T}{\partial\mathbf{A}} \tag{5.9}$$

where $\operatorname{sgn}(\cdot)$ denotes sign function and $\operatorname{vec}(\cdot)$ is vectorize operator that stacks all columns of a matrix into a long vector. Also, from the partial equation: $\partial(\mathbf{A}\mathbf{A}^T) = (\partial\mathbf{A})\mathbf{A}^T + \mathbf{A}\partial(\mathbf{A}^T)$, we can get:

$$\begin{aligned}\partial\operatorname{vec}(\mathbf{A}\mathbf{A}^T) =&(\mathbf{A}\otimes\mathbf{I}_{n_1})\partial\operatorname{vec}(\mathbf{A}) + (\mathbf{I}_{n_1}\otimes\mathbf{A})\partial\operatorname{vec}(\mathbf{A}^T)\\ =&\left((\mathbf{A}\otimes\mathbf{I}_{n_1}) + \mathbf{K}_{(n_1^2,n_1^2)}(\mathbf{A}\otimes\mathbf{I}_{n_1})\right)\partial\operatorname{vec}(\mathbf{A})\\ =&(\mathbf{I}_{n_1^2} + \mathbf{K}_{(n_1^2,n_1^2)})(\mathbf{A}\otimes\mathbf{I}_{n_1})\partial\operatorname{vec}(\mathbf{A})\end{aligned}$$

From above equation, the derivative part in Eq. (5.9) is

$$\frac{\partial\mathbf{A}\mathbf{A}^T}{\partial\mathbf{A}} = (\mathbf{I}_{n_1^2} + \mathbf{K}_{(n_1^2,n_1^2)})(\mathbf{A}\otimes\mathbf{I}_{n_1})$$

where $\mathbf{I}_{n_1^2}$ is the identity matrix with size $n_1^2\times n_1^2$ and $\mathbf{K}_{(n_1^2,n_1^2)}$ is the commutation matrix. Although the large size of the matrix $\mathbf{I}_{n_1^2}$ and $\mathbf{K}_{(n_1^2,n_1^2)}$, both of them are very sparse with a large number of zeros.

Define the column vector $\mathbf{d}$ as

$$\mathbf{d} = (\frac{\partial\mathbf{A}\mathbf{A}^T}{\partial\mathbf{A}})^T\operatorname{vec}(\operatorname{sgn}(\mathbf{A}\mathbf{A}^T)) \tag{5.10}$$

Combining Eq. (5.8) and (5.10), the Euclidean gradient of the objective function $f(\mathbf{A})$ is

$$\nabla f(\mathbf{A}) = \nabla\mathcal{J}_1(\mathbf{A}) + \rho\cdot\operatorname{ivec}(\mathbf{d}) \tag{5.11}$$

where $\operatorname{ivec}(\cdot)$ is the inverse vectorize operator, i.e., $\operatorname{ivec}(\operatorname{vec}(\mathbf{X})) = \mathbf{X}$. With the

Euclidean gradient $\nabla f(\mathbf{A})$, the Riemannian gradient with respect to $\mathbf{A} \in Gr(r_1, n_1)$ can be computed as

$$\text{grad } f(\mathbf{A}) = (\mathbf{I} - \mathbf{A}\mathbf{A}^T) \cdot \nabla f(\mathbf{A}) \tag{5.12}$$

With the Riemannian gradient, we can use the popular nonlinear ManOpt solver to solve optimization problem (5.7) effectively [Boumal et al., 2014].

For updating factor matrices $\mathbf{B}$ and $\mathbf{C}$, their objective function $f(\mathbf{B})$ and $f(\mathbf{C})$ share the similar optimization structure as $f(\mathbf{A})$. Therefore, they can be solved in the same way with corresponding Riemannian gradients. The details are omitted here.

Updating spectral embedding $\mathbf{A}^{(i)}$,$\mathbf{B}^{(j)}$ and $\mathbf{C}^{(k)}$: For matrix $\mathbf{A}^{(i)}$, the objective $f(\mathbf{A}^{(i)})$ is

$$\min_{\mathbf{A}^{(i)} \in St(r_1, n_1)} f(\mathbf{A}^{(i)}) = \langle \mathbf{A}^{(i)} {\mathbf{A}^{(i)}}^T, \mathbf{L}_A^{(i)} \rangle + \| \mathbf{A}^{(i)} {\mathbf{A}^{(i)}}^T - \mathbf{A}\mathbf{A}^T \|_F^2 \tag{5.13}$$

The objective function $f(\mathbf{A}^{(i)})$ is also invariant to $\mathbf{Q} \in \mathcal{S}(r_1)$, i.e., $f(\mathbf{A}^{(i)}) = f(\mathbf{A}^{(i)}\mathbf{Q})$. Therefore, problem (5.13) can be also regarded as an unconstrained Grassmann manifold optimization problem. The Euclidean gradient of the objective function $f(\mathbf{A}^{(i)})$ is

$$\nabla f(\mathbf{A}^{(i)}) = 2\mathbf{L}_A^{(i)} \mathbf{A}^{(i)} + 4\mathbf{A}^{(i)} {\mathbf{A}^{(i)}}^T \mathbf{A}^{(i)} - 4\mathbf{A}\mathbf{A}^T \mathbf{A}^{(i)} \tag{5.14}$$

The Riemannian gradient grad $f(\mathbf{A}^{(i)})$ can be obtained like in Eq. (5.12). The Riemannian trust-regions algorithm is then used to compute the optimal solution since the objective function $f(\mathbf{A}^{(i)})$ in Eq. (5.13) is smooth. The initial $n_1 \times r_1$ spectral embedding $\mathbf{A}^{(i)}$ can be approximated by the r_1 eigenvectors of $\mathbf{L}_A^{(i)}$ corresponding to the first r_1 smallest eigenvalues. The objective function $f(\mathbf{B}^{(j)})$ and $f(\mathbf{C}^{(k)})$ can be solved in the similar fashion.

Updating tensor $\boldsymbol{\mathcal{X}}$: The optimization problem with respect to $\boldsymbol{\mathcal{X}}$ is formulated as follows:

$$\min_{\boldsymbol{\mathcal{X}}} \| \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} \|_F^2 \quad s.t. \mathcal{P}_\Omega(\boldsymbol{\mathcal{X}}) = \mathcal{P}_\Omega(\boldsymbol{\mathcal{T}})$$

The optimal solution is given by:

$$\boldsymbol{\mathcal{X}} = \mathcal{P}_{\Omega}(\boldsymbol{\mathcal{T}}) + \mathcal{P}_{\Omega^c}(\boldsymbol{\mathcal{G}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}) \tag{5.15}$$

where Ω^c is the complement of Ω, i.e., the set of indexes of the unobserved elements.

According to above analysis, we summarize the algorithm for solving the tensor completion problem (5.4) in Algorithm 2.

Remark: Comparison with Euclidean solvers that usually deal with the orthogonality constraints by solving eigenvalue decomposition problem [Kolda and Bader, 2009], our solver seeks to find concise subspace on the Grassmann manifold for all factor matrices. As pointed out in spectral clustering [Ng et al., 2002], when the leading eigenvalues are almost equal, the best spectral embedding is better determined by the subspace rather than a particular eigenvectors as most of Euclidean solvers do. The same analysis can be analogously applied to solve Tucker decomposition in Euclidean space i.e., using eigenvalue decomposition. The Riemannian solver is thus more interpretable in a straightforward way.

5.5 Experiments

In this section, we evaluate the proposed DTD model using a real-life dataset. The experimental results demonstrate the outstanding performance of DTD compared with existing methods.

5.5.1 Datasets

We collect data from a variety of public sources. We first download the drug-disease associations from the Comparative Toxicogenomics Database [1] (CTD). The original dataset contains $1,048,547$ pairs of drug-disease associations. Here, we only focus

[1] http://ctdbase.org/downloads/

Algorithm 2: DTD

Input: $\boldsymbol{\mathcal{T}}$, Ω, $\{\mathbf{S}_A^{(i)}\}_{i=1}^{n_a}$, $\{\mathbf{S}_B^{(j)}\}_{j=1}^{n_b}$, $\{\mathbf{S}_C^{(k)}\}_{k=1}^{n_c}$ and tol, parameters α, β, γ ρ and rank (r_1, r_2, r_3).

Compute the all Laplacian matrix $\mathbf{L}_A^{(i)}$, $\mathbf{L}_B^{(j)}$ and $\mathbf{L}_C^{(k)}$
Initialize $\boldsymbol{\mathcal{X}}$, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\boldsymbol{\mathcal{G}}$ randomly, set $\boldsymbol{\mathcal{X}}_\Omega = \boldsymbol{\mathcal{T}}_\Omega$
Initialize $\mathbf{A}_0^{(i)}$ with r_1 eigenvectors of $\mathbf{L}_A^{(i)}$ corresponding to the first r_1 smallest eigenvalues. Same process to initialize $\mathbf{B}_0^{(j)}$ and $\mathbf{C}_0^{(k)}$
repeat
    Update $\boldsymbol{\mathcal{G}}_{t+1}$ by Eq. (4.2)
    Compute Riemannian gradient grad $f(\mathbf{A}_t)$ by Eq. (4.9)
    Update $\mathbf{A}_{t+1}$ by using Conjugate-gradient solver
    Update $\mathbf{B}_{t+1}$ and $\mathbf{C}_{t+1}$ the same way as updating $\mathbf{A}_{t+1}$
    for $i \leftarrow 1$ ***to*** n_a **do**
        Update $\mathbf{A}_{t+1}^{(i)}$ by using trust-regions solver
    end
    for $j \leftarrow 1$ ***to*** n_b **do**
        Update $\mathbf{B}_{t+1}^{(j)}$ by using trust-regions solver
    end
    for $k \leftarrow 1$ ***to*** n_c **do**
        Update $\mathbf{C}_{t+1}^{(k)}$ by using trust-regions solver
    end
    Update $\boldsymbol{\mathcal{X}}_{t+1}$ by Eq. (4.12)
until *Objective:* $\|f_{t+1} - f_t\|_F \leq tol$
return $\boldsymbol{\mathcal{X}}$

Table 5.2: Dataset statistics

#drug	#target	#disease	#interaction	sparsity rate
450	708	1,267	188,479	0.047%

on those drugs with DrugBank [2] identifier and diseases with OMIM [3] identifier for conveniently integrating with auxiliary information on other public datasets. As discussed before, the drug's targets, usually proteins that drugs can bind, through which drugs interact with biological pathways and possibly change their behaviors and functions, are crucial in drug discovery. The drug's targets can be collected from the DrugBank database to obtain drug-target interactions. To get dense data, we only include those drugs that interact with at least two targets. We then merge the drug-disease and drug-target interactions into data schema $< drug, target, disease >$. The dataset contains $188,479$ drug-target-disease interactions, involving 450 drugs, 708 targets and $1,267$ diseases. Data statistics are shown in Table 5.2.

The data are then encoded into a $drug \times target \times disease$ tensor $\boldsymbol{\mathcal{X}}$. An entry $\boldsymbol{\mathcal{X}}_{ijk} = 1$ in the tensor indicates that drug i binds to target j and it can treat disease k; $\boldsymbol{\mathcal{X}}_{ijk} = 0$ otherwise. The tensor is very sparse (with sparsity rate 0.047%). We further collect auxiliary information to align each mode of tensor with existing medical knowledge.

Auxiliary information: Following several previous studies [Nascimento et al., 2016, Zheng et al., 2013, Chen and Li, 2017a, Gottlieb et al., 2011, Campillos et al., 2008], we collect auxiliary information from different datasets with respect to drugs, targets, and diseases. For drugs, we define two drug-drug similarities based on drugs' chemical structures and drugs' side effects [Campillos et al., 2008]. For targets, two target-target similarities are considered based on target amino acid sequences and Gene Ontology (GO) terms, both of which can directly be obtain from Uniprot [4]

[2] https://www.drugbank.ca/
[3] https://www.omim.org/
[4] https://www.uniprot.org/

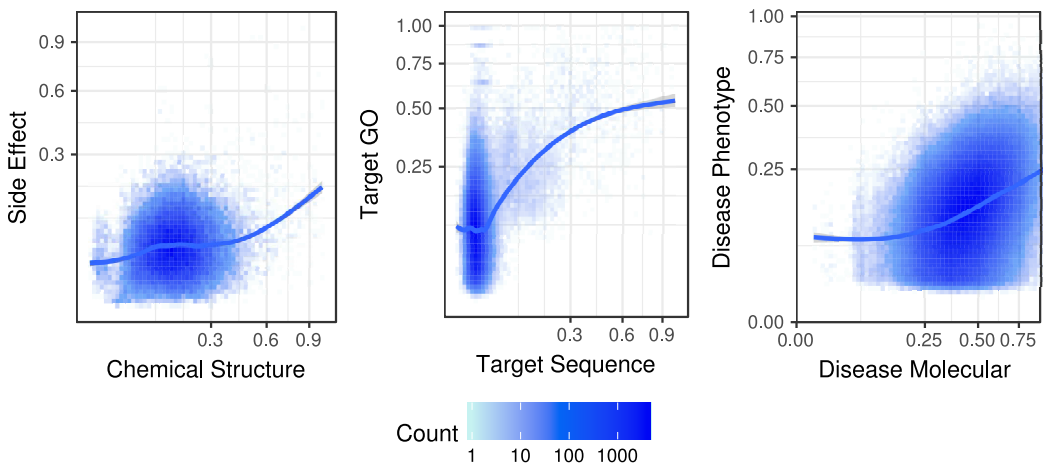

Figure 5.2: The pairwise scatter plots among different views of drugs, targets and diseases.

database [Zheng et al., 2013, Gottlieb et al., 2011]. For diseases, two disease-disease similarities are computed according to disease molecular profiles from biomedical literatures [Caniza et al., 2015] and disease phenotypes on OMIM datasets [Chen and Li, 2017a].

Each view of data contains its own features and emphasizes different aspects. Different views may share some consistency and complementary properties. We begin our investigation by examining the strength of the associations among those different views of drugs, targets, and diseases. The scatter plots of the pairwise similarity scores are shown in Figure 5.2. We further assess the strength of their associations by calculating the correlation coefficient between two views of similarity scores, respectively. All correlations are positive with the correlation coefficient values of 0.148 (drug chemical structures vs. drug side effects), 0.372 (target sequences vs. target GO terms), and 0.374 (disease molecular profiles vs. disease phenotypes). Despite small values in some of the coefficients, all of them are significant because of the large sample sizes (e.g., $802,011$ data instances in the disease scatter plot). Therefore, it's reasonable to integrate the multi-view auxiliary information in the DTD framework and to improve the overall performance for modeling drug-target-disease interactions,

as shown later.

5.5.2 Comparison Methods

As mentioned earlier, many existing methods can only handle the binary relationships of drug-target or drug-disease. We mainly compare with existing tensor completion approaches due to their abilities to represent the triple relationships of drug-target-disease. To show the effectiveness of the proposed DTD model, we compare with several existing models as described below.

- **Tucker decomposition** (TD): It only decomposes the tensor without any auxiliary information for each mode.
- **CMTF** [Acar et al., 2011]: CMTF constructs common latent factors shared by a tensor and single-view of auxiliary information by using coupled matrix-tensor CP-decomposition.
- **ConCMTF** [Bahargam and Papalexakis, 2018]: A novel constrained tensor model with non-negativity, sparsity and orthogonality constraints. Similar to CMTF, it can incorporate single-view of auxiliary information but with Tucker tensor decomposition as base.
- **FaLRTC** [Liu et al., 2013a]: FaLRTC can estimate the low rank structure by imposing the trace norm on its unfolding matrices and the algorithm is based upon alternating direction method of multipliers (ADMM).
- **TFAI** [Narita et al., 2012]: it recovers the tensor by incorporating within-mode auxiliary information and adopt alternating least squares algorithm to solve its problem.
- **Rubik** [Wang et al., 2015]: A novel knowledge-guided tensor factorization and completion framework to fit electronic health record data with non-negativity

and sparsity constraints. It is also solved by ADMM algorithm.

- **AirCP** [Ge et al., 2016]: Another tensor model that integrates single-view auxiliary information using Laplacian regularization and it is optimized by ADMM algorithm.

- **DTDone**: A degraded version of our DTD model with single-view auxiliary information for each mode of tensor.

Note that the approaches TD and FaLRTC cannot integrate any auxiliary information when completing the tensor. The rest of comparison methods can only incorporate single-view auxiliary information for each mode. One has to either only include single-view for each mode, or a straightforward concatenation (e.g., average) of all similarity matrices into one. We tries both ways but the performance do not show much differences. The reason is that concatenation of all views may not be physically meaningful because each view has its owns specific statistical property. For these single-view models, we thus chose the drug chemical structure, target sequence, and disease phenotypes, all of which have long been considered valuable knowledge in drug discovery [Nascimento et al., 2016, Zheng et al., 2013, Gottlieb et al., 2011].

For all Tucker-based tensor models, we set the rank of core tensor $\boldsymbol{\mathcal{G}}$, $r_i = 0.05n_i$ $(1 \leq i \leq 3)$; For CP-based tensor models, we set CP-rank $r = 0.05 \min(n_1, n_2, n_3)$. The regularization parameters of all comparison methods are tuned using the grid-based search algorithm for optimal performance. All methods use the same stopping threshold for a fair comparison, *i.e.*, the variation of two consecutive objective value is less than 10^{-4}. For the proposed DTD and DTDone model, we manually set the parameters $\alpha = \beta = \gamma = \rho = 0.01$. The impact of the regularization parameters on the performance of DTD will be discussed later.

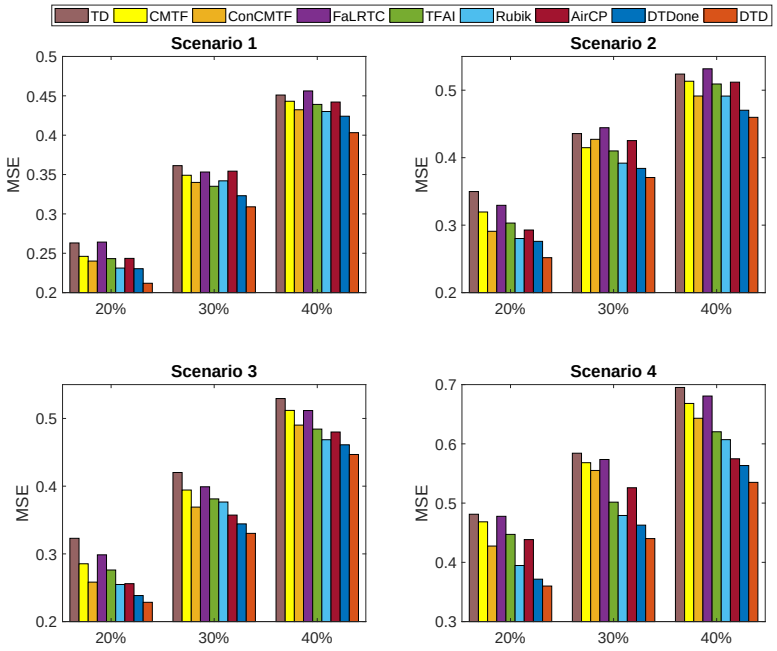

Figure 5.3: Comparison of recovery results over four scenarios with 20%, 30% and 40% test dataset.

5.5.3 Experimental Performance

To investigate the performance of different methods, we randomly selected a subset $\Omega' \in \Omega$ to be used as the unobserved elements (that is the test dataset), and evaluated the *mean squared errors* (MSE) between true values $\boldsymbol{\mathcal{T}}_{\Omega'}$ and predicted values $\boldsymbol{\mathcal{X}}_{\Omega'}$ [Narita et al., 2012]. We mainly consider the following random selection scenarios [Ge et al., 2016]: (i) Scenario 1 (random selection): we randomly select a fraction of elements across drug-target-disease tensor and assume that these are unobservable. (ii) Scenario 2 (random selection in drug mode): we randomly select a fraction of drugs, and assume that those drugs are completely unobservable across the whole tensor. (iii) Scenario 3 (random selection in target mode): similarly, we randomly select a fraction of targets. (iv) Scenario 4 (random selection in disease mode): A fraction of diseases are randomly selected. For each method, the experiment is repeated

ten times independently and the average result is reported.

Figure 5.3 shows the performance of all the methods with different sampling ratios 20%, 30% and 40% (i.e., the fractions of test dataset). DTD consistently performs better than all the comparison methods within a wide range of sampling ratios. In addition, we have the following observations. First, the methods that integrate single-view or multi-view auxiliary information achieve better performance than TD and FaLRTC, indicating the importance of those auxiliary information. With the guidance of extra knowledge, these tensor models keep their performance even with a very sparse input tensor. Second, ConCMTF (Tucker decomposition) perform slightly better than CMTF (CP decomposition) in many situations. The main difference between them is that ConCMTF seeks to find as many non-overlapping structures as possible. Such non-overlapping latent structures are more concise and become specially favorable when jointly decomposing the tensor and matrices. The proposed DTD and DTDone also preserve such non-overlapping structures by imposing orthogonality constraint on factor matrices, which leads to better representations of data. Third, the proposed DTDone further achieves better performance than ConCMTF with an average improvement of 5.76%. As discussed before, the solutions of DTDone, both factor matrices and spectral embeddings, are exactly on the Grassmann manifold and can be well coupled together with each other to obtain better performance. And such advantages make DTDone generally outperform other single-view tensor completion methods such as TEAI, Rubik and AirCP. Finally, the proposed DTD method integrating all available auxiliary information achieves better performance than those only integrating single-view auxiliary information, which illustrates that DTD successfully makes use of all useful information sources to perform effective recovery for the drug-target-disease tensor. In summary, DTD can effectively predict potential interactions of drug, target and disease by leveraging multiple auxiliary data sources and has great potential to accelerate the drug development.

Table 5.3: Precision@k and Recall@k for different methods.

Methods	**Prec@5**	**Rec@5**	**Prec@10**	**Rec@10**
CMTF	0.287	0.347	0.264	0.283
ConCMTF	0.340	0.371	0.270	0.305
TEAI	0.334	0.359	0.254	0.284
Rubik	0.291	0.362	0.269	0.296
AirCP	0.289	0.352	0.261	0.298
DTDone	0.342	0.378	0.273	0.316
DTD	**0.353**	**0.394**	**0.292**	**0.331**

Top-k Prediction: In clinics, given a disease, it is critical to know which drugs can treat this disease as well as the targets involved in the disease pathway. These pairs of drug-target related to a special disease are very important in personalized treatments. We further evaluate the performance of top-k prediction for Scenario 4. Recall that we randomly select a subset of diseases and remove all of their interactions in the Scenario 4. For each disease, we can predict a top-k list of drug-target candidates.

We adopt Precision@k and Recall@k as our evaluation metrics for different methods. Both of them have been widely used to evaluate the quality of top-k predictions [Rendle et al., 2009, Ge et al., 2016]. In our experiments, we evaluate those tensor models with auxiliary information and test for k at 5 and 10 for both precision and recall.

Table 5.3 shows the top-k prediction performance of all the methods with 20% fractions of test dataset. Overall, the proposed DTD model performs the best among all methods in terms of both precision and recall with different values of k. The trend is very similar to the results based on the MSE metric. For example, DTD performs better than CMTF with an average improvement of 16.8% in precision and 15.3% in recall. We attribute the poor performance of the CMTF approach to the sparseness of tensor and its weak abilities of coupling tensor with auxiliary matrices (e.g., overlap structure). Moreover, DTD has a better performance than DTDone, indicating the superiority of a tensor model integrating rich multi-view auxiliary information. All

Table 5.4: Top 10 novel triple relationship of (drug→target→ disease) by DTD model.

Carbamazepine→ Nuclear receptor subfamily 1 group I member 2 → Osteoporosis
Testosterone→ Estrogen receptor→Myocardial infarction
Nefazodone→D(2) dopamine receptor→Schizophrenia
Raloxifene→Estrogen receptor→Obesity
Fenofibrate→Matrix metalloproteinase→Psoriatic arthritis
Acetazolamide→Estrogen receptor→Amyotrophic lateral sclerosis
Raloxifene→Androgen receptor→Breast cancer
Amoxapine→Potassium voltage-gated channel subfamily H member 2 →Hepatocellular carcinoma
Nefazodone→Histamine H1 receptor→Schizophrenia
Promethazine→Muscarinic acetylcholine receptor M3→Obesity

these results illustrate that the proposed method can successfully predict top-k drug-target pairs by exploiting the compatible and complementary information from multi-view data sources.

Given these encouraging results, we use DTD model to predict novel triple relationships of drug-target-disease by leveraging all the observed data. The top-10 novel triple relationships are listed in Table 5.4. The results can then be evaluated by domain experts to see whether such interactions are clinically meaningful. Also, such top-k candidates can be further validated *in vitro* and *in vivo*. In summary, our proposed model efficiently predict potential interaction candidates with high accuracy, providing a systematic approach to narrow down the search space for further wet-lab investigations.

5.5.4 Parameter Studies

We further analyze the effect of key hyperparameters, the rank of core tensor (r_1, r_2, r_3) and the regularized parameters α, β, γ, and ρ.

The impact of tensor rank:

In Tucker-based tensor decomposition, the hyperparameter (r_1, r_2, r_3) controls the rank of the core tensor $\boldsymbol{\mathcal{G}}$, which also decides the number of latent features for each mode, i.e., $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$. In contrast to Tucker model, the CP decomposition only have one hyperparameter r, the CP-rank. In the experiments, we set the rank $r_i = \delta \cdot n_i$ $(1 \leq i \leq 3)$ for Tucker-based model and $r = \delta \cdot \min(n_1, n_2, n_3)$ for CP-based model. The δ is the ratio varying within $[0.05, 0.1, 0.15, 0.2, 0.25]$. We then evaluate all methods for scenario 4 with 20% as the test dataset. Figure 5.4 shows that our model benefits slightly from larger numbers of latent dimensions in terms of MSE and Prec@5 metric. But generally speaking, both DTD and DTDone are fairly robust with respect to δ.

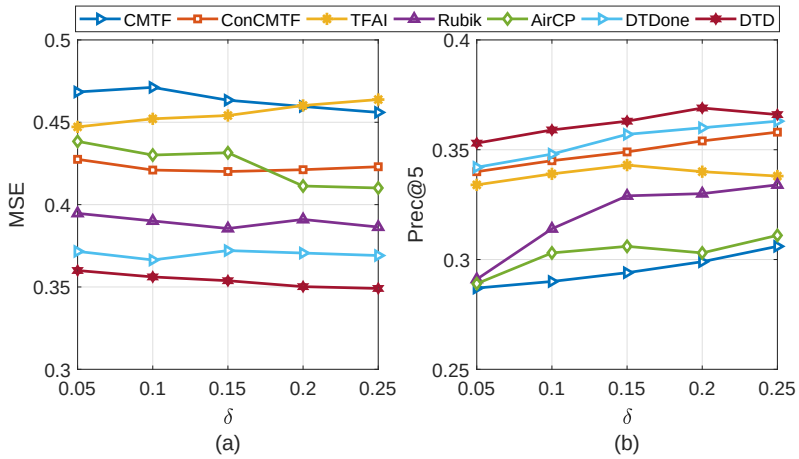

Figure 5.4: Effect of the tensor rank in scenario 4.

The impact of regularized parameters

Finally, we explore the impact of the regularization parameters on the quality of tensor completion. Recall that α, β and γ control the contributions of auxiliary information of drugs, targets, and disease, respectively. ρ controls the sparsity of factor matrices. In order to better understand the effect of these parameters, we vary their values in

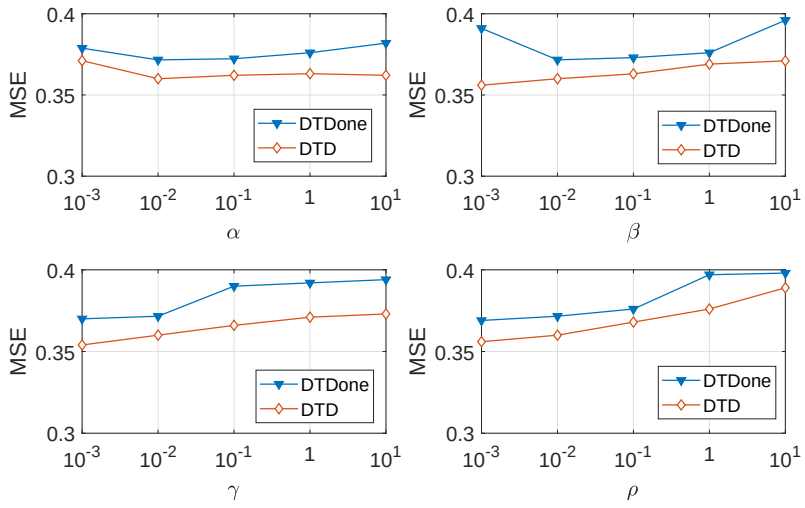

Figure 5.5: Effect of regularized parameters:α, β, γ and ρ .

the range $[0.001, 0.01, 0.1, 1, 10]$, and then evaluate DTDone and DTD for scenario 4 with 20% as testing data. When studying one variable (e.g., α), we fix all the rest variables to be 0.01. As shown in Figure 5.5, DTDone and DTD are stable over a wide range of α, β and γ in terms of MSE. Specifically, a relatively low MSE can be achieved when those regularized parameters are around 0.01.

5.6 Related Work

We review related research on computational predictions of drugs, targets, and diseases, as well as on tensor completion for high-dimensional structured data.

Modeling drug-target-disease. Treating human diseases caused by complex biological processes involves activities of many biological entities such as drugs and targets. Methods in computational pharmacology aim to find associations among those entities, and understand drug's MoAs [Hauser et al., 2017]. Two excellent surveys [Li et al., 2015a], [Ezzat et al., 2018] provide a very detailed overview of different computational methods for predicting drug-disease and drug-target relationships, re-

spectively. Among existing approaches, one very popular method is network-based inference models, e.g., a bipartite network consisting of one layer for drugs and one layer for diseases (targets). Different machine learning algorithms, such as random-walk [Chen et al., 2012], matrix factorization [Ezzat et al., 2017], and support vector machine [Yamanishi et al., 2008], have been applied to predict novel interactions of drug-disease (or target). In addition to network topology, existing medical knowledge, such as a variety of omics data, on the Web can be incorporated to better understand complex human metabolic systems [Zitnik et al., 2018, Zheng et al., 2013, Chen and Li, 2017a, Nascimento et al., 2016]. For example, a multiple kernel learning method has been developed to predict drug-target relationships by integrating multiple similarities of drugs and targets. Decagon models drug polypharmacy side effects via graph convolutional networks with additional drug-drug and protein-protein interactions [Zitnik et al., 2018].

However, current studies generally considered drug-disease and drug-target predictions as two independent tasks and the relationships of drug-target-disease is typically ignored. There exist several studies incorporating target information in the task of drug-disease predictions [Wang et al., 2014, Cheng et al., 2012, Zitnik et al., 2018]. However, their extensions are unsatisfactory because their goals are still on modeling binary relationships, not on the triple drug-target-disease patterns we aim to learn.

Tensor completion. Tensors are very powerful in real-world applications because of their ability to represent multi-aspects or high-dimensional data. In many applications, we are often interested in analyzing tensors together with matrices from additional information, such as computational phenotyping [Wang et al., 2015] and recommender system [Chen and Li, 2019d, Ge et al., 2016, Chen and Li, 2019a]. This coupled tensor-matrix factorization method has been developed over the years. For example, Acar et. al. proposed a gradient-based algorithm for coupled matrix-tensor factorization [Acar et al., 2011]. Narita et. al. incorporated valuable auxiliary

information to further improve the quality of tensor recovery [Narita et al., 2012]. Wang et. al. proposed a knowledge guided tensor factorization model for health data analysis [Wang et al., 2015]. Ge et. al. proposed a spatiotemporal dynamics recovery framework, which could capture the latent relationships among locations, memes, and times by coupled factorization [Ge et al., 2016]. However, none of these methods are guaranteed to non-overlapping structures in both tensor and matrices and can only integrate single-view of additional information for *each mode* of tensor. Some models [Kolda and Bader, 2009, Narita et al., 2012, Bahargam and Papalexakis, 2018] try to impose orthogonality constraint on factor matrices. However, their solutions are usually not unique, which is hard to couple with auxiliary information. Recently, concrete tensor/matrix representations on the manifold is a fast growing research topic [Absil et al., 2009, Kasai and Mishra, 2016, Zhang et al., 2016]. Kasai et. al. recently developed a novel Riemannian manifold preconditioning approach for tensor completion [Zhang et al., 2016], but they did not take the auxiliary information into account. The proposed approach is innovative and properly addresses the challenges in coupled tensor-matrix factorization with multi-view auxiliary information.

5.7 Conclusion

Modeling drug-target-disease interactions is important for understanding drugs' MoAs in drug development. The problem has conventionally been addressed separately by considering associations of drug-target or drug-disease, and most existing methods cannot leverage intrinsic interactions among these three biological entities. Here we present a novel approach DTD, which explicitly explores a three-way drug-target-disease tensor via coupled tensor-matrix factorization. Completing such tensor is challenging because of its large sparsity. DTD elegantly integrates multiple Web data to align existing knowledge with the tensor. With the guidance of auxiliary infor-

mation, DTD infers the concurrence of drug-target-disease more accurately than do baselines. Another distinguishing aspect of DTD is that it directly optimizes on the Grassmann manifold, which is more effective than Euclidean solvers. Experimental results on a real-world dataset demonstrate the effectiveness and efficiency of the proposed method.

The proposed DTD can easily incorporate additional domain knowledge and can be extended to detect high-order tensor (e.g., four-way *drug-target-gene-disease* tensor) with relatively little effort. Future work should aim to extend this model to simulate human metabolic systems by considering more biological entities (e.g., gene) as well as other domain knowledge (e.g., gene expression data). Deep understanding of those entities opens up opportunities to use rich Web data to assist follow-up analysis via formal pharmacological studies.

Chapter 6

Neural Tensor Network for Drug-Target-Disease Interactions

6.1 Introduction

Data-Driven Drug Discovery: Personalized medicine recommendation is one of the most promising assets to treat human disease [Wu et al., 2013b]. A critical step in personalized medicine is to understand drugs' mechanism of actions (MoAs) by exploring the biological interactions among drugs, targets, and diseases. *In vitro* experiments can be performed to identify potential associations of drug-target-disease, but such systematic screening remains an expensive and time-consuming process. It takes more than 13 years and $2.87 billion to bring a new drug into the market [Hauser et al., 2017]. Researchers are thus resorting to machine learning to understand the drugs' MoAs by mining the emergence of large-scale chemical and genomic data.

Two most prominent data-driven tasks among these recent developments are drug-disease and drug-target prediction (see [Ezzat et al., 2018, Li et al., 2015a] for surveys). In these tasks, researchers have attempted to collect a variety of omics data,

and predict new interactions of drug-disease or drug-target through network inference [Wu et al., 2013b, Chen et al., 2020], multi-view learning [Zheng et al., 2013], and deep learning [Tsubaki et al., 2018]. Beyond pairwise drug-disease or drug-target relationships, some recent studies have pointed at the importance of identifying triple-wise interactions of drug-target-disease in human metabolic systems [Capuzzi et al., 2018, Chen and Li, 2019f, Wang et al., 2018]. Among different methods, tensor factorization is a commonly used method to infer the missing entries of a drug-target-disease tensor [Chen and Li, 2019f, Wang et al., 2018].

Tensor Factorization: Tensor factorizations aim to extract latent structure from high dimensional data [Kolda and Bader, 2009]. CP (CANDECOMP/PARAFAC) and Tucker are two popular tensor models with diverse variants being successfully applied in many applications [Wang et al., 2015, Chen and Li, 2019c, He et al., 2019, Chen and Li, 2019b]. However, the CP and Tucker models (or their variants) suffer from two weaknesses.

First, their performance can be limited by linearity, which might not be expressive well for nonlinear data manifolds. Recently, a series of studies have shown that nonlinear tensor factorizations have superior performances over multilinear tensor models [Fang et al., 2015, Xu et al., 2012, Liu et al., 2019, Wu et al., 2019]. For example, NLTF [Fang et al., 2015] and InfTucker [Xu et al., 2012] are proposed to use a nonlinear Gaussian kernel. However, they rely on a prior Gaussian process over tensor data, which might be difficult to estimate in practice [Zhe et al., 2016].

The second drawback of CP or Tucker models is the data sparsity issue. To alleviate this issue, coupled tensor-matrix models are extended to jointly analyze tensor together with auxiliary information [Wang et al., 2015, Narita et al., 2012, Acar et al., 2011]. However, these methods are inherently limited by encoding *feature matrices*, which require tedious feature engineering [Shan et al., 2016]. As the number of features grows, designing and deploying them become challenging, especially for healthcare

data.

Contributions: To tackle these challenges, we propose a Neural Tensor Network (NeurTN), which seamlessly combines tensor algebra and deep neural networks to provide effective medicine recommendations. By replacing the multilinear multiplication with a neural network, NeurTN is able to characterize nonlinear dependencies among tensor data. Moreover, NeurTN incorporates a collection of heterogeneous information to alleviate the data sparsity issue. Instead of constructing them as feature matrices, NeurTN uses geometric neural networks to learn the embeddings from molecular graphs and target sequences, which allows to be trained end-to-end. Our data-driven model opens up opportunities to use large-scale omics data to discover drug' MoAs in pharmacological studies.

6.2 Preliminaries

6.2.1 Tensor Algebra

The *n-mode product* of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $\mathbf{A} \in \mathbb{R}^{J \times I_n}$ is denoted as $\boldsymbol{\mathcal{X}} \times_n \mathbf{A}$ with size $\mathbb{R}^{I_1 \cdots \times I_{n-1} \times J \times I_{n+1} \cdots \times I_N}$. Also, we have

$$(\boldsymbol{\mathcal{X}} \times_n \mathbf{A})_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} \boldsymbol{\mathcal{X}}_{i_1 i_2 \cdots i_N} \mathbf{A}_{j i_n}$$

Tensor Factorization: CP and Tucker are two widely used tensor models, which assumes a compact hidden structure in the data [Kolda and Bader, 2009].

The CP model decomposes a third-order tenor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{M \times N \times L}$ into three factor matrices: $\mathbf{U} \in \mathbb{R}^{M \times r}$, $\mathbf{V} \in \mathbb{R}^{N \times r}$, and $\mathbf{W} \in \mathbb{R}^{L \times r}$, such that a tensor entry can be estimated by:

$$\hat{\boldsymbol{\mathcal{X}}}_{ijk} = f(i, j, k | \mathbf{U}, \mathbf{V}, \mathbf{W}) = \sum_{t=1}^{r} \mathbf{U}_{it} \mathbf{V}_{jt} \mathbf{W}_{kt} \tag{6.1}$$

here r is the rank, also known as CP tensor rank.

The Tucker decomposes a tenor $\boldsymbol{\mathcal{X}}$ into a core tensor $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ with three orthogonal factor matrices: $\mathbf{U} \in \mathbb{R}^{M \times r_1}$, $\mathbf{V} \in \mathbb{R}^{N \times r_2}$, and $\mathbf{W} \in \mathbb{R}^{L \times r_3}$, such that:

$$\hat{\boldsymbol{\mathcal{X}}}_{ijk} = f(i, j, k | \boldsymbol{\mathcal{G}}, \mathbf{U}, \mathbf{V}, \mathbf{W}) = \sum_{a=1}^{r_1} \sum_{b=1}^{r_2} \sum_{c=1}^{r_3} \boldsymbol{\mathcal{G}}_{abc} \mathbf{U}_{ia} \mathbf{V}_{jb} \mathbf{W}_{kc} \tag{6.2}$$

Limitation: As we can see, both the CP and Tucker models interpret the three-way feature interactions through multilinear multiplication. For example, the CP model estimates $\hat{\boldsymbol{\mathcal{X}}}_{ijk}$ by linearly combining of latent factors with equal contribution. We argue that the multilinear strategy may be insufficient to capture the nonlinear feature interactions.

Our work builds on this line of work and addresses this limitation by learning the predictor $f(\cdot)$ using neural networks.

6.2.2 Problem Definition

The medicine recommendation task can be formulated as a tensor completion problem [Chen and Li, 2019f, Wang et al., 2018]. To be specific, the input can be organized as a drug-target-disease tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{M \times N \times L}$, where M, N, and L denote the number of drugs, targets, and diseases, respectively. An entry $\boldsymbol{\mathcal{X}}_{ijk} = 1$ if an interaction among drug i, target j, and disease k is observed; $\boldsymbol{\mathcal{X}}_{ijk} = 0$, otherwise. Our aim is to estimate the scores of unobserved elements $\hat{\boldsymbol{\mathcal{X}}}_{ijk}$, which can be used to infer novel interactions of drug-target-disease.

6.2.3 Feature Encodings

Before presenting our model, we describe the related features.

Embedding Look-up:

Given a drug i, a target j, and a disease k, their one-hot features $\mathbf{a}_i \in \mathbb{R}^M$, $\mathbf{b}_j \in \mathbb{R}^N$, and $\mathbf{c}_k \in \mathbb{R}^L$ can be obtained based on their identities. We can obtain their dense

embeddings via three lookup tables:

$$\widehat{\mathbf{u}}_i \leftarrow \text{lookup}(\mathbf{a}_i), \quad \widehat{\mathbf{v}}_j \leftarrow \text{lookup}(\mathbf{b}_i), \quad \widehat{\mathbf{w}}_k \leftarrow \text{lookup}(\mathbf{c}_i) \tag{6.3}$$

where $\widehat{\mathbf{u}}_i \in \mathbb{R}^{d_1}$, $\widehat{\mathbf{v}}_j \in \mathbb{R}^{d_2}$, and $\widehat{\mathbf{w}}_k \in \mathbb{R}^{d_3}$ are new embeddings for drug i, target j, and disease k, respectively.

Here we also incorporate medical knowledge of drugs and targets. We leave the exploration of diseases as a future work. Instead of constructing these auxiliary information as feature matrices [Zheng et al., 2013, Chen and Li, 2019f], we apply geometric neural networks to learn the structural information of both drugs' chemical structures and targets' sequences.

GNN for Molecular Graph:

A chemical molecule can be represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where nodes denote the atoms and edges denote the chemical bonds between atoms [Duvenaud et al., 2015]. Given a graph $\mathcal{G}_i$ for drug i, we extract features of a molecular graph by using a graph neural network:

$$\mathbf{r}^i_{\mathcal{G}} \leftarrow \text{GNN}(\mathcal{G}_i)$$

here $\mathbf{r}^i_{\mathcal{G}} \in \mathbb{R}^g$ is the molecule-based features for drug i.

CNN for Target Sequence:

We can also obtain representations of target sequences using a convolutional neural network (CNN) [Tsubaki et al., 2018]. Let $\mathcal{S}_j$ be an amino acid sequence for target j, we have

$$\mathbf{r}^j_{\mathcal{S}} \leftarrow \text{CNN}(\mathcal{S}_j)$$

here $\mathbf{r}^j_{\mathcal{S}} \in \mathbb{R}^s$ is the sequence-based features for target j.

Fusion Feature Encodings:

We integrate all the features for better representation learning. Given drug's features $(\widehat{\mathbf{u}}_i, \mathbf{r}^i_{\mathcal{G}})$, target's features $(\widehat{\mathbf{v}}_j, \mathbf{r}^j_{\mathcal{S}})$, and disease's feature $\widehat{\mathbf{w}}_k$, we have:

$$\mathbf{u}_i \leftarrow \mathrm{FC}(\Theta_u; \widehat{\mathbf{u}}_i \oplus \mathbf{r}^i_{\mathcal{G}}), \quad \mathbf{v}_j \leftarrow \mathrm{FC}(\Theta_v; \widehat{\mathbf{v}}_j \oplus \mathbf{r}^j_{\mathcal{S}}), \quad \mathbf{w}_k \leftarrow \widehat{\mathbf{w}}_k \tag{6.4}$$

here $\oplus$ is the concatenation operator; $\{\mathbf{u}_i, \mathbf{v}_j, \mathbf{w}_k\} \in \mathbb{R}^r$ are the new representations. $\mathbf{w}_k$ only contains one-hot feature since there is no auxiliary information for diseases. Two fully connected layer $\mathrm{FC}(\cdot)$ with parameters Θ_u and Θ_v are followed to obtain more sophisticated representations of drugs and targets. More importantly, by properly choosing the weights in Θ_u and Θ_v, we can reshape the feature vectors $\mathbf{u}_i$ and $\mathbf{v}_j$ to have the same dimension as $\mathbf{w}_k$. As such, $\{\mathbf{u}_i, \mathbf{v}_j, \mathbf{w}_k\}$ can be interpreted as the *latent factors* in CP tensor factorization as shown latter in Eq. (6.7).

6.3 The Proposed Model

To better learn multi-aspect features, the NeurTN contains three components: MLP, GCP, and CML as in Figure 6.1.

6.3.1 Multi-Layer Perceptron (MLP)

It is straightforward to feed the concatenated features of drugs, targets, and diseases into a MLP [Wu et al., 2019, He et al., 2017], in which each hidden layer can capture nonlinear interactions among $\mathbf{u}_i$, $\mathbf{v}_j$, and $\mathbf{w}_k$:

$$\begin{aligned} \mathbf{z}_L &= \mathrm{MLP}(\mathbf{u}_i \oplus \mathbf{v}_j \oplus \mathbf{w}_k), \\ \hat{\boldsymbol{\mathcal{X}}}_{ijk} &= \sigma(\mathbf{W}_L \mathbf{z}_L + \mathbf{b}_L) \end{aligned} \tag{6.5}$$

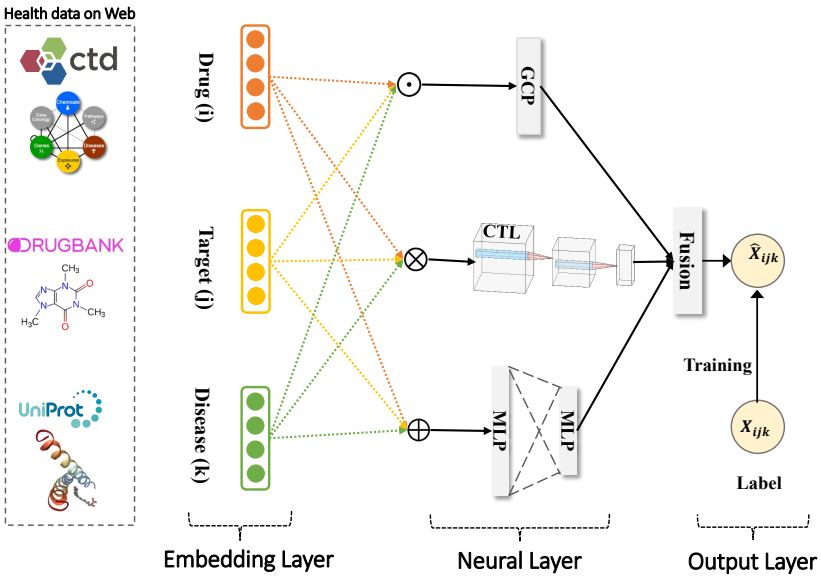

Figure 6.1: Overall architecture of NeurTN.

here $\mathbf{W}_L$ and $\mathbf{b}_L$ denote the weight and bias. We choose the ReLU$(\cdot)$ as activation function in hidden layers and the sigmoid function $\sigma(\cdot)$ as predictor function.

The MLP is capable of learning nonlinearity of the concatenated features. Nevertheless, the concatenated features may lose some information in the original embeddings that are useful for later interaction learning. To avoid such information loss, we further propose two triple-wise layers for learning multi-aspect features.

6.3.2 Generalized CP Tensor Layer (GCP)

As shown in Eq. (6.1), the CP model can interpret linear triple-wise interactions by its multilinear product. Here we generalize the CP to learn nonlinear feature interactions.

Given $\{\mathbf{u}_i, \mathbf{v}_j, \mathbf{w}_k\} \in \mathbb{R}^r$ from embedding layers, we design a novel GCP layer $\boldsymbol{\phi}(\cdot)$, which contains a pooling operator that converts a set of embedding vectors to one vector:

$$\boldsymbol{\phi}(\mathbf{u}_i, \mathbf{v}_j, \mathbf{w}_k) = \mathbf{u}_i \odot \mathbf{v}_j \odot \mathbf{w}_k \tag{6.6}$$

here $\odot$ is the element-wise product. Clearly, the GCP layer $\boldsymbol{\phi}(\cdot)$ does not introduce extra model parameters, and more importantly, it can be efficiently computed in linear time. Then, we can project the hidden vector $\boldsymbol{\phi}(\mathbf{u}_i, \mathbf{v}_j, \mathbf{w}_k)$ by:

$$\hat{\boldsymbol{\mathcal{X}}}_{ijk} = f_{out}(\mathbf{h}^T(\mathbf{u}_i \odot \mathbf{v}_j \odot \mathbf{w}_k)) \tag{6.7}$$

here $f_{out}(\cdot)$ and $\mathbf{h}$ denote the activation function and weights.

Proposition 1. *The CP tensor factorization in Eq.(6.1) is a special case of generalized tensor factorization in Eq.(6.7).*

Proof. Let f_{out} be an identity function ($f_{out}(x) = x$), $\mathbf{h}$ be a uniform weight vector of 1 ($\mathbf{h} = [1, \ldots, 1]^T \in \mathbb{R}^r$), and $\mathbf{u}_i(t)$ be the t-th element in the column vector $\mathbf{u}_i$, we have:

$$\hat{\boldsymbol{\mathcal{X}}}_{ijk} = f_{out}(\mathbf{h}^T(\mathbf{u}_i \odot \mathbf{v}_j \odot \mathbf{w}_k)) = \sum_{t=1}^{r} \mathbf{u}_i(t)\mathbf{v}_j(t)\mathbf{w}_k(t)$$

which exactly recovers the CP factorization in Eq.(6.1) and the embedding size r now becomes the tensor CP rank. □

This is a very appealing property, meaning that we can develop a nonlinear CP tensor model by learning the f_{out} and $\mathbf{h}$. For example, if we adopt a nonlinear predictor f_{out} and allow $\mathbf{h}$ to be trained from data, GCP will generalize the CP to a nonlinear tensor machine. As neural networks have strong ability to fit the data, GCP is thus capable of modeling complex feature interactions, which subsumes the linear CP model. In this work, we implement the GCP by using the sigmoid function $f_{out}(x) = 1/(1 + e^{-x})$ and training $\mathbf{h}$ by the pairwise loss function.

6.3.3 Compressed Tensor Layer (CTL)

To better capture the triple-wise feature interactions, we further propose to use an outer product on $\{\mathbf{u}_i, \mathbf{v}_j, \mathbf{w}_k\}$ via:

$$\boldsymbol{\mathcal{E}} = \mathbf{u}_i \otimes \mathbf{v}_j \otimes \mathbf{w}_k \tag{6.8}$$

here $\boldsymbol{\mathcal{E}} \in \mathbb{R}^{r \times r \times r}$ is a tensor feature map and $\otimes$ is the outer product. Our motivation for $\boldsymbol{\mathcal{E}}$ is straightforward. The map $\boldsymbol{\mathcal{E}}$ captures more signals than element-wise product in Eq.(6.6) since it encodes *any* tripe-wise feature interactions. Such strategy has been widely used to boost system performance in deep learning [He et al., 2018].

To exploit the nonlinearity of $\boldsymbol{\mathcal{E}}$, one can flat $\boldsymbol{\mathcal{E}}$ to a vector and feed it to another MLP as Eq.(6.5). However, unlike the size of input in Eq.(6.5) (e.g., $3r$), the size of $\boldsymbol{\mathcal{E}}$ (e.g., r^3) requires much more neurons. As an example, assuming we have a feature map $\boldsymbol{\mathcal{E}} \in \mathbb{R}^{64 \times 64 \times 64}$ and adopt a MLP with the half-size tower structure. In this case, even the first layer of the MLP requires $262,144 \times 131,072$ parameters, not to mention the use of more layers.

To address this issue, we turn our attention to *n-mode product* in Sec. 6.2.1. Inspired by this shrinking technique, we propose a simple but efficient CTL (Compressed Tensor Layer) to perform a feedforward computation on tensor feature map.

CTL block: Given input $\boldsymbol{\mathcal{E}}$, we apply one CTL block as:

$$\boldsymbol{\mathcal{H}} = g(\boldsymbol{\mathcal{E}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} + \boldsymbol{\mathcal{B}}) \tag{6.9}$$

where $\boldsymbol{\mathcal{H}}$ is output in the hidden layer; $g(\cdot)$ is the activation function; $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ and $\boldsymbol{\mathcal{B}}$ are weights and bias. At first glance, our CTL is very similar to Tucker in Eq.(6.2), but they are essentially different. Tucker requires the orthogonality of latent factors [Kolda and Bader, 2009] and it does not have the bias $\boldsymbol{\mathcal{B}}$ or activation function as in CTL.

CTL significantly reduces the number of parameters in the feedforward. For exam-

ple, we can compress $\boldsymbol{\mathcal{E}}$ with size $\mathbb{R}^{64\times64\times64}$ to $\mathbb{R}^{16\times16\times16}$ by weights $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\} \in \mathbb{R}^{16\times64}$ and bias $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{16\times16\times16}$, resulting in $3 \times 16 \times 64 + 16^3$ parameters.

Stacking CTL blocks: Arguably two of the key contributors of the neural networks are its nonlinearity and the stacking of multiple layers. Likewise, we stack multiple CTL blocks with ReLU$(\cdot)$ as the activation function:

$$\boldsymbol{\mathcal{H}}_1 = \text{ReLU}(\boldsymbol{\mathcal{E}} \times_1 \mathbf{A}^{(0)} \times_2 \mathbf{B}^{(0)} \times_3 \mathbf{C}^{(0)} + \boldsymbol{\mathcal{B}}^{(0)}), ...,$$
$$\boldsymbol{\mathcal{H}}_L = \text{ReLU}(\boldsymbol{\mathcal{H}}_{L-1} \times_1 \mathbf{A}^{(L-1)} \times_2 \mathbf{B}^{(L-1)} \times_3 \mathbf{C}^{(L-1)} + \boldsymbol{\mathcal{B}}^{(L-1)}),$$

As such, the input tensor $\boldsymbol{\mathcal{E}} \in \mathbb{R}^{r\times r\times r}$ is successively compressed by a sequence of three weight matrices along each mode with nonlinear transformation.

Dropout: Dropout [Srivastava et al., 2014] is a regularization technique for neural networks to prevent overfitting. The idea of dropout is simple: randomly "turn off" neurons with probability ρ during training, and use all neurons when testing. We also apply a dropout layer on the feature map $\boldsymbol{\mathcal{E}}$, i.e., randomly dropping ρ percent of its elements.

Prediction layer: At last, the output of the last hidden layer $\boldsymbol{\mathcal{H}}_L$ is transformed to the final predictive score:

$$\hat{\boldsymbol{\mathcal{X}}}_{ijk} = \sigma(\mathbf{W}_o \times \text{Reshape}(\boldsymbol{\mathcal{H}}_L) + \mathbf{b}_o) \tag{6.10}$$

where Reshape$(\cdot)$ flats $\boldsymbol{\mathcal{H}}_L$ into a vector. The output layer is a fully connected layer with the sigmoid function as predictor.

6.3.4 The Overall Model

Joint Training:

So far we have developed three instantiations of nonlinear tensor models: MLP, GCP, and CTL. We present our unified model NeurTN by joint learning these three modules. Formally, let $\mathbf{z}_L$, $\boldsymbol{\phi}$, and $\boldsymbol{\mathcal{H}}_L$ denote the outputs of the last hidden layers of

MLP, GCP, and CTL, respectively. Then, we have:

$$\begin{aligned} \mathbf{F} &= \mathbf{z}_L \oplus \boldsymbol{\phi} \oplus \text{Reshape}(\boldsymbol{\mathcal{H}}_L), \\ \hat{\boldsymbol{\mathcal{X}}}_{ijk} &= \sigma(\mathbf{W}_f \mathbf{F} + \mathbf{b}_f) \end{aligned} \tag{6.11}$$

here we choose the sigmoid function $\sigma(\cdot)$ as final predictor.

Relation to Wide&Deep Learning: Our NeurTN shares a similar spirit with the well-known Wide&Deep Learning [Cheng et al., 2016, He et al., 2017], which shows that joint learning wide and deep models has the benefits of memorization and generalization. Our GCP can be regarded as a wide component whereas the MLP and CTL can be viewed as deep components. The key difference is that our NeurTN is able to learn multi-aspect tensor data, while existing work can only learn two-dimensional matrices.

Model Optimization:

We adopt pairwise learning methods to optimize model parameters [Bordes et al., 2013, Yang et al., 2015]. The idea behind pairwise learning is that an observed triplet should be predicted with a higher score than an unobserved one. This can be achieved by minimizing:

$$\mathcal{L}(\Theta) = \sum_{(i,j,k)\in\mathcal{D}^+} \sum_{(i',j',k')\in\mathcal{D}^-} \max(0, 1 + f(i',j',k') - f(i,j,k)) \tag{6.12}$$

here $f(\cdot)$ and Θ denote our predictive function and model parameters in Eq.(6.11). $\mathcal{D}^+$ denotes the set of positive triplets (e.g., $\boldsymbol{\mathcal{X}}_{ijk} = 1$), and $\mathcal{D}^-$ denotes the set of negative triplets corresponding to $\mathcal{D}^+$ by sampling from unobserved elements. Following [Bordes et al., 2013, Yang et al., 2015], for each positive training triplet (i,j,k), we randomly sample one negative training triplet (i',j',k') in the training step.

It is worth mentioning that our work is different from recent deep tensor networks [Novikov et al., 2015, Socher et al., 2013, Lebedev et al., 2015]. The tensor in [Socher et al., 2013] aimed to connect nodes in knowledge graphs. [Novikov et al., 2015,

Lebedev et al., 2015] focused on establishing relationships between tensor and deep learning, not on nonlinear tensor factorization. Our model here is more general to study nonlinear patterns for multi-aspect tensor data.

6.3.5 Complexity Analysis

The computational complexity of our model comes from MLP, GCP and CTL modules. For MLP, the complexity for matrix multiplication is $O(d_1 r)$, where r is the feature size in Eq.(6.4); d_1 is the embedding size in the hidden layer. The complexity for GCP is $O(r)$. The computational complexity for the tensor-matrix multiplication in CTL is $O(r^3 d_2^2)$, where d_2 is the embedding size of hidden layers in CTL. In practice, the embedding size d_1 and d_2 are typically small and the feature size $r \ll \min(M, N, L)$. The overall complexity can be simplified as $O(|c| \cdot r^3)$, where $|c|$ is a constant.

6.4 Experiments

In this section, we aim to answer the following questions:

RQ1: Do our proposed MLP, GCP, CTL, and NeurTN capture better nonlinear feature interactions?

RQ2: How does the proposed NeurTN outperform the state-of-the-art coupled tensor-matrix factorizations?

RQ3: How do different modules (e.g., MLP, GCP, and CTL) affect the performance of NeurTN?

6.4.1 Experimental Settings

Datasets: We collect data from three databases [Chen and Li, 2019f, Wang et al., 2018]: CTD[1], DrugBank[2], and UniProt[3]. We only focus on drugs that have DrugBank identifier for later collecting auxiliary information. As such, we obtain $436,322$ triplets of drug-target-disease, involving $1,901$ drugs, $2,514$ targets, and $2,923$ diseases. For drugs, their SMILES, a string encoding of chemical structures, are downloaded from DrugBank. These SMILES strings can be converted to molecular graphs using RDKit tool[4], which can be then fed into the GNN module. For targets, their amino acid sequences are collected from UniProt and can be used by the CNN module without any pre-processing.

Baselines: We mainly compare with tensor-based models that can learn the drug-target-disease data. (1) CP and Tucker [Kolda and Bader, 2009]: both are multilinear models. (2) nTucker [Zhe et al., 2016]: a nonlinear Tucker based on Gaussian process. (3) CoSTCo [Liu et al., 2019]: a recent CNN-based tensor model. (4) CMTF [Acar et al., 2011]: a tensor-matrix model regarding auxiliary information as feature matrices. (5) TFAI [Narita et al., 2012]: a tensor model with mode regularization. (6) AirCP [Ge et al., 2016]: a tensor-matrix model with graph regularization. (7) NTF [Wu et al., 2019]: a neural network with MLP. (8) Rubik [Wang et al., 2015]: a tensor model with non-negativity and sparsity constraints. (9) DTD [Chen and Li, 2019f]: a recent tensor-matrix model in drug discovery.

Parameter Settings: For tensor-matrix models CMTF, TEAI, AirCP, Rubik, and DTD, the feature matrices for auxiliary information are constructed using feature engineering as [Zheng et al., 2013, Chen and Li, 2019f], The parameter settings for all the baselines are carefully tuned to achieve optimal performance. For NeurTN,

[1]http://ctdbase.org/downloads/
[2]https://www.drugbank.ca/
[3]https://www.uniprot.org/
[4]http://rdkit.org/

the embedding size r in Eq.(4) is searched in $[16, 32, 64, 128]$. For MLP and CTL, we both employ three hidden layers with dropout ratio $\rho = 0.3$ and each layer sequentially decreases the half size of inputs. Our models are built upon PyTorch[5] with Adam [Kingma and Ba, 2015] optimizer. We search the batch size and the learning rate within $\{128, 256, 512, 1024\}$ and $\{0.001, 0.005, 0.01, 0.05, 0.1\}$, respectively. We use grid-based search to find the best parameter settings. We tune model parameters using validation set and terminate training if the performance does not improve for 100 epochs.

Evaluation Protocols: We randomly split the dataset into 80% training, 10% validation, and 10% test sets. The validation set is used for tuning hyper-parameters and the final results are conducted on the test set. To better construct negative test triplets, we adopt similar procedures as [Ge et al., 2016, Chen and Li, 2019f, Yang et al., 2015, Bordes et al., 2013]: 1) Scenario 1 (random sample): for each positive test triplet (i, j, k), we randomly sample a negative triplet (i', j', k') such that (i', j', k') is unobserved, i.e., $\boldsymbol{\mathcal{X}}_{i'j'k'} = 0$; 2) Scenario 2 (sample drug mode): we corrupt the drug mode by replacing the drug i with a new drug i' so that the (i', j, k) is unobserved; 3) Scenario 3 (sample target mode): we corrupt the triplet (i, j, k) by (i, j', k) in the target mode; 4) Scenario 4 (sample disease mode): the triplet (i, j, k) is replaced with (i, j, k'). In addition, we apply *filtered settings* [Bordes et al., 2013] such that those test negative samples will not appear in the training step.

To evaluate final results, we adopt two widely used top-n metrics: Hit@n and NDCG@n [He et al., 2017, Bordes et al., 2013]. Moreover, we use the strategy in [Bordes et al., 2013] to avoid heavy computation on all triplets. For example, in Scenario 1, we randomly generate 100 negative samples (i', j', k') for each test (i, j, k). Based on the ranking of these triplets, Hit@n and NDCG@n can be evaluated.

[5]https://pytorch.org/

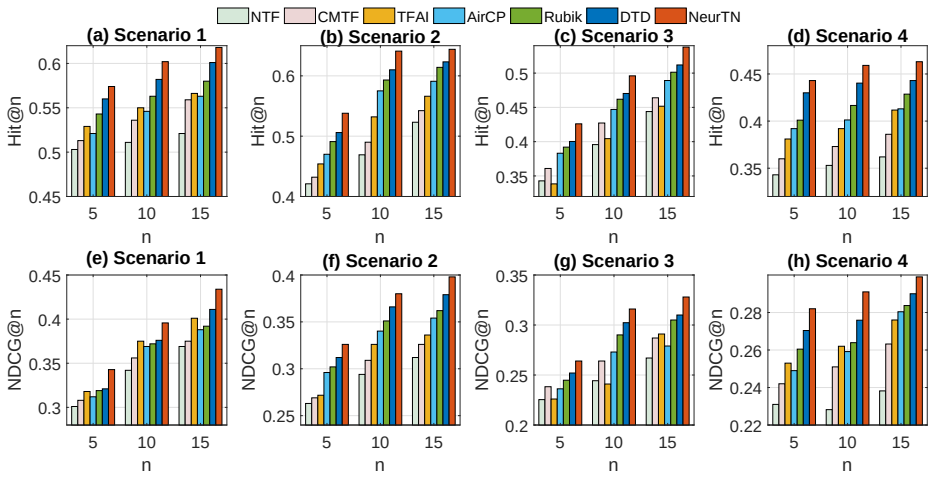

Figure 6.2: Evaluation of top-n performance for different scenarios in terms of Hit@n (a-d) NDCG@n (e-h).

Table 6.1: Results of different methods without auxiliary information.

Model	Hit@5	NDCG@5	Hit@10	NDCG@10	Hit@15	NDCG@15
CP	0.412	0.201	0.441	0.243	0.462	0.251
Tucker	0.423	0.212	0.452	0.249	0.459	0.254
nTucker	0.438	0.223	0.463	0.257	0.467	0.261
CoSTCo	0.437	0.230	0.461	0.259	0.469	0.266
MLP	0.428	0.219	0.456	0.255	0.461	0.258
GCP	0.459	0.241	0.478	0.262	0.473	0.269
CTL	0.462	0.250	0.482	0.269	0.480	0.271
NeurTN	**0.475**	**0.259**	**0.491**	**0.277**	**0.486**	**0.279**

6.4.2 Effect of Neural Tensor Models (RQ1)

The proposed MLP (Eq.(6.5)), GCP (Eq.(6.7)), CTL (Eq.(6.10)), and the unified NeurTN (Eq.(6.11)) are able to capture nonlinear feature interactions. In this part, we compare them with baselines CP, Tucker, nTucker, and CoSTCo, which are pure tensor machines without any auxiliary data of drugs or targets. For fair comparison, we only use the one-hot features in Eq.(6.3) as inputs for the proposed models. Due to page limitation, we only show the top-n performances for Scenario 1, and similar trends can be observed under different scenarios. Table 6.2 shows the results *w.r.t.* Hit@n and NDCG@n.

First, nonlinear tensor models consistently outperform the multilinear CP and

Tucker models. For example, nTucker performs better than Tucker due to its nonlinear Gaussian process; GCP outperforms the CP model and gains average improvements of 7.39% on Hit@n and 11.63% on NDCG@n. These improvements are statistically significant and mainly stem from the powerful representation of neural network. Second, CoSTCo is comparable to nTucker and MLP but worse than GCP. CoSTCo adopts a CNN to perform convolution. Nevertheless, the drug-target-disease data do not have spatial locality. As such, CoSTCo may be insufficient to capture nonlinear patterns for biological data. Third, the performances of CTL are better than GCP, implying a good representations of a deeper neural network. Finally, NeurTN achieves the best performances. Presumably this owes to the high expressiveness of fusing wide and deep components.

6.4.3 Overall Performance Comparison (RQ2)

Now we compare the overall performance of NeurTN with the baselines of interest. The results of CP, Tucker, nTucker, and CoSTCo are omitted due to their inferior performances without auxiliary data. Figure 6.2 shows the performance of Top-n recommendations, where n ranges from 5 to 15.

As can be seen, NeurTN achieves the best performance over all the comparison methods across different scenarios. In addition, we have the following observations. First, the coupled tensor-matrix factorization methods (e.g., CMTF, TFAI, AirCP, Rubik, and DTD) that integrate auxiliary information of drugs and targets achieve better performance than NTF, indicating the important contribution of those auxiliary information in drug discovery. By integrating valuable domain knowledge, these tensor models keep high performance even with very sparse observed data.

Second, our proposed NeurTN outperforms coupled tensor-matrix factorization methods in all scenarios. The superior performance of NeurTN mainly benefits from its deep neural networks. It is intuitive that neural networks would have stronger abil-

ity to fit the data, while the multilinear assumptions in coupled tensor-matrix factorizations do not. More importantly, NeurTN utilizes geometric neural networks GNN and CNN, which can learn features from molecular graphs and protein sequences in the training process. Such end-to-end representation learning can potentially obtain more interpretable data-driven features instead of predefining hand-crafted feature matrices in coupled tensor-matrix factorizations.

6.4.4 Importance of Components (RQ3)

To further understand the importance of each component in our neural networks, we perform some ablation studies. Table 6.2 shows the performance of our default method and its variants in the case of Scenario 1.

Our results are summarized as follows: (1) Remove MLP: Without MLP layers, we find that the performance is slightly worse. Although the black-box nature of MLP, its hierarchical structure is still helpful to learn more complex interactions; (2) Remove GCP: Not surprisingly, the results are worse than the default method. This suggests that the GCP can capture triple-wise feature interactions in a nonlinear fashion; (3) Remove CTL: This variant substantially decreases the overall performance with a large margin, verifying the effectiveness the CTL neural network in capturing the useful feature interactions from feature outer product space; (4) Remove GNN: The chemical structure of a drug determines its pharmacological activity. Removing

Table 6.2: Ablation analysis on our variant models. '↓' means a severe performance drop.

Architecture	Hit@5	NDCG@5	Hit@10	NDCG@10
(0) Default	0.574	0.343	0.602	0.396
(1) Remove MLP	0.558	0.337	0.571	0.380
(2) Remove GCP	0.533↓	0.335	0.553↓	0.371↓
(3) Remove CTL	0.527↓	0.329↓	0.544↓	0.362↓
(4) Remove GNN	0.515↓	0.317↓	0.526↓	0.341 ↓
(5) Remove CNN	0.519↓	0.324↓	0.530↓	0.346 ↓

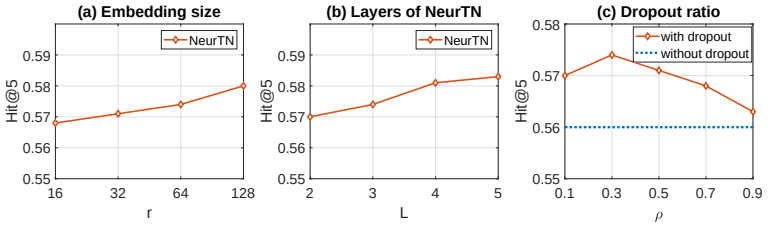

Figure 6.3: (a) The impact of embedding size r. (b) The impact of the number of layers L. (c) The impact of dropout ratio ρ.

GNN thus decreases the overall performance; (5) Remove CNN: Similar, amino acid sequences determine the therapeutic function of targets. Deleting the CNN module hurts the performance.

Embedding Size of NeurTN: The embedding size r in Eq.(6.4) affects the representation ability of NeurTN. We vary r within $[16, 32, 64, 128]$. As shown in Figure 6.3(a), NeurTN benefits from a large embedding size in Scenario 1. Results on other scenarios have similar trends and are omitted here.

Layers of NeurTN: We also conduct experiments to see whether using a deeper network is beneficial to the learning task. To this end, we vary the number of layers in the NeurTN within $L = [2, 3, 4, 5]$. As shown in Figure 6.3(b), stacking more layers gradually enhances the performance. We attribute the improvements to the usage of stacking more layers to model complex drug-target-disease interactions.

Dropout Regularization: Figure 6.3(c) shows the performances of NeurTN *w.r.t.* dropout ratio. Our results show that dropout offers better performance. Specifically, using a dropout ratio $\rho \approx 0.3$ achieves an optimal accuracy.

6.5 Conclusion

A critical step in personalized medicine is to understand drugs' MoAs by exploring the biological interactions among drugs, targets, and diseases in human metabolic

systems. Here we present a novel NeurTN, which seamlessly combines the tensor factorization and deep neural network to capture the nonlinear relationships among health data. Extensive results show the superiority of NeurTN against other counterparts. In the future, we aim to further incorporate the auxiliary information of diseases under the NeurTN.

Chapter 7

Future Work

Network mining for drug discovery is an active research area. In this dissertation, we presented several approaches to perform network analysis for drug repositioning, drug combinations, and drug-target-disease interactions in human metabolic systems. Our studies open up opportunities to use large-scale omics data to predict drugs' MoAs in pharmacological studies. Here we also raise some challenging and promising directions to be further explored in the future.

- **Multi-Network Integration**: Integrating data from various sources/domains, is one of the popular topics in modern machine learning systems. In the past few decades, significant amount of genomic and proteomic data have been accumulated. Those heterogeneous data sources, such as drugs' side-effect, drugs' ligand binding sites, targets' Gene Ontology annotations, diseases' pathway, MicroRNAs, and diseases' Human Phenotype Ontology, can also be integrated to provide a wealth of information for inferring drugs' MoAs. One possible strategy is to construct those heterogeneous data as networks and perform the task of multi-network analysis by analyzing the dependencies of networks in drug discovery.

- **Scalable Network Mining**: Biological data, particularly the large-scale omics

data, pose some of significant computational challenges in real-world applications. This problem is even more challenging in the area of multi-network mining integration. Nowadays it is crucial to develop a machine learning model to handle networks with millions or billions of nodes with reasonable time and storage. To address the time/space challenges posed by big data, we plan to develop some scalable machine learning algorithms, such as careful selection of training data using sampling techniques or *learning to learn*, inspired by human meta-cognition. These strategies may sacrificed accuracy but will earn dramatic speedup for training process. In addition, effective parallelization is a potential guiding principle for massive emerging datasets.

Bibliography

[Absil et al., 2009] Absil, P.-A., Mahony, R., and Sepulchre, R. (2009). *Optimization algorithms on matrix manifolds*. Princeton University Press.

[Acar et al., 2011] Acar, E., Kolda, T. G., and Dunlavy, D. M. (2011). All-at-once optimization for coupled matrix and tensor factorizations. *arXiv preprint arXiv:1105.3422*.

[Agarwal et al., 2006] Agarwal, S., Branson, K., and Belongie, S. (2006). Higher order learning with graphs. In *Proceedings of the 23rd international conference on Machine learning*, pages 17–24. ACM.

[Althouse et al., 2015] Althouse, B. M., Scarpino, S. V., Meyers, L. A., Ayers, J. W., Bargsten, M., Baumbach, J., Brownstein, J. S., Castro, L., Clapham, H., Cummings, D. A., et al. (2015). Enhancing disease surveillance with novel data streams: challenges and opportunities. *EPJ Data Science*, 4(1):17.

[Altshuler et al., 2000] Altshuler, D., Daly, M., and Kruglyak, L. (2000). Guilt by association. *Nature genetics*, 26(2):135–138.

[Araujo et al., 2017] Araujo, M., Mejova, Y., Weber, I., and Benevenuto, F. (2017). Using facebook ads audiences for global lifestyle disease surveillance: Promises and limitations. In *WebSci*.

[Bahargam and Papalexakis, 2018] Bahargam, S. and Papalexakis, E. (2018). Constrained coupled matrix-tensor factorization and its application in pattern and topic detection. In *ASONAM*.

[Bansal et al., 2014] Bansal, M., Yang, J., Karan, C., Menden, M. P., Costello, J. C., Tang, H., Xiao, G., Li, Y., Allen, J., Zhong, R., et al. (2014). A community computational challenge to predict the activity of pairs of compounds. *Nature biotechnology*, 32(12):1213.

[Barretina et al., 2012] Barretina, J., Caponigro, G., Stransky, N., Venkatesan, K., Margolin, A. A., Kim, S., Wilson, C. J., Lehár, J., Kryukov, G. V., Sonkin, D., et al. (2012). The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603.

[Bhadra et al., 2017] Bhadra, S., Kaski, S., and Rousu, J. (2017). Multi-view kernel completion. *Machine Learning*, 106(5):713–739.

[Bleakley and Yamanishi, 2009] Bleakley, K. and Yamanishi, Y. (2009). Supervised prediction of drug–target interactions using bipartite local models. *Bioinformatics*, 25(18):2397–2403.

[Bokhari et al., 2003] Bokhari, S. U., Gopal, U. M., and Duckworth, W. C. (2003). Beneficial effects of a glyburide/metformin combination preparation in type 2 diabetes mellitus. *The American journal of the medical sciences*, 325(2):66–69.

[Bordes et al., 2013] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *NeurIPS*.

[Borisy et al., 2003] Borisy, A. A., Elliott, P. J., Hurst, N. W., Lee, M. S., Lehár, J., Price, E. R., Serbedzija, G., Zimmermann, G. R., Foley, M. A., Stockwell, B. R., et al. (2003). Systematic discovery of multicomponent therapeutics. *Proceedings of the National Academy of Sciences*, 100(13):7977–7982.

[Boumal et al., 2014] Boumal, N., Mishra, B., Absil, P.-A., and Sepulchre, R. (2014). Manopt, a matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1):1455–1459.

[Boyd et al., 2011] Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122.

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

[Brunner and Van Driel, 2004] Brunner, H. G. and Van Driel, M. A. (2004). From syndrome families to functional genomics. *Nature Reviews Genetics*, 5(7):545–551.

[Campillos et al., 2008] Campillos, M., Kuhn, M., Gavin, A.-C., Jensen, L. J., and Bork, P. (2008). Drug target identification using side-effect similarity. *Science*, 321(5886):263–266.

[Caniza et al., 2015] Caniza, H., Romero, A. E., and Paccanaro, A. (2015). A network medicine approach to quantify distance between hereditary disease modules on the interactome. *Scientific reports*, 5:17658.

[Cao et al., 2017] Cao, B., He, L., Wei, X., Xing, M., Yu, P. S., Klumpp, H., and Leow, A. D. (2017). t-bne: Tensor-based brain network embedding. In *SIAM*.

[Capra et al., 2009] Capra, J. A., Laskowski, R. A., Thornton, J. M., Singh, M., and Funkhouser, T. A. (2009). Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3d structure. *PLoS Comput Biol*, 5(12):e1000585.

[Capuzzi et al., 2018] Capuzzi, S. J., Thornton, T. E., Liu, K., Baker, N., Lam, W. I., O'Banion, C. P., Muratov, E. N., Pozefsky, D., and Tropsha, A. (2018). Chemotext: A publicly available web server for mining drug–target–disease relationships in pubmed. *Journal of chemical information and modeling*, 58(2):212–218.

[Carroll and Chang, 1970] Carroll, J. D. and Chang, J.-J. (1970). Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika*, 35(3):283–319.

[Chen et al., 2020] Chen, H., Cheng, F., and Li, J. (2020). idrug: Integration of drug repositioning and drug-target prediction via cross-network embedding. *PLOS Computational Biology*, 16(7):1–20.

[Chen et al., 2019] Chen, H., Iyengar, S. K., and Li, J. (2019). Large-scale analysis of drug combinations by integrating multiple heterogeneous information networks. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 67–76.

[Chen and Li, 2017a] Chen, H. and Li, J. (2017a). A flexible and robust multi-source learning algorithm for drug repositioning. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 510–515. ACM.

[Chen and Li, 2017b] Chen, H. and Li, J. (2017b). Learning multiple similarities of users and items in recommender systems. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 811–816. IEEE.

[Chen and Li, 2018a] Chen, H. and Li, J. (2018a). Drugcom: Synergistic discovery of drug combinations using tensor decomposition. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 899–904. IEEE.

[Chen and Li, 2018b] Chen, H. and Li, J. (2018b). Exploiting structural and temporal evolution in dynamic link prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 427–436. ACM.

[Chen and Li, 2019a] Chen, H. and Li, J. (2019a). Adversarial tensor factorization for context-aware recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 363–367.

[Chen and Li, 2019b] Chen, H. and Li, J. (2019b). Collaborative ranking tags and items via cross-domain recommendation. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE.

[Chen and Li, 2019c] Chen, H. and Li, J. (2019c). Collective tensor completion with multiple heterogeneous side information. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE.

[Chen and Li, 2019d] Chen, H. and Li, J. (2019d). Data poisoning attacks on cross-domain recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2177–2180.

[Chen and Li, 2019e] Chen, H. and Li, J. (2019e). Finding stable clustering for noisy data via structure-aware representation. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE.

[Chen and Li, 2019f] Chen, H. and Li, J. (2019f). Modeling relational drug-target-disease interactions via tensor factorization with multiple web sources. In *WWW*.

[Chen et al., 2015] Chen, H., Zhang, H., Zhang, Z., Cao, Y., and Tang, W. (2015). Network-based inference methods for drug repositioning. *Computational and mathematical methods in medicine*, 2015.

[Chen et al., 2012] Chen, X., Liu, M.-X., and Yan, G.-Y. (2012). Drug–target interaction prediction by random walk on the heterogeneous network. *Molecular BioSystems*, 8(7):1970–1978.

[Chen et al., 2016] Chen, X., Ren, B., Chen, M., Wang, Q., Zhang, L., and Yan, G. (2016). Nllss: predicting synergistic drug combinations based on semi-supervised learning. *PLoS Comput Biol*, 12(7):e1004975.

[Cheng et al., 2012] Cheng, F., Liu, C., Jiang, J., Lu, W., Li, W., Liu, G., Zhou, W., Huang, J., and Tang, Y. (2012). Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS computational biology*, 8(5):e1002503.

[Cheng et al., 2016] Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al. (2016). Wide & deep learning for recommender systems. In *DLRS*.

[Chiang and Butte, 2009] Chiang, A. P. and Butte, A. J. (2009). Systematic evaluation of drug-disease relationships to identify leads for novel drug uses. *Clinical pharmacology and therapeutics*, 86(5):507.

[Cichocki et al., 2015] Cichocki, A., Mandic, D., De Lathauwer, L., Zhou, G., Zhao, Q., Caiafa, C., and Phan, H. A. (2015). Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32(2):145–163.

[De Lathauwer et al., 2000] De Lathauwer, L., De Moor, B., and Vandewalle, J. (2000). A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278.

[Deng et al., 2015] Deng, Y., Gao, L., Wang, B., and Guo, X. (2015). Hposim: an r package for phenotypic similarity measure and enrichment analysis based on the human phenotype ontology. *PloS one*, 10(2):e0115692.

[Ding et al., 2005] Ding, C., He, X., and Simon, H. D. (2005). On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 606–610. SIAM.

[Ding et al., 2006] Ding, C., Li, T., Peng, W., and Park, H. (2006). Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135. ACM.

[Dudley et al., 2011] Dudley, J. T., Deshpande, T., and Butte, A. J. (2011). Exploiting drug–disease relationships for computational drug repositioning. *Briefings in bioinformatics*, page bbr013.

[Duvenaud et al., 2015] Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. In *NeurIPS*.

[Edelman et al., 1998] Edelman, A., Arias, T. A., and Smith, S. T. (1998). The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353.

[Eguale et al., 2016] Eguale, T., Buckeridge, D. L., Verma, A., Winslade, N. E., Benedetti, A., Hanley, J. A., and Tamblyn, R. (2016). Association of off-label drug use and adverse drug events in an adult population. *JAMA internal medicine*, 176(1):55–63.

[Ezzat et al., 2018] Ezzat, A., Wu, M., Li, X.-L., and Kwoh, C.-K. (2018). Computational prediction of drug–target interactions using chemogenomic approaches: an empirical survey. *Briefings in bioinformatics*.

[Ezzat et al., 2017] Ezzat, A., Zhao, P., Wu, M., Li, X.-L., and Kwoh, C.-K. (2017). Drug-target interaction prediction with graph regularized matrix factorization. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 14(3):646–656.

[Fang et al., 2015] Fang, X., Pan, R., Cao, G., He, X., and Dai, W. (2015). Personalized tag recommendation through nonlinear tensor factorization using gaussian kernel. In *AAAI*.

[Foucquier and Guedj, 2015] Foucquier, J. and Guedj, M. (2015). Analysis of drug combinations: current methodological landscape. *Pharmacology research & perspectives*, 3(3).

[Friedman et al., 2001] Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.

[Garcia-Albornoz and Nielsen, 2015] Garcia-Albornoz, M. and Nielsen, J. (2015). Finding directionality and gene-disease predictions in disease associations. *BMC systems biology*, 9(1):35.

[Ge et al., 2016] Ge, H., Caverlee, J., Zhang, N., and Squicciarini, A. (2016). Uncovering the spatio-temporal dynamics of memes in the presence of incomplete information. In *CIKM*.

[Getoor and Diehl, 2005] Getoor, L. and Diehl, C. P. (2005). Link mining: a survey. *Acm Sigkdd Explorations Newsletter*, 7(2):3–12.

[Gonzalez et al., 2007] Gonzalez, G., Uribe, J. C., Tari, L., Brophy, C., and Baral, C. (2007). Mining gene-disease relationships from biomedical literature: weighting protein–protein interactions and connectivity measures. In *Biocomputing 2007*, pages 28–39. World Scientific.

[Gottlieb et al., 2011] Gottlieb, A., Stein, G. Y., Ruppin, E., and Sharan, R. (2011). Predict: a method for inferring novel drug indications with application to personalized medicine. *Molecular systems biology*, 7(1):496.

[Graul et al., 2014] Graul, A., Cruces, E., and Stringer, M. (2014). The year's new drugs & biologics, 2013: Part i. *Drugs of today (Barcelona, Spain: 1998)*, 50(1):51–100.

[Greco and Vicent, 2009] Greco, F. and Vicent, M. J. (2009). Combination therapy: opportunities and challenges for polymer–drug conjugates as anticancer nanomedicines. *Advanced drug delivery reviews*, 61(13):1203–1213.

[Harshman et al., 1970] Harshman, R. A. et al. (1970). Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis.

[Haupt et al., 2013] Haupt, V. J., Daminelli, S., and Schroeder, M. (2013). Drug promiscuity in pdb: protein binding site similarity is key. *PLoS one*, 8(6):e65894.

[Hauser et al., 2017] Hauser, A. S., Attwood, M. M., Rask-Andersen, M., Schiöth, H. B., and Gloriam, D. E. (2017). Trends in gpcr drug discovery: new agents, targets and indications. *Nature Reviews Drug Discovery*, 16(12):829.

[He et al., 2019] He, H., Henderson, J., and Ho, J. C. (2019). Distributed tensor decomposition for large scale health analytics. In *WWW*.

[He et al., 2018] He, X., Du, X., Wang, X., Tian, F., Tang, J., and Chua, T.-S. (2018). Outer product-based neural collaborative filtering. In *IJCAI*.

[He et al., 2017] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. (2017). Neural collaborative filtering. In *WWW*.

[Hopkins, 2008] Hopkins, A. L. (2008). Network pharmacology: the next paradigm in drug discovery. *Nature chemical biology*, 4(11):682–690.

[Horn et al., 1990] Horn, R. A., Horn, R. A., and Johnson, C. R. (1990). *Matrix analysis*. Cambridge university press.

[Hu, 2018] Hu, K. (2018). *Methods and Analyses in the Study of Human DNA Methylation*. PhD thesis, Case Western Reserve University.

[Hu et al., 2015] Hu, K., Ting, A. H., and Li, J. (2015). Bspat: a fast online tool for dna methylation co-occurrence pattern analysis based on high-throughput bisulfite sequencing data. *BMC bioinformatics*, 16(1):220.

[Huang et al., 2014] Huang, L., Li, F., Sheng, J., Xia, X., Ma, J., Zhan, M., and Wong, S. T. (2014). Drugcomboranker: drug combination discovery based on target network analysis. *Bioinformatics*, 30(12):i228–i236.

[Iadevaia et al., 2010] Iadevaia, S., Lu, Y., Morales, F. C., Mills, G. B., and Ram, P. T. (2010). Identification of optimal drug combinations targeting cellular networks: integrating phospho-proteomics and computational network analysis. *Cancer research*, 70(17):6704–6714.

[Iwata et al., 2015] Iwata, H., Sawada, R., Mizutani, S., Kotera, M., and Yamanishi, Y. (2015). Large-scale prediction of beneficial drug combinations using drug efficacy and target profiles. *Journal of chemical information and modeling*, 55(12):2705–2716.

[Jaeger et al., 2017] Jaeger, S., Igea, A., Arroyo, R., Alcalde, V., Canovas, B., Orozco, M., Nebreda, A. R., and Aloy, P. (2017). Quantification of pathway cross-talk reveals novel synergistic drug combinations for breast cancer. *Cancer research*, 77(2):459–469.

[Kasai and Mishra, 2016] Kasai, H. and Mishra, B. (2016). Low-rank tensor completion: a riemannian manifold preconditioning approach. In *ICML*.

[Kingma and Ba, 2015] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *ICLR*.

[Kolda and Bader, 2009] Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, 51(3):455–500.

[Kuhn et al., 2015] Kuhn, M., Letunic, I., Jensen, L. J., and Bork, P. (2015). The sider database of drugs and side effects. *Nucleic acids research*, page gkv1075.

[Kwak et al., 2010] Kwak, H., Lee, C., Park, H., and Moon, S. (2010). What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. AcM.

[Lebedev et al., 2015] Lebedev, V., Ganin, Y., Rakhuba, M., Oseledets, I., and Lempitsky, V. (2015). Speeding-up convolutional neural networks using fine-tuned cp-decomposition. In *ICLR*.

[Lee and Seung, 1999] Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788.

[Lee and Seung, 2001] Lee, D. D. and Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562.

[Lehár et al., 2009] Lehár, J., Krueger, A. S., Avery, W., Heilbut, A. M., Johansen, L. M., Price, E. R., Rickles, R. J., Short Iii, G. F., Staunton, J. E., Jin, X., et al. (2009). Synergistic drug combinations tend to improve therapeutically relevant selectivity. *Nature biotechnology*, 27(7):659–666.

[Lewis et al., 2011] Lewis, S. N., Nsoesie, E., Weeks, C., Qiao, D., and Zhang, L. (2011). Prediction of disease and phenotype associations from genome-wide association studies. *PLoS One*, 6(11):e27175.

[Li et al., 2011] Li, J., Gong, B., Chen, X., Liu, T., Wu, C., Zhang, F., Li, C., Li, X., Rao, S., and Li, X. (2011). Dosim: An r package for similarity between diseases based on disease ontology. *BMC bioinformatics*, 12(1):266.

[Li et al., 2015a] Li, J., Zheng, S., Chen, B., Butte, A. J., Swamidass, S. J., and Lu, Z. (2015a). A survey of current trends in computational drug repositioning. *Briefings in bioinformatics*, 17(1):2–12.

[Li et al., 2015b] Li, P., Huang, C., Fu, Y., Wang, J., Wu, Z., Ru, J., Zheng, C., Guo, Z., Chen, X., Zhou, W., et al. (2015b). Large-scale exploration and analysis of drug combinations. *Bioinformatics*, 31(12):2007–2016.

[Li and Chen, 2009] Li, X. and Chen, H. (2009). Recommendation as link prediction: a graph kernel-based machine learning approach. In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pages 213–216. ACM.

[Lin, 2007] Lin, C.-J. (2007). Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779.

[Lin et al., 2011] Lin, Z., Liu, R., and Su, Z. (2011). Linearized alternating direction method with adaptive penalty for low-rank representation. In *NIPS*.

[Lindsley, 2014] Lindsley, C. W. (2014). New statistics on the cost of new drug development and the trouble with cns drugs.

[Liu et al., 2019] Liu, H., Li, Y., Tsang, M., and Liu, Y. (2019). Costco: A neural tensor completion model for sparse tensors. In *KDD*.

[Liu et al., 2013a] Liu, J., Musialski, P., Wonka, P., and Ye, J. (2013a). Tensor completion for estimating missing values in visual data. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):208–220.

[Liu et al., 2013b] Liu, J., Wang, C., Gao, J., and Han, J. (2013b). Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 252–260. SIAM.

[Liu et al., 2018] Liu, Y., Safavi, T., Dighe, A., and Koutra, D. (2018). Graph summarization methods and applications: A survey. *ACM Computing Surveys (CSUR)*, 51(3):62.

[LoRusso et al., 2012] LoRusso, P. M., Canetta, R., Wagner, J. A., Balogh, E. P., Nass, S. J., Boerner, S. A., and Hohneker, J. (2012). Accelerating cancer therapy development: the importance of combination strategies and collaboration. summary of an institute of medicine workshop. *Clinical Cancer Research*, 18(22):6101–6109.

[Lu et al., 2016] Lu, C., Yan, S., and Lin, Z. (2016). Convex sparse spectral clustering: Single-view to multi-view. *IEEE Transactions on Image Processing*, 25(6):2833–2843.

[Luo et al., 2016] Luo, H., Wang, J., Li, M., Luo, J., Peng, X., Wu, F.-X., and Pan, Y. (2016). Drug repositioning based on comprehensive similarity measures and bi-random walk algorithm. *Bioinformatics*, page btw228.

[Mahé et al., 2006] Mahé, P., Ralaivola, L., Stoven, V., and Vert, J.-P. (2006). The pharmacophore kernel for virtual screening with support vector machines. *Journal of Chemical Information and Modeling*, 46(5):2003–2014.

[Martínez et al., 2017] Martínez, V., Berzal, F., and Cubero, J.-C. (2017). A survey of link prediction in complex networks. *ACM Computing Surveys (CSUR)*, 49(4):69.

[Martínez et al., 2015] Martínez, V., Navarro, C., Cano, C., Fajardo, W., and Blanco, A. (2015). Drugnet: Network-based drug–disease prioritization by integrating heterogeneous data. *Artificial intelligence in medicine*, 63(1):41–49.

[Menden et al., 2019] Menden, M. P., Wang, D., Mason, M. J., Szalai, B., Bulusu, K. C., Guan, Y., Yu, T., Kang, J., Jeon, M., Wolfinger, R., et al. (2019). Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen. *Nature communications*, 10(1):2674.

[Narita et al., 2012] Narita, A., Hayashi, K., Tomioka, R., and Kashima, H. (2012). Tensor factorization using auxiliary information. *Data Mining and Knowledge Discovery*, 25.

[Nascimento et al., 2016] Nascimento, A. C., Prudêncio, R. B., and Costa, I. G. (2016). A multiple kernel learning algorithm for drug-target interaction prediction. *BMC bioinformatics*, 17(1):1.

[Ng et al., 2002] Ng, A. Y., Jordan, M. I., and Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In *NIPS*.

[Novikov et al., 2015] Novikov, A., Podoprikhin, D., Osokin, A., and Vetrov, D. P. (2015). Tensorizing neural networks. In *NeurIPS*.

[Pang et al., 2014] Pang, K., Wan, Y.-W., Choi, W. T., Donehower, L. A., Sun, J., Pant, D., and Liu, Z. (2014). Combinatorial therapy discovery using mixed integer linear programming. *Bioinformatics*, 30(10):1456–1463.

[Paul and Dredze, 2013] Paul, M. J. and Dredze, M. (2013). Drug extraction from the web: Summarizing drug experiences with multi-dimensional topic models. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 168–178.

[Paul et al., 2010] Paul, S. M., Mytelka, D. S., Dunwiddie, C. T., Persinger, C. C., Munos, B. H., Lindborg, S. R., and Schacht, A. L. (2010). How to improve r&d productivity: the pharmaceutical industry's grand challenge. *Nature reviews Drug discovery*, 9(3):203–214.

[Perra et al., 2011] Perra, N., Balcan, D., Gonçalves, B., and Vespignani, A. (2011). Towards a characterization of behavior-disease models. *PloS one*, 6(8):e23084.

[Preuer et al., 2017] Preuer, K., Lewis, R. P., Hochreiter, S., Bender, A., Bulusu, K. C., and Klambauer, G. (2017). Deepsynergy: Predicting anti-cancer drug synergy with deep learning. *Bioinformatics*, 1:9.

[Radivojac et al., 2013] Radivojac, P., Clark, W. T., Oron, T. R., Schnoes, A. M., Wittkop, T., Sokolov, A., Graim, K., Funk, C., Verspoor, K., Ben-Hur, A., et al. (2013). A large-scale evaluation of computational protein function prediction. *Nature methods*, 10(3):221.

[Ray et al., 2016] Ray, B., Ghedin, E., and Chunara, R. (2016). Network inference from multimodal data: a review of approaches from infectious disease transmission. *Journal of biomedical informatics*, 64:44–54.

[Rendle et al., 2009] Rendle, S., Balby Marinho, L., Nanopoulos, A., and Schmidt-Thieme, L. (2009). Learning optimal ranking with tensor factorization for tag recommendation. In *KDD*.

[Ricci et al., 2011] Ricci, F., Rokach, L., and Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer.

[Richardson and Domingos, 2002] Richardson, M. and Domingos, P. (2002). Mining knowledge-sharing sites for viral marketing. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 61–70. ACM.

[Robinson et al., 2008] Robinson, P., Kohler, S., Bauer, S., Seelow, D., Horn, D., and Mundlos, S. (2008). The human phenotype ontology: a tool for annotating and analyzing human hereditary disease. *American Journal of Human Genetics*, 83:610–615.

[Rowinsky et al., 1991] Rowinsky, E. K., Gilbert, M., McGuire, W., Noe, D., Grochow, L., Forastiere, A., Ettinger, D., Lubejko, B., Clark, B., and Sartorius, S. (1991). Sequences of taxol and cisplatin: a phase i and pharmacologic study. *Journal of clinical oncology*, 9(9):1692–1703.

[Santos et al., 2017] Santos, R., Ursu, O., Gaulton, A., Bento, A. P., Donadi, R. S., Bologa, C. G., Karlsson, A., Al-Lazikani, B., Hersey, A., Oprea, T. I., et al. (2017). A comprehensive map of molecular drug targets. *Nature reviews Drug discovery*, 16(1):19.

[Sarker and Gonzalez, 2015] Sarker, A. and Gonzalez, G. (2015). Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *Journal of biomedical informatics*, 53:196–207.

[Scholkopf and Smola, 2001] Scholkopf, B. and Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

[Schwikowski et al., 2000] Schwikowski, B., Uetz, P., and Fields, S. (2000). A network of protein–protein interactions in yeast. *Nature biotechnology*, 18(12):1257.

[Shan et al., 2016] Shan, Y., Hoens, T. R., Jiao, J., Wang, H., Yu, D., and Mao, J. (2016). Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *KDD*.

[Shao et al., 2015] Shao, W., He, L., and Philip, S. Y. (2015). Multiple incomplete views clustering via weighted nonnegative matrix factorization with l_ {2, 1} regularization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 318–334. Springer.

[Shashua and Hazan, 2005] Shashua, A. and Hazan, T. (2005). Non-negative tensor factorization with applications to statistics and computer vision. In *Proceedings of the 22nd international conference on Machine learning*, pages 792–799. ACM.

[Shi et al., 2017] Shi, C., Li, Y., Zhang, J., Sun, Y., and Philip, S. Y. (2017). A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):17–37.

[Smith and Waterman, 1981] Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197.

[Socher et al., 2013] Socher, R., Chen, D., Manning, C. D., and Ng, A. (2013). Reasoning with neural tensor networks for knowledge base completion. In *NeurIPS*.

[Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

[Steinbeck et al., 2006] Steinbeck, C., Hoppe, C., Kuhn, S., Floris, M., Guha, R., and Willighagen, E. L. (2006). Recent developments of the chemistry development kit (cdk)-an open-source java library for chemo-and bioinformatics. *Current pharmaceutical design*, 12(17):2111–2120.

[Sun and Han, 2012] Sun, Y. and Han, J. (2012). Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 3(2):1–159.

[Sun et al., 2015] Sun, Y., Sheng, Z., Ma, C., Tang, K., Zhu, R., Wu, Z., Shen, R., Feng, J., Wu, D., Huang, D., et al. (2015). Combining genomic and network characteristics for extended capability in predicting synergistic drugs for cancer. *Nature communications*, 6:8481.

[Takarabe et al., 2012] Takarabe, M., Kotera, M., Nishimura, Y., Goto, S., and Yamanishi, Y. (2012). Drug target prediction using adverse event report systems: a pharmacogenomic approach. *Bioinformatics*, 28(18):i611–i618.

[Tsigelny, 2018] Tsigelny, I. F. (2018). Artificial intelligence in drug combination therapy. *Briefings in bioinformatics*.

[Tsubaki et al., 2018] Tsubaki, M., Tomii, K., and Sese, J. (2018). Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics*, 35(2):309–318.

[Tucker, 1963] Tucker, L. R. (1963). Implications of factor analysis of three-way matrices for measurement of change. *Problems in measuring change*, 15:122–137.

[Tucker, 1966] Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311.

[Van Driel et al., 2006] Van Driel, M. A., Bruggeman, J., Vriend, G., Brunner, H. G., and Leunissen, J. A. (2006). A text-mining analysis of the human phenome. *European journal of human genetics*, 14(5):535–542.

[van Laarhoven et al., 2011] van Laarhoven, T., Nabuurs, S. B., and Marchiori, E. (2011). Gaussian interaction profile kernels for predicting drug–target interaction. *Bioinformatics*, 27(21):3036–3043.

[Wang et al., 2013] Wang, H., Huang, H., Ding, C., and Nie, F. (2013). Predicting protein–protein interactions from multimodal biological data sources via nonnegative matrix tri-factorization. *Journal of Computational Biology*, 20(4):344–358.

[Wang et al., 2017] Wang, Q., Gao, J., and Li, H. (2017). Grassmannian manifold optimization assisted sparse spectral clustering. In *CVPR*.

[Wang et al., 2018] Wang, R., Li, S., Wong, M. H., and Leung, K. S. (2018). Drug-protein-disease association prediction and drug repositioning based on tensor decomposition. In *BIBM*.

[Wang et al., 2014] Wang, W., Yang, S., Zhang, X., and Li, J. (2014). Drug repositioning by integrating target information through a heterogeneous network model. *Bioinformatics*, 30(20):2923–2930.

[Wang et al., 2015] Wang, Y., Chen, R., Ghosh, J., Denny, J. C., Kho, A., Chen, Y., Malin, B. A., and Sun, J. (2015). Rubik: Knowledge guided tensor factorization and completion for health data analytics. In *KDD*.

[Welling and Weber, 2001] Welling, M. and Weber, M. (2001). Positive tensor factorization. *Pattern Recognition Letters*, 22(12):1255–1261.

[Wold et al., 1987] Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

[Wu et al., 2013a] Wu, C., Gudivada, R. C., Aronow, B. J., and Jegga, A. G. (2013a). Computational drug repositioning through heterogeneous network clustering. *BMC systems biology*, 7(5):S6.

[Wu et al., 2019] Wu, X., Shi, B., Dong, Y., Huang, C., and Chawla, N. V. (2019). Neural tensor factorization for temporal interaction learning. In *WSDM*.

[Wu et al., 2013b] Wu, Z., Wang, Y., and Chen, L. (2013b). Network-based drug repositioning. *Molecular BioSystems*, 9(6):1268–1281.

[Xie et al., 2013] Xie, J., Kelley, S., and Szymanski, B. K. (2013). Overlapping community detection in networks: The state-of-the-art and comparative study. *Acm computing surveys (csur)*, 45(4):43.

[Xie et al., 2009] Xie, L., Xie, L., and Bourne, P. E. (2009). A unified statistical model to support local sequence order independent similarity searching for ligand-binding sites and its application to genome-based drug discovery. *Bioinformatics*, 25(12):i305–i312.

[Xu et al., 2017] Xu, Q., Xiong, Y., Dai, H., Kumari, K. M., Xu, Q., Ou, H.-Y., and Wei, D.-Q. (2017). Pdc-sgb: Prediction of effective drug combinations using a stochastic gradient boosting algorithm. *Journal of theoretical biology*, 417:1–7.

[Xu et al., 2003] Xu, W., Liu, X., and Gong, Y. (2003). Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM.

[Xu et al., 2012] Xu, Z., Yan, F., and Qi, Y. (2012). Infinite tucker decomposition: Nonparametric bayesian models for multiway data analysis. In *ICML*.

[Yamanishi, 2014] Yamanishi, Y. (2014). Predicting drug-target interaction networks from the integration of chemical, genomic, and pharmacological spaces. In *International Symposium on Tumor Biology in Kanazawa & Symposium on Drug Discoverry in Academics: Program & Abstracts*, number 2014, pages 38–39.

[Yamanishi et al., 2008] Yamanishi, Y., Araki, M., Gutteridge, A., Honda, W., and Kanehisa, M. (2008). Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, 24(13):i232–i240.

[Yang et al., 2015] Yang, B., Yih, W.-t., He, X., Gao, J., and Deng, L. (2015). Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*.

[Yu et al., 2014] Yu, G., Wang, L.-G., Yan, G.-R., and He, Q.-Y. (2014). Dose: an r/bioconductor package for disease ontology semantic and enrichment analysis. *Bioinformatics*, 31(4):608–609.

[Zhang et al., 2016] Zhang, H., Reddi, S. J., and Sra, S. (2016). Riemannian svrg: Fast stochastic optimization on riemannian manifolds. In *NIPS*.

[Zhang et al., 2017] Zhang, Q., Perra, N., Perrotta, D., Tizzoni, M., Paolotti, D., and Vespignani, A. (2017). Forecasting seasonal influenza fusing digital indicators and a mechanistic disease model. In *WWW*.

[Zhao et al., 2011] Zhao, X.-M., Iskar, M., Zeller, G., Kuhn, M., Van Noort, V., and Bork, P. (2011). Prediction of drug combinations by integrating molecular and pharmacological data. *PLoS Comput Biol*, 7(12):e1002323.

[Zhe et al., 2016] Zhe, S., Zhang, K., Wang, P., Lee, K.-c., Xu, Z., Qi, Y., and Ghahramani, Z. (2016). Distributed flexible nonlinear tensor factorization. In *NeurIPS*.

[Zheng et al., 2013] Zheng, X., Ding, H., Mamitsuka, H., and Zhu, S. (2013). Collaborative matrix factorization with multiple similarities for predicting drug-target interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1025–1033. ACM.

[Zitnik et al., 2018] Zitnik, M., Agrawal, M., and Leskovec, J. (2018). Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):457–466.

[Zou et al., 2018] Zou, B., Lampos, V., and Cox, I. (2018). Multi-task learning improves disease models from web search. In *WWW*.