Establishing Advanced Deep Learning Models for Predicting Drug Side Effects

Kyoto University

NGUYEN DUC ANH

Dedication

This thesis is dedicated to my father NGUYEN Duc-Chau, my mother TRAN Thi-Minh, and my brother NGUYEN Duc-Binh for continuously supporting my studies. It is the fulfillment of a promise that I made a long time ago to my family.

NGUYEN Duc-Anh
Kyoto, Japan

Abstract

A drug side effect or an adverse drug reaction is a response to a medicine that is noxious and unintended occurring at doses, which can be a single drug or a drug combination (drug-drug interactions), normally used in humans. Drug side effects are responsible for significant patient morbidity and mortality, costing billions of dollars annually. Hence, determining drug side effects is an important task in pharmacology to guide drug safety. Traditionally, drug side effects are obtained from clinical trials or surveillance reports of released drugs on the market, which are time-consuming and costly. To deal with these disadvantages, machine learning models integrating various kinds of drug data sources have been applied to obtain fast, inexpensive, and highly accurate predictions of drug side effects. The prediction results provide not only potential side effects but also the mechanisms which can support further clinical verification to improve drug side effect studies.

In this thesis, we explore machine learning models used in predicting drug side effects with a focus on deep learning models with the highest prediction performances. Basically, deep learning models aim to learn latent vector representations of drugs in low dimensional spaces which reflect drug properties causing side effects. We analyze the remaining problems in learning latent representations of drugs of the current cutting-edge methods and then propose new advanced models. The contributions of the thesis include 1) we present a comprehensive survey on data resources, tasks, and machine learning models used in drug side effect studies; 2) we present CentSmoothie, a central-smoothing hypergraph neural network for predicting drug-drug interactions, that not only learns representations of drugs but also latent representations of side effects to improve the prediction performances; 3) we present SPARSE for further improving CentSmoothie in terms of prediction accuracy and explaining the potential biological interpretation of the drug-drug interactions. We summarize the organization of the thesis as follows.

In Chapter 1, we introduce the predicting drug side effect problems with relevant background and terminologies.

In Chapter 2, we survey and classify data resources in drug side effects and

machine learning models used on them. Data resources related to drug side effects consist of two types: (i) clinical data and (ii) non-clinical data. The clinical data contains observations of side effects in clinical treatments, which are often electronic health records or records from adverse report systems. The non-clinical data contains information on the chemical, physical, and biological properties of drugs and biological systems. The results showed that the deep learning models integrating both types of data achieved the highest prediction performance on the side effects of each drug, showing the prominence of the deep learning models. In Chapter 3, we present CentSmoothie, a central-smoothing hypergraph neural network for predicting drug-drug interactions (DDI). DDI is usually represented as a graph in that nodes are drugs and edges are interacting drug pairs with side effects as labels. The task is to predict the labels of all pairs of nodes in the DDI graph. Existing work often uses graph neural networks to learn vector representations of drug nodes on the DDI graph and uses them to predict interactions. One drawback of this method is the lack of learning side effect representations. Side effects have complex relationships, for example, co-occurrences. Previous methods often represent each side effect as a one-hot vector indicating the presence of the side effect. This representation considers that side effects are independent, potentially under-utilizing the side effect relationships. Hence, it is necessary to learn representations of both side effects and drugs altogether. To address the above drawback, we propose to encode DDI data with a hypergraph that a node in the hypergraph can be either a drug or a side effect and each hyperedge is a triple of two drugs and a side effect that they cause. CentSmoothie, with the core idea that the side effect is caused by a single combination of the properties of two corresponding drugs, was proposed to learn on the new DDI hypergraph. The experimental results on the largest DDI benchmark dataset showed that CentSmoothie outperformed existing methods with 0.9348 and 0.8749 in AUC (area under the ROC curve) and AUPR (area under the precision-recall curve) while the second-best method was only 0.9044 and 0.8410, respectively.

In Chapter 4, we present SPARSE, a model for learning multiple types of latent combinations of drug-drug interactions. In CentSmoothie, we assumed that the side effect is caused by a single combination of the properties of two

corresponding drugs. However, in reality, a side effect might have multiple, different mechanisms that cannot be represented by a single combination of latent representations of drugs. Furthermore, DDI data is sparse, suggesting that using a sparsity regularization would help to learn better latent representations to improve prediction performances. To solve these remaining problems, we propose SPARSE, which encodes the DDI hypergraph and drug features to latent spaces to learn multiple types of combinations of latent features of drugs and side effects, controlling the model sparsity by a sparse prior. The experimental results on the largest DDI benchmark data showed that SPARSE achieved an AUC of 0.9524 and AUPR of 0.882, which was higher than Cent-Smoothie with 0.9348 and 0.8749. We also validated the prediction results by analyzing the biological properties such as target proteins of the top prediction obtained by the learned latent interactions of SPARSE. For the top 10 cases, we could find relevant references for all cases, suggesting the prominence of prediction and the usefulness of SPARSE in practice.

In Chapter 5, we conclude our work in establishing advanced deep learning models for predicting drug side effects and give some possible future directions to enhance the models.

Acknowledgement

I would like to express my gratefulness to the people with the guidance and support to accomplish my Ph.D.

First of all, I would like to express my sincere gratitude to my supervisor Professor Hiroshi Mamitsuka of Kyoto University for his continuous support and valuable advice. His excellent guidance helped me to improve myself and overcome the difficult time of doing research for my Ph.D. study.

I also would like to sincerely thank Senior Lecture Canh-Hao Nguyen of Kyoto University for his patience in correcting my writing many times and his excellent suggestions for solving difficult problems.

I also thank my fellow labmates Dr. Dai-Hai Nguyen and Dr. Peter Petschner for the interesting discussions and advice, and the laboratory's secretary Junko Yamamoto for her support in doing many documents and procedures.

I would like to thank the Otsuka Toshimi Scholarship Foundation for providing me with the scholarship. I still remember the wonderful moments that we had on the gathering days.

Last but not least, I would like to especially thank my family members, my father NGUYEN Duc-Chau, my mother Tran Thi-Minh, and my brother Nguyen Duc-Binh for their spiritual support during my Ph.D. study and my life.

Publication notes

Chapter 2 is based on a survey paper on data, tasks, and machine learning methods on drug side effect (adverse drug reaction) studies, which appeared in the journal of Briefings in Bioinformatics [Nguyen et al., 2021]. Chapter 3 is based on a paper for a deep learning model that could predict drug-drug interactions using a novel hypergraph neural network namely CentSmoothie, which was uploaded to arXiv [Nguyen et al., 2022a]. Chapter 4 is based on a paper for further improvement of CentSmoothie, which appeared in the journal of Bioinformatics (also the proceedings of the 30th International Conference on Intelligent Systems for Molecular Biology (ISMB 2022)) [Nguyen et al., 2022b]. Altogether, the content of this thesis is based on three papers in the following publication list:

List of publications by the author

1. Duc Anh Nguyen, Canh Hao Nguyen, and Hiroshi Mamitsuka, "A survey on adverse drug reaction studies: data, tasks and machine learning methods", *Briefings in bioinformatics*, p.164-177, 2021. DOI: https://doi.org/10.1093/bib/bbz14.

2. Duc Anh Nguyen, Canh Hao Nguyen, and Hiroshi Mamitsuka, "Central-smoothing hypergraph neural networks for predicting drug-drug interactions", *arXiv*, p.1-11, 2022. DOI:https://doi.org/10.48550/arXiv.2112.07837.

3. Duc Anh Nguyen, Canh Hao Nguyen, Peter Petschner, and Hiroshi Mamitsuka. "SPARSE: a sparse hypergraph neural network for learning multiple types of latent combinations to accurately predict drug-drug interactions". *Bioinformatics*, p.i333-i341, 2022. DOI: https://doi.org/10.1093/bioinformatics/btac250.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computational needs for predicting drug side effects

According to WHO, an adverse drug reaction (ADR) or a drug side effect (side effect for short) is a response to a medicine which is noxious and unintended, and which occurs at doses normally used in humans [WHO, 1972]. In reports of 2011, drug side effects accounted for nearly 6% of total hospitalizations in the USA, which cost billions of dollars and was responsible for significant patient morbidity and mortality [Poudel et al., 2017, Weiss et al., 2013]. Therefore, studies of drug side effects are important in drug discovery.

The traditional methods for obtaining drug side effects often use clinical trials or post-marketing surveillance reports [Hoots et al., 2018]. However, these methods are costly and time-consuming, leading to the need for developing methods to support the process of determining drug side effects.

Nowadays, with the development of technologies and standardization, there exist numerous databases related to drugs and side effects, for example, more than 7 million electronic health records of patients [FDA, 2019], 113 million chemical substances [Kim et al., 2016], biological knowledge for mechanisms of more than 15 thousand drugs [Kanehisa and Goto, 2000, Wishart et al., 2018]. By integrating these various kinds of data, computational methods, especially deep learning, can be used to make highly accurate, inexpensive, and fast drug side effect predictions. These results not only provide potential drug side effects but also potential mechanisms for further clinical verification to enhance drug side effect studies.

Objectives of the Thesis

In light of the need for computational models for predicting drug side effects, we aim to provide a systematic survey of available data resources for deep insights into the information that can be used. We also aim to establish novel prediction models that can provide highly accurate drug side effects, especially for potential drug-drug interactions. Furthermore, the models should provide suggestions to support explanations of the reasons causing drug-drug interactions in some cases. We believe that the models developed in our research can contribute to the drug development process to guide drug safety.

Overview of the Thesis

In the thesis, Chapter 2 gives a survey of data resources used in drug side effect studies with corresponding tasks and methods. First, we describe data resources consisting of two kinds: clinical data and non-clinical data. We also present a commonly used way to represent drug information from these data resources: vectors of drug descriptors for the presence of drug properties in terms of chemical, physical, and biological features. Next, we present three main tasks used in drug side effect studies with corresponding methods in drug side effect benchmark data creation, drug side effect prediction, and drug side effect mechanism analysis. We note here that in this chapter, we only illustrate methods for predicting drug side effects where the input is a single drug and the output is the corresponding side effects, then compare the performances of existing methods.

Chapter 3 addresses the problem of predicting (drug) side effects for drug-drug interactions, where the input is a pair of drugs and the output is corresponding drug side effects. First, we describe related work on the problem, with a focus on the existing state-of-the-art methods of graph representation for drug-drug interaction data. Next, we analyze the problem with this kind of graph representation in terms of the lack of expressing side effect relationships and propose a new representation in the form of a hypergraph to leverage the side effect relationships. Then, we propose a deep learning model namely

CentSmoothie to learn drugs and side effects altogether with the assumption that the side effect should be close to the combination of the properties of the two corresponding drugs, which is also the midpoint. Finally, we show empirical experimental results to verify the performance advantage of CentSmoothie in comparison with existing cutting-edge methods in testing data.

Chapter 4 presents the next model to further improve the prediction performance of CentSmoothie. First, we show the remaining problems of CentSmoothie in two aspects: i) there is only one combination of properties of the drugs for each side effect while in reality, there might exist multiple ways to combine drug properties, and ii) not considering the problem of sparsity given a very few percentages of known drug-drug interactions, which might impair the model performance. Next, we introduce SPARSE, an advanced hypergraph neural network model to solve both problems. Finally, we show empirical experimental results to verify the advantage of SPARSE in comparison with CentSmoothie. In addition, we provide some examples for the ability to explain the reasons for side effects predicted from SPARSE.

Chapter 5 gives concluding remarks on the thesis and discusses some potential further directions to improve the work.

Chapter 2

A survey of drug side effect studies: data, task, and methods

2.1. Introduction

In general, data used in drug side effect studies consist of clinical and non-clinical data. The clinical data contains observations of drug side effects from clinical treatments of patients. These observations have not only adverse drug reactions but also personal contexts, such as dosages of treatments, ages, genders, and diseases of patients. Since different patients can have different adverse drug reactions, such personal contexts can support to build personalized drug side effect prediction models.

The non-clinical data contains information about biological systems such as drug-protein interactions and biological processes. In fact, there are various possible mechanisms in drug side effects, for example, by interactions of drugs with proteins, but the details of these mechanisms are still unknown [Mann and Andrews, 2007, Rieder, 1994]. By integrating clinical data with non-clinical data, it is expected that the quality of drug side effects studies will be improved, and drug side effect mechanisms can be revealed.

Since there are different machine learning methods using various kinds of drug side effect data resources, an overview of current methods in drug side effect studies is necessary. Table 2.1 summarizes the latest survey papers related to drug side effect studies up to 2018. These studies often use either clinical data

[Poloju and Muniganti, 2018] or non-clinical data [Chen et al., 2016]. There is only one survey that uses both kinds of data [Ho et al., 2016], but there is no detailed analysis of methods, such as providing a taxonomy or conducting experiments to compare performances of methods. Recently, there are new studies of drug side effects with the emerging of using machine learning methods, leading to a need for a more detailed classification for these methods. Moreover, drug side effect studies are not only drug side effect prediction [Chen et al., 2016] but also analyzing drug side effect mechanisms by revealing biological components associated with drug side effects [Wang et al., 2013]. Motivated by this, we give a broader view of drug side effect studies containing drug side effect data resources and how computational tasks in drug side effect studies use these kinds of data.

The content of this chapter can be summarized as follows. 1) We summarize the drug side effect data resources containing both clinical and non-clinical data. 2) We summarize a wide range of drug descriptors used in drug side effect studies. 3) We analyze methods used in drug side effect studies in three main tasks: (i) drug side effect benchmark data creation, (ii) drug side effect prediction, and (iii) drug side effect mechanism analysis (We focus on papers in the main journals with the most numbers of papers on this topic such as Bioinformatics, BMC Informatics, Briefing in Bioinformatics, and Nucleic acid research, then we follow cited papers. Papers are collected up to Feb 2019.). In each task, we analyze data and commonly used machine learning methods. 4) We conduct an experiment to compare the drug side effect prediction performances of eight commonly used methods.

2.2. Data resources in drug side effect studies

In this section, we summarize commonly used data resources in drug side effect studies. Fig. 2.1 illustrates a hierarchical classification of data resources in drug side effect studies containing two groups: clinical and non-clinical data.

Table 2.1: Recent surveys on drug side effect studies (up to 2018).

Paper	Task		Data		Method analysis
	Clinical data extraction	Drug side effect prediction	Clinical data	Drug side effect & non-clinical data	
Poloju et al., 2018 [Poloju and Muniganti, 2018]	✓		✓		
Chen et al., 2016 [Chen et al., 2016]		✓		✓	✓
Ho et al., 2016 [Ho et al., 2016]	✓	✓	✓	✓	

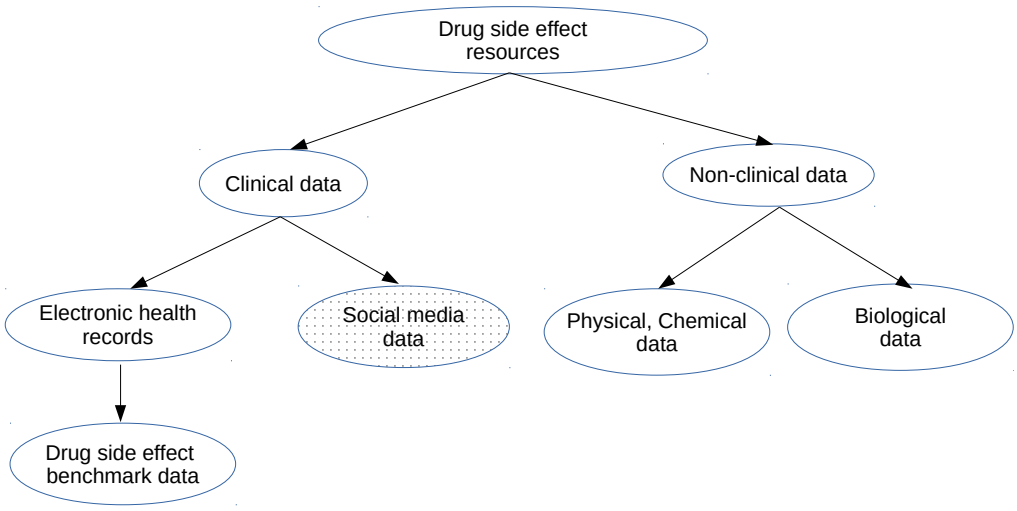


Figure 2.1: Data resource hierarchy in drug side effect studies.

2.2.1 Clinical data

Clinical data contains observations of drug side effects in clinical treatments, which are often electronic health records (EHR) or records from adverse report systems. Each record contains drugs and observed drug side effects. In addition, personal contexts such as demographic and dosage information are also stored. There is evidence that drug side effects are different from differ-

Table 2.2: Commonly used clinical data resources.

Data resources	Personal context	Drug side effect benchmark	
		Monopharmacy	Polypharmacy
FAERS [FDA, 2019]	✓		
OMOP-CDM [Hripcsak et al., 2015]	✓		
SIDER [Kuhn et al., 2015]		✓	
Liu' dataset [Liu et al., 2012]		✓	
AEOLUS [Banda et al., 2016]		✓	
OFFSIDES [Tatonetti et al., 2012]		✓	
TWOSIDES [Tatonetti et al., 2012]			✓

ent patients [Alberti and Cavaletti, 2014], therefore, these personal contexts are important to build personalized drug side effect prediction models [Bao et al., 2017].

Table 2.2 provides the commonly used clinical data resources. For personal contexts, it has FDA Adverse Event Reporting System (FAERS) [FDA, 2019] and Medical Outcomes Partnership Common Data Model (OMOP CDM) [Stang et al., 2010]. There are four main tables in FAERS: demographics, drug, therapy, and reaction. The demographics table describes patient information containing patient identification, age, gender, weight, location, and other related information. The amount and routes of drug administration with patient identifications come from the drug table, and the time of drug treatments is from the drug therapy table. The reaction table contains the drug adverse reactions with patient identifications.

OMOP CDM is a data model provided by Observational Health Data Sciences and Informatics [Hripcsak et al., 2015], which is an international collaboration with the aim to create and apply data analytic solutions to a large number of observational health databases. There are four domains of OMOP CDM v5.0: standardized clinical data, standardized health system data, standardized health economics data, and standardized derived elements. Standardized clinical data contains the core information with clinical events and demographic information of patients. With OMOP CDM, millions of health records from different resources are transformed into pre-defined tables of the four domains, supporting further analysis Simpson et al. [2013].

FAERS was used to extract drug side effect benchmark datasets, which con-

Table 2.3: Commonly used non-clinical databases.

Database	Elements					Having Interactions
	Chemical / Drug	Protein / Gene	Pathway	drug side effect term	Disease	
DrugBank [Wishart et al., 2007]	✓	✓				✓
PubChem [Kim et al., 2015]	✓					
PDB [Berman et al., 2000]		✓				
BindingDB [Liu et al., 2006]	✓	✓				✓
HPRD [Keshava Prasad et al., 2008]		✓				✓
CTD [Davis et al., 2008]	✓	✓			✓	✓
KEGG [Kanehisa and Goto, 2000]	✓	✓	✓		✓	✓
SuperTarget [Günther et al., 2007]	✓	✓				✓
ADReCS [Cai et al., 2014]				✓		
DART [Ji et al., 2003]	✓	✓		✓		✓
TTD [Chen et al., 2002]	✓	✓	✓		✓	✓
Bio2RDF [Belleau et al., 2008]	✓	✓	✓	✓	✓	✓

tain reliable drug side effect associations [Kuhn et al., 2015, Tatonetti et al., 2012]. SIDER, a common drug side effect benchmark dataset for many drug side effect studies, was extracted from FAERS for drug side effects caused by single drugs (monopharmacy) [Kuhn et al., 2015]. Liu' dataset [Liu et al., 2012] is a benchmark dataset extracted from SIDER into the binary format with additional drug information. AEOLUS is also a monopharmacy dataset extracted from FAERS and has more drug side effect associations than SIDER. Extracting from FAERS with a criterion of removing bias data, OFFSIDES for drug side effects caused by single drugs, and TWOSIDES for drug side effects caused by combinations of two drugs (polypharmacy) were created [Tatonetti et al., 2012]. However, SIDER, AEOLUS, OFFSIDES, and TWOSIDES only contain two kinds of information: drugs and drug side effects. As far as we know, there is no benchmark Drug side effect data for academic research that con-

tains personal contexts such as diseases, and duration of drug treatments.

In recent years, data from social media such as Facebook, Twitter is another kind of data to analyze drug side effects. This social media data contains comments from patients during drug treatments. However, the tasks on this kind of data are mainly drug side effect identification using techniques of natural language processing [Emadzadeh et al., 2017, Huynh et al., 2016, Lee et al., 2017], which are considered as data pre-processing steps, therefore, we do not cover them in this survey.

2.2.2 Non-clinical data

The non-clinical data contains information about chemical, physical, and biological properties of drugs and biological systems, which can help revealing mechanisms of drugs and drug side effects. In fact, drug side effects are the results of complex reactions of drugs with biological components. Some studies have shown that drug side effects can be the results of reactions of drug chemicals with proteins [Mann and Andrews, 2007, Rieder, 1994, Ring and Brockow, 2002], which interrupts normal biological processes leading to abnormal reactions in human bodies. By using this kind of data, we can improve the performance of models and extract possibly associated biological components with drug side effects.

Table 2.3 summarize the commonly used non-clinical databases in drug side effect studies in two aspects: elements in each database and interactions among elements existing or not. For example, ADReCS [Cai et al., 2014] is a database for only drug side effect term definitions, and KEGG [Kanehisa and Goto, 2000] contains information about proteins, drugs, biological pathways, diseases, and interactions among them such as drugs with proteins targets. To link these databases, Bio2RDF [Belleau et al., 2008] provides interconnections among elements of different databases.

Finally, the connection between clinical and non-clinical data can be illustrated by a network in Fig. 2.2. The clinical data provides information of drug side effect connections with personal contexts. The non-clinical data contains connections of drug-drug, drug-protein, protein-protein, and protein-biological pathway. This network is used to support some computational tasks repre-

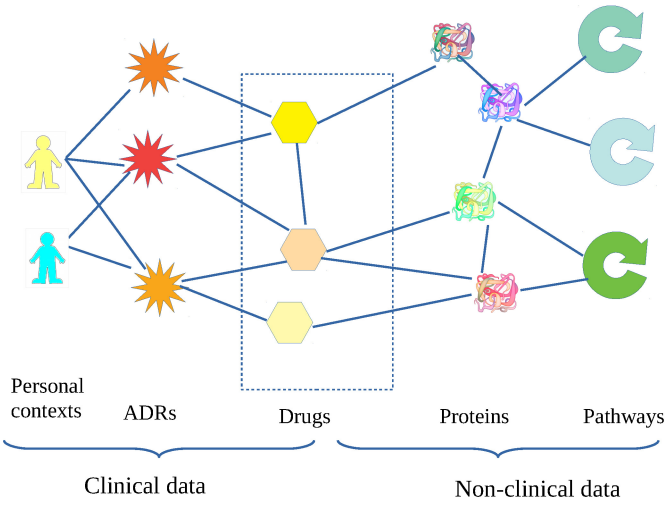


Figure 2.2: A network for clinical and non-clinical data.

sented in Section 4.

2.3. Drug descriptors

One possible way of encoding drugs is to use descriptors, which are physical, chemical, and biological characteristics of a drug. Since the quality of these descriptors impacts drug side effect prediction performances, the understanding of drug descriptors is a basic need. Fig. 2.3 presents a classification for drug descriptors. In general, drug descriptors can be categorized into two classes: physical or chemical descriptors (PC-descriptors) and biological descriptors (BIO-descriptors).

2.3.1 Physical or chemical (PC) descriptors

The PC-descriptors describe the structure of drug molecules and their physical, and chemical properties [Grisoni et al., 2018, Testa and Kier, 1991, Todeschini and Consonni, 2008]. Based on their dimensionalities and properties, this class of descriptors can be divided into 3 subgroups: structural descriptors, spatial descriptors, and other miscellaneous descriptors.

The structural descriptors describe features of molecular structures such as

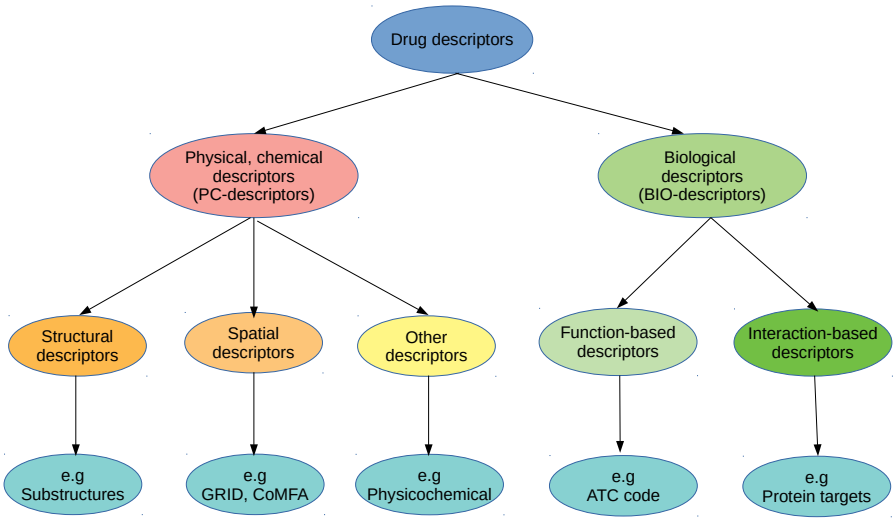


Figure 2.3: Different kinds of drug descriptors.

Table 2.4: Two groups of structural descriptors implemented in CDK [Steinbeck et al., 2003].

Group	Name	Number of descriptors
Variable-size	Daylight family [Daylight, 2018]	-
Fixed-size	E-State fragments [Hall and Kier, 1995]	79
	Klekota-Roth [Klekota and Roth, 2008]	4860
	MACCS keys [Durant et al., 2002]	166
	PubChem descriptors [Kim et al., 2015]	881
	CDK substructures [Steinbeck et al., 2003]	307

atom counters, atom pairs, rings, and other substructures. Table 2.4 presents two groups of structural descriptors (fingerprints) implemented in Chemistry Development Kit (CDK) [Steinbeck et al., 2003]: variable-size and fixed-size groups. The former group generates substructures from a given set of molecules, in which the number of substructures can be changed depending on the provided molecule set [Daylight, 2018]. In contrast, the latter group uses predefined substructures, for example, MACCS keys and PubChem descriptors. An illustration of PubChem descriptors is shown in Fig. 2.4. The PubChem

descriptors contain pre-defined 881 bits, which are divided into seven sections with corresponding bits. For instance, bit 308, which belongs to section 3 of simple atom pairs, indicates the existence of O-H connection.

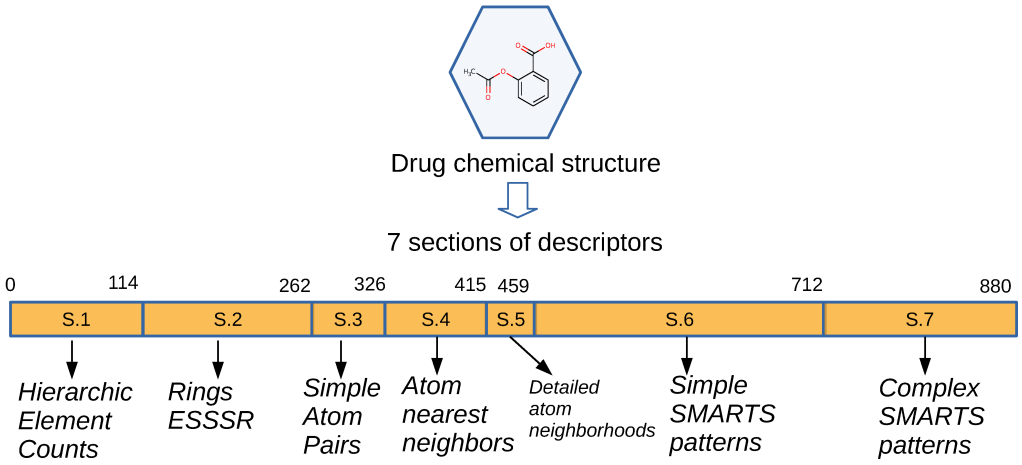


Figure 2.4: Seven sections in PubChem descriptors.

The spatial descriptors describe spatial properties of drug molecules. In PubChem 3D database [Bolton et al., 2011], 3D conformers descriptors of molecules are used. These descriptors are calculated by OMEGA [Openeye scientific software, 2018], a tool published by OpenEye. Molecular interaction fields (MIFs) are another kind of spatial descriptors for drugs. MIFs describe spatial variation of the interaction energy between a molecular target and a chosen probe. Probes are small molecules representing common interactions such as hydrophobic, hydrogen bond donors, and acceptors [Wermuth, 2011]. Some well known MIFs are GRID [Goodford, 1985], VolSurf [Crivori et al., 2000], CoMFA [Kubinyi, 1998], and MetaSite [Cruciani et al., 2005]. Fig. 2.5 illustrates the idea of GRID descriptors. A molecule is put into a cube with grids. An empirical energy function will be used to calculate the interaction field of each cell at position (x, y, z) of the cube. The energy function is defined by:

$$E_{xyz} = \sum E_{lj} + \sum E_{el} + \sum E_{hb}$$

where E_{lj}, E_{el}, and E_{eb} are the Lennard-Jones function, the electronic function, and the hydrogen bound function, respectively [Goodford, 1985].

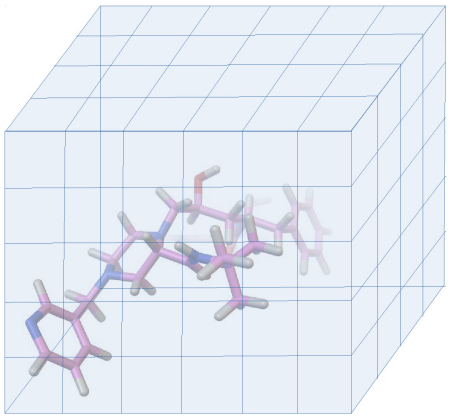


Figure 2.5: A molecule with 3D GRID.

Other miscellaneous descriptors such as physicochemical properties of drugs also affect the action of drugs. Lipophilicity [Testa et al., 2000, Young et al., 1988] impacts solubility, absorption, distribution, membrane penetration, and plasma protein binding of drugs. Hydrogen bond [van de Waterbeemd and Kansy, 1992] is another physical property of electrostatic attraction, which takes two out of five Lipinski's rules [Lipinski et al., 1997]. Size/geometric features of drugs such as molecular weight and atom counters can also reflect drug properties.

2.3.2 Biological (BIO) descriptors

The BIO-descriptors describe biological properties of drugs, which can be classified into two subgroups: function-based descriptors and interaction-profile descriptors. The function-based descriptors describe purposes of drugs in therapy. ATC code [WHO, 2019], which is a classification system for drugs based on therapeutic properties, is a typical example of function-based descriptors.

The interaction-profile descriptors describe associated biological components of drugs containing protein targets and associated biological pathways of drugs [Liu et al., 2012, Yamanishi et al., 2012]. These interaction-profile descriptors are taken from the databases having drug interaction information in Table 2.3, such as DrugBank [Wishart et al., 2007], BindingDB [Liu et al., 2006], and Bio2RDF [Belleau et al., 2008].

2.4. Tasks, data, and methods

In this section, we summarize three main computational tasks in drug side effect studies: (i) drug side effect benchmark data creation, (ii) drug side effect prediction, and (iii) drug side effect mechanism analysis. Fig. 2.6 provides an overview of drug side effect studies of these three tasks. In each task, we analyze objectives, data, and commonly used methods. The main notations for this chapter used in the following subsections are described in Table 2.5.

Table 2.5: Main notations in Chapter 2.

Notation	Description
$i \in \{1, \ldots, d\}$	a drug index in a set of given d drugs
$j \in \{1, \ldots, s\}$	a drug side effect index in a set of given s drug side effects
$\mathbf{x}_i \in \mathbb{R}^e$	a descriptor vector of size e of drug i
$\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_d]^\mathsf{T} \in \mathbb{R}^{d \times e}$	a descriptor matrix of given d drugs, T is the transpose operator.
$y_{i,j} \in \mathbb{R}$	an association score of drug i and drug side effect j
$\mathbf{y}_i = [y_{i,1} \ldots y_{i,s}]^\mathsf{T} \in \mathbb{R}^s$	a vector for association scores of drug i with s drug side effect
$\mathbf{Y} = [\mathbf{y}_1 \ldots \mathbf{y}_d]^\mathsf{T} \in \mathbb{R}^{d \times s}$	a given drug side effect association score matrix
$\mathbf{h}_i \in \mathbb{R}^m$	a vector of size m representing associated biological components of drug i
$\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_d]^\mathsf{T} \in \mathbb{R}^{d \times m}$	a given drug-biological component matrix

2.4.1 Task 1: Drug side effect benchmark data creation

Clinical data for drug side effects contains millions of records with redundant information, for example, some records contain similar information. Creating a drug side effect benchmark dataset is a necessary task in drug side effect

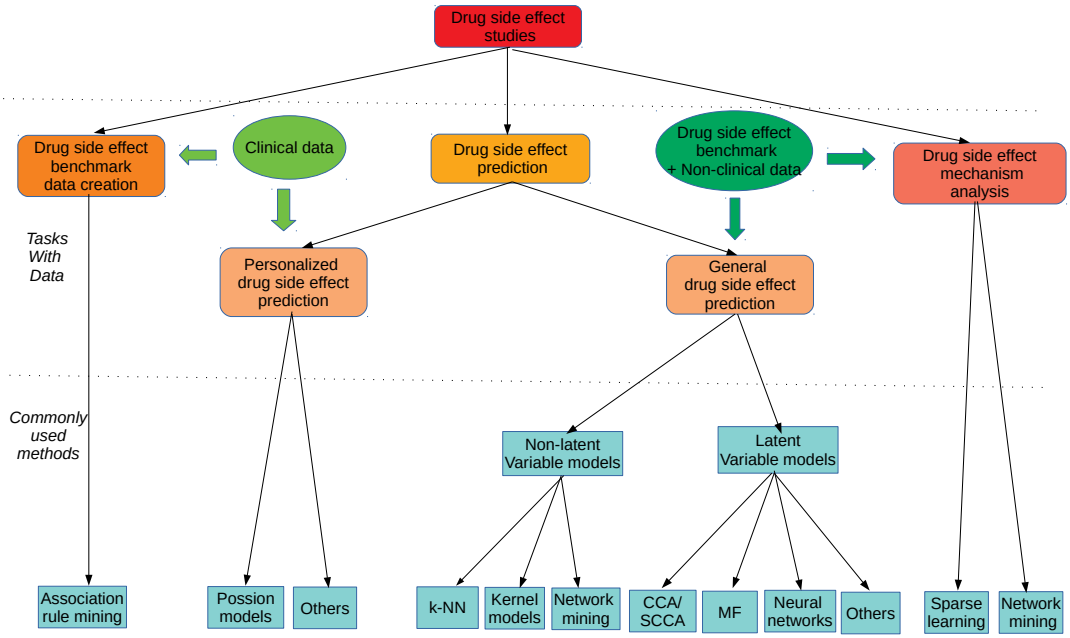


Figure 2.6: Computational tasks of drug side effect studies: Data and commonly used methods.

studies. It helps other studies in evaluating performances of new methods and comparing them with existing methods.

Table 2.6: Contingency table for Fisher's exact test.

		Number of records of drugs	
		drug i	other drugs
drug side effect j	Yes	n_1	n_3
	No	n_2	n_4

In drug side effect studies, benchmark data is extracted from clinical records to retrieve reliable drug side effect associations, which are pairs of drugs with corresponding drug side effects. However, drug side effect pairs have different levels of association significance in clinical records. Some pairs of drug side effects rarely appear in the clinical records, leading to their low association significance. In addition, some records often contain a combination of more than one drug, making the verification of drug side effect associations difficult.

To check the significance of drug side effect associations, association rule mining or statistical significance tests can be applied [Montastruc et al., 2011]. We will briefly explain a typical significance test, Fisher's exact test [Agresti, 1992]. Consider drug i and drug side effect j in a clinical database, the association information of drug i and drug side effect j is stored in a contingency table as in Table 2.6. In this table, n_1 denotes the number of records containing drug side effect j of drug i, while n_2 is that of the other drugs. The number of records that do not contain drug side effect j of drug i is n_3, and that of the other drugs is n_4. The Fisher's exact test evaluates the significance of the association of drug i and drug side effect j by a p-value:

$$p = \frac{(n_1 + n_2)!(n_3 + n_4)!(n_1 + n_3)!(n_2 + n_4)!}{n_1!n_2!n_3!n_4!(n_1 + n_2 + n_3 + n_4)!}$$

This technique was used on FAERS to extract SIDER, a monopharmacy drug side effect benchmark dataset used in a large number of drug side effect studies [Kuhn et al., 2010, 2015]. The technique was also used to extract OFFSIDES for monopharmacy drug side effects, which are drug side effects of drugs that do not appear in the drug's package insert and TWOSIDES for polypharmacy drug side effects of drug-drug interactions [Tatonetti et al., 2012].

2.4.2 Task 2: Drug side effect prediction

Predicting drug side effects of drugs, or drug side effect association scores is an important objective of drug side effect studies. Depending on the personal context information is used or not, studies in drug side effect prediction can be divided into two classes: personalized drug side effect prediction and general drug side effect prediction. In the following subsections, we analyze machine learning methods according to each class. We here note that the prediction task in this chapter is illustrated with single drug input. The prediction for drug-drug interactions will be addressed in Chapters 3 and 4.

Personalized drug side effect prediction

The personalized drug side effect prediction uses personal contexts taken from clinical data with information such as dosages of treatments, gender, and

age of each patient. Therefore, the prediction result will be different among patients even with the same drugs. For this prediction, we focus on methods using Poisson models, which are commonly used models for personalized drug side effect prediction.

i. Poisson models

The aim of using Poisson models is to predict the probabilities of the numbers of occurrences of drug side effects during drug treatments. It is assumed that these numbers follow Poisson distributions with expectations depending on the taken drugs [Bao et al., 2017, Simpson et al., 2013]. For simplicity, considering a patient p in drug treatment, the probabilities of numbers of occurrences of s drug side effects $\mathbf{\Phi}(\tilde{\mathbf{y}}|\tilde{\mathbf{x}}) \in \mathbb{R}^s$ are calculated by:

$$\mathbf{\Phi}(\tilde{\mathbf{y}}|\tilde{\mathbf{x}}) = [P(\tilde{y}_1|\phi_1(\tilde{\mathbf{x}})) \dots P(\tilde{y}_j|\phi_j(\tilde{\mathbf{x}})) \dots P(\tilde{y}_s|\phi_s(\tilde{\mathbf{x}}))]^\mathsf{T}$$

where $\tilde{\mathbf{x}} = [\tilde{x}_{p,1} \dots \tilde{x}_{p,i} \dots \tilde{x}_{p,d}]^\mathsf{T}$ is a vector indicating drugs taken by patient p during the treatment, $\tilde{\mathbf{y}} = [\tilde{y}_1 \dots \tilde{y}_j \dots \tilde{y}_s]^\mathsf{T}$ is a vector denoting the numbers of occurrences of s drug side effects, and $P(\tilde{y}_j|\phi_j(\tilde{\mathbf{x}}))) = \phi_j(\tilde{\mathbf{x}})^{\tilde{y}_j} e^{-\phi_j(\tilde{\mathbf{x}})} / \tilde{y}_j!$ is the Poisson distribution for the number of occurrences of drug side effect j with expectation $\phi_j(\tilde{\mathbf{x}})$. A commonly used formulation of ϕ_j is:

$$\phi_j(\tilde{\mathbf{x}}) = \exp\left(\theta_{p,j} + \sum_{i=1}^{d} \tilde{x}_{p,i}.w_{i,j}\right)$$

where $\theta_{p,j}$ is a parameter depending on the patient, leading to differences in drug side effect occurrences of different patients, and $w_{i,j}$ is a parameter used as a weight for the association of drug i and drug side effect j [Simpson et al., 2013]. This formulation shows a multiplicative contribution of each drug to the expectation of the number of occurrences of each drug side effect.

However, the existing Poisson models have a limitation in terms of integrating other information such as weights, genders of patients and also non-clinical data.

ii. Other methods

There are other methods that were used to combine drugs with personal contexts into medical case vectors. A feature-based similarity method was proposed to learn weights for these medical case vectors with the idea to distinguish cases having a drug side effect from cases not having the drug side effect [Yang et al., 2014]. These medical case vectors were also used as inputs for a classification problem [Chen, 2018].

General drug side effect prediction

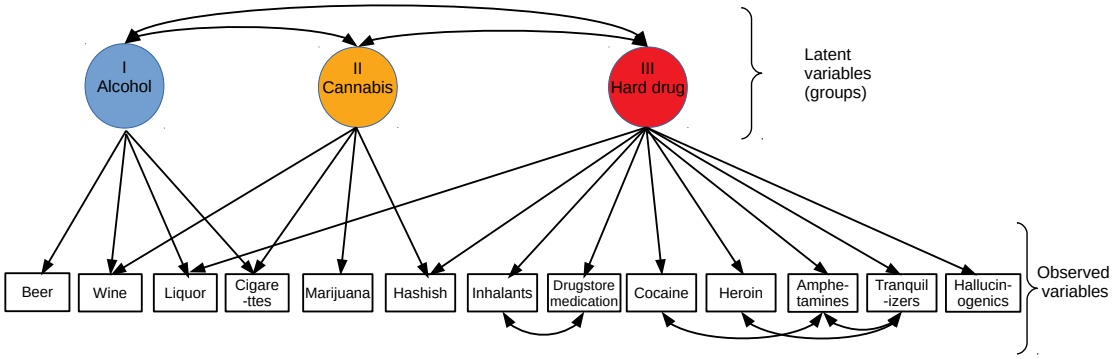


Figure 2.7: An example of latent variables with thirteen psychoactive substances [Huba et al., 1981].

In contrast to personalized drug side effect prediction, general drug side effect prediction predicts drug side effect association scores without using personal contexts. A common approach for this class is to combine knowledge of drugs from non-clinical data to enrich drug information and apply machine learning methods to build drug side effect prediction models. As presented in Section 2.3, drug information is described by various types of drug descriptors. The drug side effect prediction models receive the drug descriptors as the inputs and output all corresponding drug side effects.

In this study, we consider general drug side effect prediction as a multi-label classification problem such that each drug side effect is a label and each drug can have many labels [Muñoz et al., 2017, Zhang et al., 2015]. The prediction models calculate the association scores, which are real numbers, of each drug

with all labels. The final labels of the drug are selected from these scores by a ranking method. In detail, a drug side effect prediction model is formulated as a function $\mathbf{f} : \mathbb{R}^e \rightarrow \mathbb{R}^s$, where e is the number of descriptors and s is the number of drug side effects. Given a drug with descriptor vector $\mathbf{x} \in \mathbf{R}^e$, the model predicts drug side effect association scores with s drug side effects: $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^s$.

We further classify the models into two classes: *i. non-latent variable models* and *ii. latent variable models*. Latent variables are ones that are not directly observed or measured and needed to infer from observed data. Fig. 2.7 presents an example of latent variables from a study on finding patterns of psychoactive substances used in adolescents [Huba et al., 1981]. There are thirteen psychoactive substances from beer to hallucinogenics, which are observed variables. In addition, there are some correlated pairs of substance usage, for example, cocaine and amphetamines. The study suggested that these substances can be grouped into three groups: alcohol, cannabis, and hard drug. The patterns of substance usage will be taken from these three groups, which are called latent variables.

A latent variable model is a model that contains latent variables obtained from observed ones. In application to drug side effect prediction, latent variables of drugs can be interpreted as groups of drug descriptors that are highly correlated with each other. The representations of drugs in the space created by latent variables are called latent vectors.

In the following contents, we first describe the formulation for function $\mathbf{f}$ according to models of the two classes: *non-latent variable models* and *latent variable models*, which are based on the criteria that latent vectors of drugs are learned or not. Then we present an experiment to compare the prediction performances of these models.

i. Non-latent variable models

In non-latent variable models, drug descriptors are used to predict drug side effect associations without learning drug latent vectors. We present three typical methods: *a. k nearest neighbors*, *b. kernel methods* and *c. mining networks of drug side effects*.

a. k nearest neighbors

The idea of using k nearest neighbors (k-NN) is that drugs having similar descriptor vectors tend to have similar drug side effects [Cao et al., 2015, Liu et al., 2012, Muñoz et al., 2016, Pauwels et al., 2011, Zhang et al., 2015]. Suppose that there is a similarity measure $sim : \mathbb{R}^e \times \mathbb{R}^e \rightarrow \mathbb{R}$, for example, cosine similarity. To predict drug side effect association scores $\mathbf{f}(\mathbf{x})$, first the top k most similar drugs to $\mathbf{x}$ are identified resulting in a set of indices of the similar drugs $T(\mathbf{x}, k)$. Then the drug side effect association scores are calculated by:

$$\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \dots f_j(\mathbf{x}) \dots f_s(\mathbf{x})]^\mathsf{T}.$$

where f_j is a weighted average function:

$$f_j(\mathbf{x}) = \sum_{i \in T(\mathbf{x},k)} w_i(\mathbf{x}) y_{i,j}, \quad j \in \{1 \dots s\},$$

with weights w_i are obtained from drug similarities, for example:

$$w_i(\mathbf{x}) = \frac{sim(\mathbf{x}, \mathbf{x}_i)}{\sum_{i' \in T(\mathbf{x},k)} sim(\mathbf{x}, \mathbf{x}_{i'})} \tag{2.1}$$

Some extensions of KNN were also applied, for example the linear neighborhood similarity method (LNSM) [Zhang et al., 2016]. In LNSM, the similarity weights are calculated such that a drug descriptor vector is a linear combination of descriptor vectors of the neighbor drugs with corresponding similarity weights.

b. Kernel methods

The idea of using kernel methods, for example, support vector machines (SVMs), is to use classification functions calculated from kernel functions in the form of inner products of drug descriptor vectors [Jahid and Ruan, 2013, Liu et al., 2012, Pauwels et al., 2011, Yamanishi et al., 2012]. To predict drug side effect association scores $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \dots f_j(\mathbf{x}) \dots f_s(\mathbf{x})]^\mathsf{T}$, the kernel methods use the following form for f_j:

$$f_j(\mathbf{x}) = g(\sum_{i=1}^{d} w_{i,j} y_{i,j} K(\mathbf{x}, \mathbf{x}_i)), \quad j \in \{1 \dots s\}$$

where g is a function, for example, a sign function. $K : \mathbb{R}^e \times \mathbb{R}^e \rightarrow \mathbb{R}$ is a kernel function, for example, a radial basis function (rbf): $K(\mathbf{x}, \mathbf{x}_i) = \exp(-\frac{(\mathbf{x}-\mathbf{x}_i)^\mathsf{T}(\mathbf{x}-\mathbf{x}_i)}{2\delta^2})$

with a hyperparameter δ, and $w_{i,j}$ is a parameter used as a weight for the association of drug i and drug side effect j.

Different from k-NN, the kernel methods learn weights from a training process, which depends on both drugs and drug side effects, while weights in k-NN are calculated only from drug similarities.

c. Mining networks of drug side effects

Consider a drug side effect network $G = (V, E)$, where V is a set of nodes of d drug and s drug side effects: $V = \{v_1, \ldots, v_i, \ldots, v_d\} \cup \{\nu_1, \ldots, \nu_j, \ldots, \nu_s\}$, and E is a set of edges of drug nodes-drug side effect nodes for known drug side effects of drugs and drug nodes-drug nodes for drug similarities. The idea of mining this network is that if a drug and a drug side effect in the network are well-connected, they possibly have a high association score Cami et al. [2011], Davazdahemami and Delen [2018], Lin et al. [2013]. This approach can be formulated in two steps:

1. Calculate partial connection scores $r(v_i, \nu_j) \in \mathbb{R}^l$ of each pair of drug node v_i and drug side effect node ν_j using l different measures on G. A commonly used measure is the Jaccard index [Cami et al., 2011, Davazdahemami and Delen, 2018]. Let $N_i = \{v | (v, v_i) \in E\}$ be a set of neighbor nodes of drug node v_i, and $N_j = \{v | (v, \nu_j) \in E\}$ be that of drug side effect node ν_j, the partial connection score calculated by Jaccard index is: $|N_i \cap N_j| / |N_i \cup N_j|$, where $|\,.\,|$ denotes the cardinality of a set. Some other measures such as Dice index and Adamic/Adar index were also applied [Davazdahemami and Delen, 2018]. Random walk [Tong et al., 2006] was also applied to calculate r [Rahmani et al., 2016].

2. Calculate drug side effect association scores $\mathbf{f}(\mathbf{x})$ of a drug with descriptor vector $\mathbf{x}$. Let $v(\mathbf{x})$ be the corresponding node in G of the drug. The association scores are obtained by:

$$\mathbf{f}(\mathbf{x}) = [f(r(v(\mathbf{x}), \nu_1)) \ldots f(r(v(\mathbf{x}), \nu_j)) \ldots f(r(v(\mathbf{x}), \nu_s))]^\mathsf{T}$$

where f was often a binary function [Lin et al., 2013] or a logistic regression function (LR) [Cami et al., 2011]. In addition, random forest (RF) was also applied to f [Davazdahemami and Delen, 2018].

However, a problem with mining drug side effect networks is sparsity that there are too few edges between drugs and drug side effects, for example, in SIDER dataset, the edge density is 0.017. This makes the prediction less effective since there is only a small number of drug side effects predicted for each drug.

ii. Latent variable models

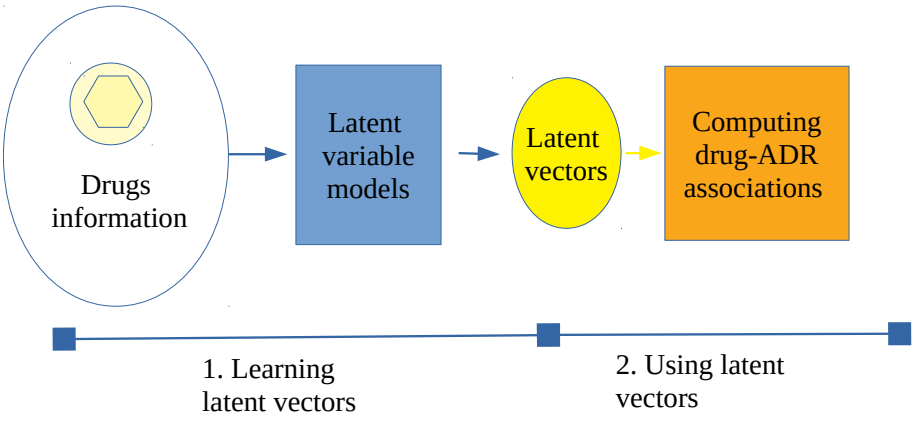


Figure 2.8: Learning latent variables and using latent variables.

In latent variable models, drug side effect association scores are calculated by using drug latent vectors learned from drug descriptors. Fig. 2.8 illustrates two stages of using latent models: learning latent vectors of drugs and then using these latent vectors for prediction. It is expected that latent vectors can remove redundant information from drug descriptors, for example, unnecessary descriptors. In addition, calculating with latent vectors of small size can reduce the complexity of high-dimensional data. In this chapter, we briefly describe three commonly used latent variable models (canonical correlation analysis, matrix factorization, and neural networks), and some other miscellaneous models.

a. Canonical correlation analysis

The aim of using canonical correlation analysis (CCA) is to find weight vectors $\mathbf{a} \in \mathbb{R}^e$ and $\mathbf{b} \in \mathbb{R}^s$ such that the correlation of the projections of drug descriptor matrix $\mathbf{X}$ and drug side effect association matrix $\mathbf{Y}$ is maximized

[Yamanishi et al., 2012]:

$$\arg\max_{\mathbf{a},\mathbf{b}} \frac{(\mathbf{Xa})^\top(\mathbf{Yb})}{\sqrt{(\mathbf{Xa})^\top(\mathbf{Xa})}\sqrt{(\mathbf{Yb})^\top(\mathbf{Yb})}}.$$

The first pair of $(\mathbf{Xa}, \mathbf{Yb})$ is called the first pair of canonical variables (latent variables). The remaining pairs of canonical variables have an additional constraint in that they are uncorrelated with existing pairs of canonical variables. c pairs of weight vectors $\mathbf{a}$ and $\mathbf{b}$ form two weight matrices: $\mathbf{A} \in \mathbb{R}^{e\times c}$ and $\mathbf{B} \in \mathbb{R}^{s\times c}$, respectively.

The latent vector of a drug with descriptor vector $\mathbf{x}$ is calculated by: $\mathbf{z}(\mathbf{x}) = \mathbf{A}^\top\mathbf{x}$. drug side effect association scores $\mathbf{f}(\mathbf{x})$ are obtained by minimizing the distance of latent vectors:

$$\mathbf{f}(\mathbf{x}) = \arg\min_{\mathbf{y}\in\mathbb{R}^s} \left\| \mathbf{z}(\mathbf{x}) - \mathbf{B}^\top\mathbf{y} \right\|.$$

where $\| \, . \, \|$ is a norm, for example, Euclidean norm.

Sparse canonical correlation analysis (SCCA), a variant of CCA, was also applied to predict drug side effect association scores Pauwels et al. [2011]. In SCCA, L_1 regularization is applied to columns of $\mathbf{A}$ and $\mathbf{B}$, leading to their sparsity.

b. Matrix factorization

The idea of using matrix factorization (MF) is illustrated in Fig.2.9 [Poleksic and Xie, 2018]. It is assumed that drugs and drug side effects share c unknown latent variables. Then the drug side effect association matrix $\mathbf{Y}$ is decomposed into two matrices of latent vectors of drugs and drug side effects in the space of latent variables: $\mathbf{U} \in \mathbb{R}^{d\times c}$ and $\mathbf{V} \in \mathbb{R}^{s\times c}$, such that $\mathbf{Y} \approx \mathbf{U}\mathbf{V}^\top$. Supposing there is a drug similarity matrix $\mathbf{S}_d \in \mathbb{R}^{d\times d}$ calculated from drug descriptors matrix $\mathbf{X}$, and a drug side effect similarity matrix $\mathbf{S}_s \in \mathbb{R}^{s\times s}$ calculated from drug side effect definitions, the objective function is:

$$\arg\min_{\mathbf{U},\mathbf{V}} \left\| \mathbf{Y} - \mathbf{U}\mathbf{V}^\top \right\| + \Re(\mathbf{U}, \mathbf{V}, \mathbf{S}_d, \mathbf{S}_s),$$

where the first part is the error from matrix factorization, and the second one is the regularization for $\mathbf{U}$ and $\mathbf{V}$ given $\mathbf{S}_d$ and $\mathbf{S}_s$, for example, Laplacian regularization.

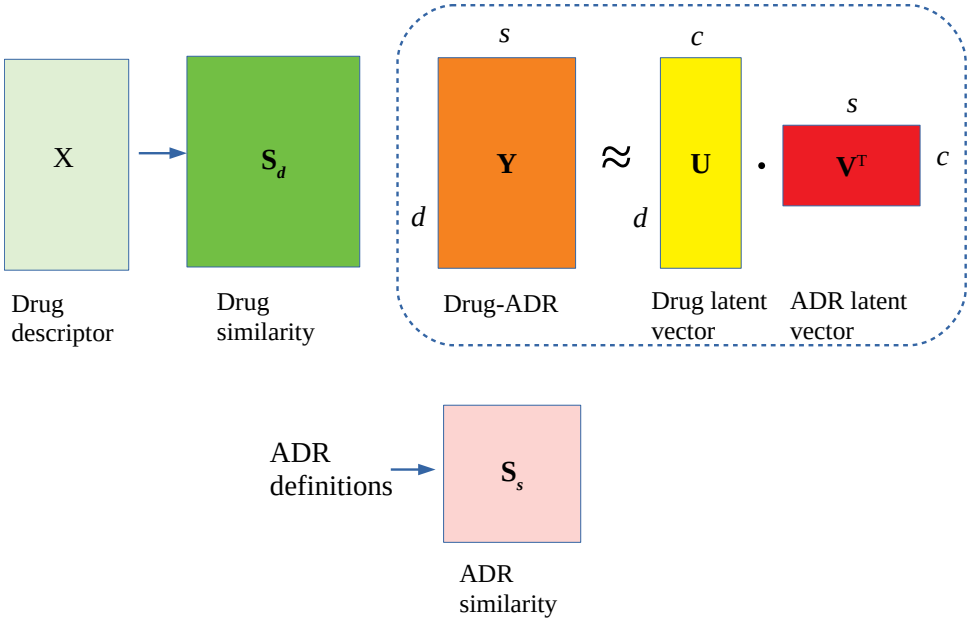


Figure 2.9: An illustration of matrix factorization.

To calculate drug side effect association scores $\mathbf{f}(\mathbf{x})$, first k-NN is applied to calculate a new latent vector $\mathbf{z}(\mathbf{x})$ from the existing drug latent vectors:

$$\mathbf{z}(\mathbf{x}) = \sum_{i \in T(\mathbf{x},k)} w_i(\mathbf{x})\mathbf{u}_i$$

where $\mathbf{u}_i \in \mathbb{R}^c$ is the latent vector of drug i such that $\mathbf{u}_i^\mathsf{T}$ corresponds to the i^{th} row of $\mathbf{U}$, $T(\mathbf{x},k)$ is the set of indices of the top k most similar drugs to $\mathbf{x}$, and $w_i(\mathbf{x})$ are similarity weights defined in Equation 2.1.

Then, the drug side effect association scores are obtained by:

$$\mathbf{f}(\mathbf{x}) = \mathbf{V}\mathbf{z}(\mathbf{x})$$

Different from CCA, MF only focuses on $\mathbf{Y}$ to learn latent vectors and uses $\mathbf{X}$ as additional information which can be omitted from the regularization part. Meanwhile, CCA requires both $\mathbf{X}$ and $\mathbf{Y}$ to obtain latent vectors.

c. Neural networks

Neural networks, which are machine learning models featured by the ability to learn non-linear relationships, were applied to predict drug side effect

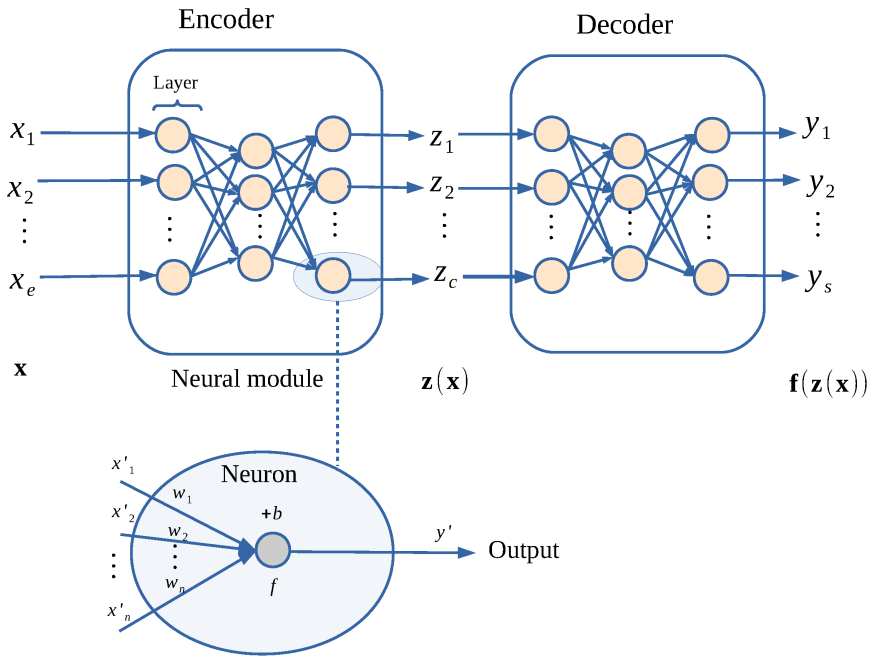


Figure 2.10: An illustration of a neural network.

association [Dey et al., 2018, Wang et al., 2019, Zitnik et al., 2018]. Fig. 2.10 illustrates this technique in detail. The basic components of neural networks are *neurons*. Each neuron receives an input vector $\mathbf{x}' = [x'_1\ x'_2 \dots x'_n]^\mathsf{T}$ and outputs a value y' by a function: $y' = f(\mathbf{w}^\mathsf{T}\mathbf{x}' + b)$, where b is a bias, $\mathbf{w} = [w_1\ w_2 \dots w_n]^\mathsf{T}$ is a weight vector, and f is an activation function, for example, a sigmoid function, making non-linear combinations. A neural network module is composed of multiple layers of neurons that the output of each neuron of a layer is used as an input for neurons of other layers. The outputs of a neural module, for example, named *Encoder*, given an input vector $\mathbf{x}$ is denoted by $Encoder(\mathbf{x})$.

To predict drug side effect association scores $\mathbf{f}(\mathbf{x})$, there are two steps to process:

1. Obtain the latent vector: $\mathbf{z}(\mathbf{x}) = Encoder(\mathbf{x})$, where *Encoder* is a neural module receiving drug descriptor vector $\mathbf{x}$ as the input vector.

2. Predict drug side effect association scores: $\mathbf{f}(\mathbf{x}) = Decoder(\mathbf{z}(\mathbf{x}))$, where *Decoder* is a neural module receiving drug latent vector $\mathbf{z}(\mathbf{x})$ as the input vector.

An advantage of using neural networks is the ability to approximate any continuous function. If there is no hidden layer, neural networks become logistic regression functions. The architecture of neural networks can be more complex when changing connections of neurons and the numbers of layers, for example, a multi-layer feedforward neural network (MLN) [Wang et al., 2019], or a deep convolutional neural network (DCN) [Dey et al., 2018]. These complex neural networks aim to approximate mapping functions from inputs to outputs better. However, the number of parameters in a neural network is often much larger than that of other models. This problem leads to increasing computational complexity and the potential for overfitting of neural networks.

d. Other methods

There are some miscellaneous methods to obtain latent vectors of drugs to predict drug side effect associations, for example, mapping drugs into a drug side effect space [Dimitri and Lió, 2017, Xiao et al., 2017] and mapping drugs into a metabolic reaction space [Shaked et al., 2016]. In mapping drugs into a drug side effect space, groups of highly correlated drug side effects were extracted, then each drug was represented by a vector over these groups. In mapping drugs into a metabolic reaction space, flux variability analysis (FVA) was applied to represent drug-protein/gene interaction profiles by a vector over metabolic reactions [Mahadevan and Schilling, 2003], then these vectors were used to predict drug side effects.

Performance comparison in general drug side effect prediction

We conducted experiments to compare the general drug side effect prediction performances on monopharmacy cases of eight machine learning models. There were four non-latent variable models: LNSM [Muñoz et al., 2017, Zhang et al., 2016], SVMs [Jahid and Ruan, 2013, Liu et al., 2012], RF [Davazdahemami and Delen, 2018, Liu et al., 2012] and LR[Cami et al., 2011, Liu et al., 2012], and four latent variable models: CCA [Pauwels et al., 2011, Yamanishi et al., 2012], MF [Poleksic and Xie, 2018], MLN [Wang et al., 2019], DCN [Dey et al., 2018] (The convolutional network proposed in [Zitnik et al., 2018] addressed polypharmacy drug side effects, so we do not compare.).

i.Experimental setups

We ran experiments with AEOLUS dataset [Banda et al., 2016], a monopharmacy dataset for drug side effect prediction, which was also used in [Muñoz et al., 2017] (AEOLUS is the largest one among AEOLUS, SIDER, and Liu's datasets.). We only selected drugs appearing in DrugBank and drug side effects occurring in more than 50 drugs. The final statistical information of the dataset is provided in Table 2.7, containing the number of drugs, the numbers of drug side effects, the numbers of drug side effect associations, the average, minimum, and maximum numbers of drug side effects per each drug.

Table 2.7: Statistics of the used dataset.

#drugs	#ADRs	#drug-ADR pairs	#ADRs/drug		
			Avg.	Min	Max
1,385	2,707	605,121	445	1	2,703

Table 2.8: Statistics of the used drug descriptors.

Name	Source	Size
PCBio	Pubchem+Bio2RDF	7, 593
2DChem	PubChem	[N_ATOMS_OF_DRUG , 53]

In the experiments, we used PCBio and Chem2D as two kinds of drug descriptors with information presented in Table 2.8. PCBio descriptors are the combinations of PubChem descriptors taken from PubChem and chemical, physical, and biological descriptors taken from Bio2RDF. We extracted descriptors with information from DrugBank in Bio2RDF as in [Muñoz et al., 2016], and selected descriptors occurring in at least 3 drugs. 2DChem descriptors are drug chemical descriptors represented in the form of a matrix such that each row of the matrix corresponds to chemical features of an atom in a drug. To represent 2DChem descriptors, we extracted 53 chemical properties of each atom in the drug's molecule, hence each drug is represented in the form of a matrix that the number of rows equals to the number of atoms of the drug and the number of columns is 53 (see supplement materials). In our experiments,

2DChem descriptors are only used for DCN model [Dey et al., 2018], other models use PCBio descriptors.

Two commonly used metrics were selected to evaluate prediction performance: area under the ROC curve (AUC) and area under the precision-recall curve (AUPR) [Dimitri and Lió, 2017, Pauwels et al., 2011, Yamanishi et al., 2012, Zhang et al., 2015].

We used ten-fold cross-validation for the experiment. The hyperparameters of each model were selected by grid searches to obtain the highest prediction performances. In detail, the number of neighbors for LNSM was 60, SVMs were run with an RBF kernel and the soft-margin hyperparameter was 1. RF was run with 80 estimators. CCA had 60 pairs of canonical variables, MF had 60 latent factors, and MLF had two hidden layers with sizes of 1000 and 800. DCN had the same architecture described in [Dey et al., 2018] with 4 convolutional and pooling layers.

We calculated the average computational time of each fold. The computational time was evaluated on a computer with Intel Core i7-6700 CPU and 16 GB RAM.

ii.Experimental results

Table 2.9: Performance comparison of drug side effect prediction models on Aeolus dataset and PCBio descriptors. Results for AUC and AUPR contain mean and standard error values in the format $value \times 10^{-2}$.

	Models						
	Non-latent models				Latent models		
	LNSM	SVMs	RF	LR	CCA	MF	MLN
AUC ($\times 10^{-2}$)	86.07 ±0.56	89.26 ±0.47	86.82 ±0.41	89.00 ±0.40	64.51 ±1.05	87.13 ±0.03	**89.55** ±0.39
AUPR ($\times 10^{-2}$)	59.04 ±1.58	67.57 ±1.63	61.92 ±1.11	66.75 ±1.08	34.17 ±2.07	61.03 ±1.13	**68.70** ±1.23
Time (s)	73	22642	181	3658	317	**25**	186

Table 2.10: Summary of the models in terms of performance, non-linearity, and dimensional reduction.

Models		AUC ranking	Time ranking	Non-linearity	Dimensional reduction
Non-latent	LNSM	6	2		
	SVMs	2	8	✓	
	RF	5	3	✓	
	LR	3	6	✓	
Latent	CCA	7	4		✓
	MF	4	1		✓
	MLN	1	4	✓	✓
	DCN	8	7	✓	✓

The results of prediction performances and computational time are presented in Table 2.9. In addition, DCN with 2DChem descriptors achieved 73.80±0.46 in AUC, 39.10±0.63 in AUPR, and 4862(s) of computational time.

The results show that MLN is the model having the highest prediction performances in both AUC and AUPR (89.55 $\times 10^{-2}$ and 68.70 $\times 10^{-2}$). SVMs are the second highest model with 89.26 $\times$ 10^{-2} and 67.57 $\times$ $10^{-}2$ for AUC and AUPR. In terms of computational time, MF is the fastest model, and SVMs is the slowest one. CCA and DCN are the two models having the lowest prediction performances.

We summarize the properties of the models in terms of linearity and dimensional reduction and rank the performances of the models in AUC and computational time as in Table 2.10. This table shows that in balancing between prediction accuracy and computational time, two latent variable models, MLN and MF, are the two most promising ones. In addition, latent variable models learn latent representation vectors of small size for drugs, which are much smaller than the original size of the drug descriptor vectors. This dimensional reduction can help to remove redundant information from drug descriptors. We also can see that three out of the four highest AUC models are non-linear, suggesting that there are non-linear relationships between drug descriptors and drug

side effects.

2.4.3 Task 3: Drug side effect mechanism analysis

The objective of this task is to reveal associated biological components such as proteins or pathways of drug side effects. In this task, non-clinical data of drug-protein interactions, protein-pathways, chemical-pathways is combined with clinical data, usually drug side effect benchmark data. There are two commonly used approaches for this task: *i. using sparse learning* and *ii. using network mining*.

i. Using sparse learning

In the sparse learning approach, the idea is to consider associated biological components of each drug as a feature vector, and then find associated features corresponding to drug side effects. To do this, weight vectors over biological components and drug side effects are used with sparse constraints by applying $L1$ regularization. The remaining subsets with high weights of biological components and drug side effects are associated with each other. We describe two studies using this approach with logistic regression and canonical correlation analysis.

Logistic regression with regularization was proposed to obtain associated biological pathways with each drug side effect [Wallach et al., 2010]. To obtain pathways associated with drug side effect j, let $\mathbf{w}_j \in \mathbb{R}^m$ be weights over m pathways obtaining from:

$$\arg\min_{\mathbf{w}_j} \frac{1}{d} \sum_{i=1}^{d} \Big(- y_{i,j} \log \frac{1}{1+\exp(-\mathbf{h}_i \cdot \mathbf{w}_j)} \\ - (1-y_{i,j}) \log(1 - \frac{1}{1+\exp(-\mathbf{h}_i \cdot \mathbf{w}_j)}) \Big) + \lambda_w \left\| \mathbf{w}_j \right\|_1 .$$

where λ_w is a regularization parameter.

$L1$ regularization $\left\| \mathbf{w}_j \right\|_1$ forces $\mathbf{w}_j$ to be a sparse vector. The corresponding pathways with high weights are associated with drug side effect j.

SCCA was applied to obtain subsets of correlation of drug side effects and pathways [Zheng et al., 2014]. By applying SCCA into two matrices $\mathbf{Y}$ and

H of drug side effect and drug-biological component, respectively, two sparse weight matrices $\mathbf{A} \in \mathbb{R}^{s \times c}$ and $\mathbf{B} \in \mathbb{R}^{m \times c}$ are obtained. The corresponding subsets of drug side effects and pathways of each pair of $(\mathbf{a}_l, \mathbf{b}_l)$ with $l = 1 \ldots c$ are correlated.

ii. Using network mining

The idea of using networks of drug side effect-biological components is similar to mining drug side effect networks for drug side effect prediction. If a biological component and a drug side effect are well-connected in a network of biological component-drug side effects, they are highly associated with each other. The technique was used in [Chen et al., 2013, Jiang et al., 2014, Wang et al., 2013] to build a protein-drug side effect network and discover associated proteins with each drug side effect. Dijkstra algorithm, a well-known method to calculate the shortest paths in a graph, was used on the network of biological components-drug side effects to obtain associated biological pathways of drug side effects [Wang et al., 2011].

2.5. Discussion

This survey addresses drug side effect-related studies in three aspects: data, drug descriptors, and tasks with corresponding methods. We divide data resources into clinical and non-clinical data. Clinical data contains important personal context information such as drug side effects, diseases, dosages of treatments, and demographic information. Non-clinical data contains more detailed information about drugs and biological systems with chemical, and physical properties of drugs, drug-protein interactions, and biological pathways.

We summarize the commonly used drug descriptors in drug side effect studies. In addition to traditional physical and chemical descriptors, many studies integrate biological descriptors of drugs to have better drug information.

There are three main tasks in drug side effect studies: creating drug side effect benchmark data, drug side effect prediction, and drug side effect mechanism analysis. Association rule mining is the commonly used method for creating drug side effect benchmark data. The drug side effect prediction task is

classified into two classes: personalized drug side effect prediction and general drug side effect prediction. In the former class, Poisson models are widely used. In the latter class, the commonly used machine learning models can be categorized into non-latent variable models and latent variable models. The non-latent variable models predict drug side effects without learning latent variables, while the latent variable models learn latent vectors of small size to represent drugs such that these latent vectors can help the prediction efficiently. The experimental results show that MLN is the model having the highest prediction performances, and the latent variable models have the potential for further development. In drug side effect mechanism analysis, using sparse learning and network mining are two commonly used approaches.

From this survey, we have three remarks on problems in the current drug side effect studies as follows in current drug side effect studies as follows:

1) Most drug side effect prediction studies address monopharmacy cases in SIDER benchmark data. There are few studies that proposed models for polypharmacy prediction, for example, predicting with TWOSIDES benchmark data [Zitnik et al., 2018], in spite of the fact that most of the significant drug side effects come from drug combinations [Tatonetti et al., 2012, Zitnik et al., 2018].

2) Drug side effect data resources are not effectively used. Recent drug side effect studies only use either clinical data without non-clinical data information or use drug side effect benchmark data and non-clinical data without personal context information. There are no studies that combine full clinical data with non-clinical data. In addition, current drug side effect benchmark data such as SIDER, OFFSIDES, and TWOSIDES only contain drugs and drug side effects, other personal context information still remains in original clinical records.

3) Machine learning models are mostly used as black boxes for drug side effect prediction since they only output association scores of drugs and drug side effects. In drug side effect discovery, explaining drug side effect mechanisms is a big challenge. It is not only a problem of predicting corresponding drug side effects of drugs but also how drug side effects occur. However, predicting and revealing drug side effect mechanisms are now considered as two separate parts. Designing drug side effect prediction models which reveal related information about drug side effect mechanisms seems to be an important topic.

In conclusion, the use of machine learning models in drug side effect studies is likely to develop in the future. Effectively using available data with suitable models still remains a big challenge. It is not only drug side effect prediction that is an important task but also revealing drug side effect mechanisms is another task to concentrate on.

Chapter 3

CentSmoothie: Learning a single combination of drug properties on a drug-drug interaction hypergraph for predicting drug-drug interactions

3.1. Introduction

A drug-drug interaction (DDI) is a reaction between two drugs, whereby the effects of one drug are modified by the concomitant use of the second drug. A DDI might cause (drug) side effects, which are unwanted effects and are responsible for significant patient morbidity and mortality [Magro et al., 2012]. Therefore, it is a very important task to predict drug-drug interactions to guide drug safety. Given drug information and known (drug) side effects of many pairs of drugs, one wishes to learn a model to predict side effects of all pairs of drugs, which include new pairs of drugs without known side effects or known pairs (to denoise or complete side effect data). DDI data is usually represented as a graph with nodes for drugs, edges for drug pairs that interact, with (binary vector) labels for (known) side effects [Zitnik et al., 2018]. The task is to predict

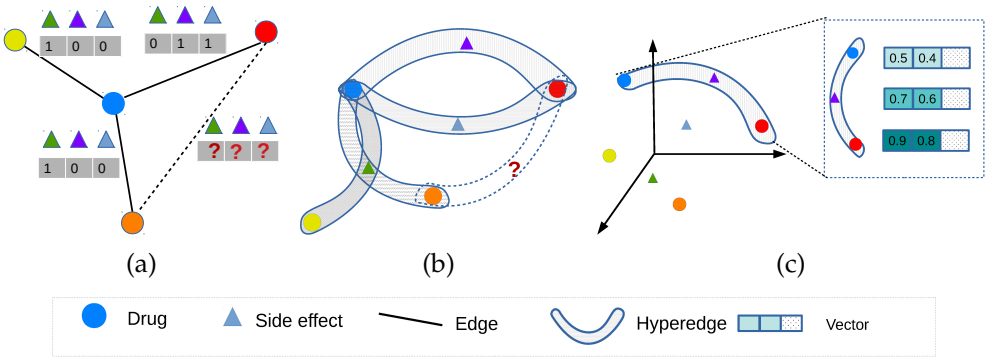


Figure 3.1: Illustrative examples of (a) a traditional graph and (b) a (proposed) hypergraph for drug-drug interactions, and (c) central-smoothing assumption.

labels of all pairs of nodes in the DDI graph. Fig. 3.1a shows an example of a DDI graph, where the dotted edge with question marks is the pair of drugs with labels to be predicted.

Recently, graph neural networks have emerged as a prominent approach for this task with high prediction performance [Feng et al., 2020, Zitnik et al., 2018]. Graph neural networks for predicting DDI have two steps: learning new representations of drugs from a DDI graph and using these representations for predictions. One drawback of this approach is the lack of learning label (i.e. side effect) representations. There are many side effects with complicated relationships. For example, our largest dataset has 964 side effects, where the number of drug pairs for one side effect (positive samples in supervised learning) ranges from 288 to 22,520. Previous methods represent each side effect as an independent one-hot vector, potentially under-utilizing the relationship among side effects [Chu et al., 2019, Feng et al., 2020, Zitnik et al., 2018]. Considering the relationship between side effects would be beneficial for predicting side effects, especially the ones with only small numbers of positive samples (i.e. infrequent side effects). Hence, it is desirable to learn the representations for both drugs and side effects, namely both nodes and edge labels, together.

To this end, we propose to encode DDI data with a hypergraph [Nguyen and Mamitsuka, 2020]. A node in the hypergraph can be either a drug or a side effect. A hyperedge is a triple of two drugs and a side effect that they caused. Hence, a pair of drugs with multiple side effects (interactions) will

result in many hyperedges in the hypergraph. Fig. 3.1b illustrates an example of a hypergraph corresponding to the DDI graph in Fig. 3.1a. Existing learning methods of hypergraph neural networks are based on a *smoothing assumption* that the representations of nodes in a hyperedge should be close to each other [Bai et al., 2019, Feng et al., 2019]. However, this assumption is not necessarily appropriate for our DDI problem, since each node representation should reflect (chemical or biological) properties of the corresponding drug and interacting drugs do not necessarily need to have similar properties.

We propose CentSmoothie, a central-smoothing hypergraph neural network that uses our idea, *central-smoothing assumption* (see Fig. 3.1c) for each hyperedge in the hypergraph for DDI. The idea is to learn k-dimensional representation vectors for nodes in a hyperedge such that (i) a drug node representation reflects the property of the corresponding drug and (ii) a side effect node representation reflects a combination of some properties of the two drugs that cause the corresponding side effect [Corrie and Hardman, 2011]. To implement (ii), we first assume that a side effect representation should be related to the midpoint of the representations of the two interacting drugs, reflecting the combination of the two drug properties. Furthermore, there might have different side effects of the same two drugs, suggesting that each side effect might be obtained by a partial combination of the two drug properties. Hence, we propose that the representation for each side effect is learned to be close to a weighted midpoint of the corresponding two drug representations.

We formulate the above assumption, and then define the central-smoothing hypergraph Laplacian to be used in each layer of the hypergraph neural network with spectral convolution [Feng et al., 2019]. We also provide a computational method with the complexity of O(n) for the proposed hypergraph Laplacian.

We conducted extensive experiments to verify the performance advantages of CentSmoothie in both synthetic and real datasets. Our experimental results demonstrated that CentSmoothie significantly outperformed existing spectral-based convolutional hypergraph neural networks in all cases. In particular, CentSmoothie achieved higher performances over baselines for real datasets with more infrequent side effects, which are more difficult to predict, justifying

the benefit of learning label (side effect) representations.

3.2. Related Work

Existing work in predicting DDI can be divided into two approaches: non-graph based and graph based ones. In the non-graph based approach, predefined feature vectors, indicating the existence of chemical substructures and interacting proteins of drugs, are used. The side effects can be predicted by using a model (for example, a multilayer feedforward neural network), which receives the feature vectors of two drugs as input and the vector indicating the side effects of the two drugs as output [Chu et al., 2019, Rohani and Eslahchi, 2019].

In the graph based approach, topological information of graphs is used to enhance the representations of nodes, leading to higher performance than the non-graph based approach. There are two types of graphs that can be used: molecular graphs of drugs and a DDI graph. For a DDI graph where nodes are drugs and edges are interactions between drugs, graph neural networks (GNNs) are applied to learn a new representation of a drug node based on its neighbors. Recent results show that GNNs for predicting DDI achieve cutting-edge performance [Feng et al., 2020, Zitnik et al., 2018]. An extension of a DDI graph can be a DDI heterogeneous graph, where nodes are drugs and side effects and edges are pairs of interacting drugs or drug-side effects [Zhang et al., 2019a]. However, the DDI heterogeneous graph cannot preserve triples of drug-drug-side effects.

GNNs can be further divided into two approaches: spectral convolution and spatial convolution [Wu et al., 2020]. In the spectral convolution, at first, the graph Laplacian is defined, and then each GNN layer is constructed from the graph Fourier transformation given the graph Laplacian [Feng et al., 2019, Kipf and Welling, 2016]. The spatial convolution approach uses node spatial relation that a node is updated based on information from neighbor nodes [Gilmer et al., 2017, Zhang et al., 2019a].

Different from existing work for predicting drug-drug interactions, we formulate the drug-drug interactions in the form of a hypergraph and develop a

new hypergraph neural network (HGNN) on the DDI hypergraph.

In HGNNs, recent work has inherited the spectral convolution approach on graphs to adapt to hypergraphs by defining the hypergraph Laplacian [Feng et al., 2019]. Once the hypergraph Laplacian is defined, HGNNs can be constructed in the same manner as that for GNNs. Another approach for HGNNs is the spatial convolution approach with attention mechanisms [Bai et al., 2019].

3.3. Background

In this section we briefly describe the hypergraph Laplacian being derived from a smoothness measure [Nguyen and Mamitsuka, 2020]. Let $G = (V, E)$ be a general hypergraph, where V is the node set and $E \subset 2^V$ is the hyperedge set. Let $W = diag(w(e_1), ..., w(e_{|E|})) \in \mathbb{R}^{|E|\times|E|} \succcurlyeq \mathbf{0}$ be the diagonal matrix that $w(e)$ is the weight of hyperedge e. Let $x \in R^{|V|}$ be values of nodes on the hypergraph that x_u is the value of x at node u.

The hypergraph Laplacian is usually defined to be used in a similar manner to the graph Laplacian: to evaluate the smoothness of a function on a graph. Let $sh(x, G)$ be a smoothness measure of x on G and $ss(x, e)$ be a smoothness measure of x on hyperedge e. The smoothness on the hypergraph usually has the following form [Nguyen and Mamitsuka, 2020]:

$$sh(x, G) = \mathcal{T}_{e\in E} w(e) ss(x, e) \tag{3.1}$$

where $\mathcal{T}$ is an aggregation operator, such as sum (the most commonly used one), max, or l_p norm [Nguyen and Mamitsuka, 2020]. The usual smoothing assumption on hypergraphs is that nodes within a hyperedge should be close to each other [Bai et al., 2019, Chan and Liang, 2020, Feng et al., 2019], and then the smoothness measure on each hyperedge is calculated by:

$$ss(x, e) = \sum_{(u,v)\in e} (x_u - x_v)^2. \tag{3.2}$$

When $\mathcal{T}$ is a sum operator, the smoothness of a function on a hypergraph can

be found in the following form:

$$sh(x, G) = \sum_{e \in E} w(e) \sum_{(u,v) \in e} (x_u - x_v)^2 \tag{3.3}$$

$$= x^{\mathbf{T}} L x \tag{3.4}$$

which has the quadratic form with L, and L is then called the hypergraph Laplacian of the hypergraph. In the next section, we will propose a new smoothing assumption on hypergraphs and then define a new hypergraph Laplacian.

3.4. CentSmoothie: Central-Smoothing Hypergraph Neural Networks

3.4.1 Problem Setting

We formulate the problem of predicting DDI as follows.

Input: Given a hypergraph of drug-drug interactions: $G = (V, E)$, where the node set $V = V_D \cup V_S$ consists of a drug node set V_D and a side effect node set V_S, a known hyperedge set $E \subset V_D \times V_D \times V_S$ (Since two drugs in a drug pair are unordered, two triples (u, v, t) and (v, u, t) ($u, v \in V_D$ and $t \in V_S$) are the same), and the feature vectors of drugs: $X_D \in R^{|V_D| \times K_0}$, where K_0 is the feature size. The feature vectors of side effects are one-hot vectors.

Output: For each triple $e = (u, v, t) \in V_D \times V_D \times V_S$, t is predicted to be a side effect of u and v if the score of the triple is larger than a threshold.

3.4.2 Central-Smoothing Hypergraph Laplacian

The key idea is a central-smoothing assumption: each hyperedge is called *central-smooth* if a weighted version of the midpoint of drug node representations is close enough to the representation of the side effect node. It is motivated by biological research that a side effect of a pair of drugs is caused by a combination of properties of the two drugs [Corrie and Hardman, 2011]. Assuming that representations reflecting all properties of drugs are obtained in a k-dimensional space, the combination containing properties of the two drugs

should reflect corresponding side effects. We show that among commonly used combination operators: average, concatenation, max-pooling, and min-pooling, the average (also the midpoint) is a good option. First, our operator for combining two drug properties for side effects needs to satisfy the following two criteria: (i) order invariance in the k-dimensional space since the drug pair has no order and (ii) effects of both positive and negative embedding values must be kept to cover the whole embedding space. We can see that concatenation violates (i) and max-pooling and min-pooling violate (ii), but the average (midpoint) satisfies both criteria. In addition, a weighted midpoint, which in the ideal case, would contain properties from each drug, represents a specific combination of the properties, potentially reflecting the cause of a side effect.

Central-smoothing measure on a hyperedge. In the embedding space of K-dimension, considering dimension k with the embedding of nodes: $X_k \in \mathbb{R}^{|V|}$ that $X_{k,u} \in \mathbb{R}$ is the embedding of node $u \in V$. Given a hyperedge $e = (u, v, t)$, a weight $W_{k,t} \in \mathbb{R}^+$ is a parameter indicating the relevance of side effect t on dimension k. We assign the weight of side effect t to the hyperedge ($w_k(e) = W_{k,t}$), and let $\mathbf{W}_k = diag(w_k(e_1), ..., w_k(e_{|E|}))$ be the diagonal matrix of the hyperedge weights. The central-smoothing measure on dimension k of the hyperedge is defined as:

$$ss^c(X_k, e) = W_{k,t}(\frac{X_{k,u} + X_{k,v}}{2} - X_{k,t})^2. \tag{3.5}$$

Central-smoothing measure on the hypergraph. For hypergraph G, the central-smoothing measure on dimension k is defined as the sum of the central-smoothing measures on all hyperedges:

$$sh^c(X_k, G) = \sum_{e \in E} W_{k,t}(\frac{X_{k,u} + X_{k,v}}{2} - X_{k,t})^2. \tag{3.6}$$

Central-smoothing hypergraph Laplacian. Since $sh^c(X_k, G)$ is a nonnegative quadratic form, there exists a $\mathbf{L}_k \in \mathbb{R}^{|\mathbb{V}| \times |\mathbb{V}|}$ such that $sh^c(X_k, G) = X_k^T \mathbf{L}_k X_k$. We call $\mathbf{L}_k$ as the *central-smoothing hypergraph Laplacian*, which can be derived as follows.

Let $H \in \mathbb{R}^{|V| \times |E|}$ be a weighted oriented incidence matrix of G that for a

hyperedge $e \in E$, $H_{u,e} = H_{v,e} = \frac{1}{2}$ and $H_{t,e} = -1$, we have:

$$\begin{aligned} sh^c(X_k, G) &= \sum_{e \in E} W_{k,t}(\frac{X_{k,u} + X_{k,v}}{2} - X_{k,t})^2 \\ &= X_k^T H \mathbf{W}_k H^\mathsf{T} X_k \qquad (3.7) \\ &\stackrel{\text{def}}{=} X_k^T \mathbf{L}_k X_k. \qquad (3.8) \end{aligned}$$

Then,

$$\mathbf{L}_k = H \mathbf{W}_k H^\mathsf{T}. \qquad (3.9)$$

Proof: Let $H_{\cdot,e} \in \mathbb{R}^{|V| \times 1}$ be the column of H corresponding to hyperedge e. We have:

$$\begin{aligned} &\sum_{e \in E} W_{k,t}(\frac{X_{k,u} + X_{k,v}}{2} - X_{k,t})^2 \\ &= \sum_{e \in E} (\frac{X_{k,u} + X_{k,v}}{2} - X_{k,t}) W_{k,t} (\frac{X_{k,u} + X_{k,v}}{2} - X_{k,t}) \\ &= \sum_{e \in E} (X_k H_{\cdot,e}) W_{k,t} (X_k H_{\cdot,e}) \\ &= X_k^\mathsf{T} H \mathbf{W}_k H^T X_k \qquad \square \end{aligned}$$

Computing the central-smoothing hypergraph Laplacian. The central-smoothing hypergraph Laplacian $\mathbf{L}_k$ in (3.9) can be computed with the time complexity of $O(|E|)$. Concretely, each element $\mathbf{L}_{k,i,j}$ can be computed by:

$$\mathbf{L}_{k,i,j} = \sum_{e \in E | i,j \in e} w_k(e) H_{i,e} H_{j,e}. \qquad (3.10)$$

We have four cases:

- $\mathbf{L}_{k,i,j} = \mathbf{L}_{k,j,i} = \frac{1}{4} \sum_{t \in V_s | (i,j,t) \in E} W_{k,t}$ if $i! = j \in V_D$.
- $\mathbf{L}_{k,i,j} = \mathbf{L}_{k,j,i} = -\frac{1}{2} n_d(i,j) W_{k,j}$ if $i \in V_D, j \in V_S$.
- $\mathbf{L}_{k,i,i} = \frac{1}{4} \sum_{t | t \in V_S} m_d(i,t) W_{k,t}$ if $i \in V_D$.
- $\mathbf{L}_{k,i,i} = q(i) W_{k,i}$ if $i \in V_S$.

where $n_d(i,j) = |\{(u,v,j) \in E | u = i \vee v = i\}|$, $m_d(i,t) = |\{u|(i,u,t) \vee (u,i,t) \in E\}|$, $q(i) = |\{(u,v,i)|(u,v,i) \in E\}|$.

Proof: Given the formulation for L_k:

$$\mathbf{L}_{k,i,j} = \sum_{e \in E | i,j \in e} w_k(e) H_{i,e} H_{j,e}.$$

We have:

1. $i, j \in V_D, i! = j$, meaning that $H_{i,e} = H_{j,e} = \frac{1}{2}$, hence:

$$\begin{aligned}
\mathbf{L}_{k,i,j} &= \sum_{e \in E | i,j \in e} w_k(e) H_{i,e} H_{j,e} \\
&= \frac{1}{4} \sum_{e \in E | i,j \in e} w_k(e) \\
&= \frac{1}{4} \sum_{t \in V_S | e=(i,j,t) \in E} W_{k,t}
\end{aligned}$$

2. $i \in V_D, j \in V_S$, meaning that $H_{i,e} = \frac{1}{2}$ and $H_{j,e} = -1$, hence:

$$\begin{aligned}
\mathbf{L}_{k,i,j} = \mathbf{L}_{k,j,i} &= \sum_{e \in E | i,j \in e} w_k(e) H_{i,e} H_{j,e} \\
&= \frac{-1}{2} \sum_{e \in E | i,j \in e} w_k(e) = \frac{-1}{2} \sum_{e \in E | i,j \in e} W_{k,j} \\
&= \frac{-1}{2} W_{k,j} \sum_{e \in E | i,j \in e} 1 \\
&= \frac{-1}{2} W_{k,j} \sum_{e=(u,v,j) \in E | u=i \vee v=i} 1 \\
&= \frac{-1}{2} W_{k,j} n_d(i,j)
\end{aligned}$$

where $n_d(i,j) = |\{(u,v,j) \in E | u = i \vee v = i\}|$.

3. $i = j \in V_D$, $H_{i,e} = H_{j,e} = \frac{1}{2}$, hence:

$$\begin{aligned}\mathbf{L}_{k,i,i} &= \sum_{e=(u,v,t)\in E | u=i \vee v=i} w_k(e) H_{i,e} H_{i,e} \\ &= \frac{1}{4} \sum_{e=(u,v,t)\in E | u=i \vee v=i} w_{k,t} \\ &= \frac{1}{4} \sum_{t\in V_S} w_{k,t} \sum_{e=(u,v,t)\in E | u=i \vee v=i} 1 \\ &= \frac{1}{4} \sum_{t\in V_S} w_{k,t} m_d(i,t)\end{aligned}$$

where $m_d(i,t) = |\{u | (i,u,t) \vee (u,i,t) \in E\}|$.

4. $i = j \in V_S$, meaning that $H_{i,e} = H_{j,e} = -1$, hence:

$$\begin{aligned}\mathbf{L}_{k,i,i} &= \sum_{e=(u,v,t)\in E | u=i \vee v=i} w_k(e) H_{i,e} H_{i,e} \\ &= \sum_{e=(u,v,i)\in E} w_k(e) = W_{k,i} \sum_{e=(u,v,i)\in E} 1 \\ &= W_{k,i} q(i)\end{aligned}$$

where $q(i) = |\{(u,v,i) | (u,v,i) \in E\}|$.

Complexity analysis. Given N convolution layers, the computational complexity for all central-smoothing hypergraph Laplacian is $O(N \cdot K \cdot |E|)$. Each $\mathbf{L}_k$ can be computed with a complexity of $O(|E|)$ by iterating over all hyperedges in E once, and for each hyperedge, the side effect weight is added to the corresponding elements in $\mathbf{L}_k$ and we have $N \cdot K$ Laplacian matrices to compute. We note that K here is referred to the size of latent features, and this is not the original input features. In practice, even if the size of the original input features is very large, the number of latent features can be very small (≤ 200), which is computationally tractable.

Non-weighted version. In our experiments, we will examine the need for the weight of each side effect. So we here show a non-weighted version of central-smoothing hypergraph Laplacian, called CentSimple by fixing $\mathbf{W}_k$ to be an identity matrix, where the central-smoothing hypergraph Laplacian in (3.9) becomes $\tilde{\mathbf{L}}_k = HH^\top$.

3.4.3 Central-Smoothing Hypergraph Neural Networks (HGNNs)

Transforming input features to latent spaces

We first transform the input feature vector of drugs and one-hot vector of side effects to the K-dimension latent space by using a two-layer feedforward neural network for drugs, and a one-layer feedforward neural network (as an embedding table) for side effect, respectively, as follows:

$$X_D^{(0)} = f_D(X_D) \tag{3.11}$$

$$X_S^{(0)} = f_S(X_S), \tag{3.12}$$

where $X_D \in \mathbb{R}^{|K_0| \times |V_D|}$ is the drug input features with feature size K_0, $X_S \in \mathbb{R}^{|V_S| \times |V_S|}$ is the one-hot vector of side effect, $X_D^{(0)} \in \mathbb{R}^{K \times |V_D|}$, $X_S^{(0)} \in \mathbb{R}^{K \times |V_S|}$ and f_D and f_S are the corresponding feedforward neural networks.

Convolution layers on the latent spaces

We adapt HGNN layers [Feng et al., 2019] using $\mathbf{L}_k$ at dimension k. Given hypergraph Laplacian $\mathbf{L}_k$, we have the normalized adjacency matrix with a self-loop at each node:

$$\tilde{A}_k = 2I - d_{\mathbf{L}_k}^{-1/2} \mathbf{L}_k d_{\mathbf{L}_k}^{-1/2} \tag{3.13}$$

where $d_{\mathbf{L}_k}$ is the degree matrix, corresponding to Laplacian $\mathbf{L}_k$ and I is the identity matrix.

Let $\tilde{D}_k$ be the corresponding degree matrix of $\tilde{A}_k$, each layer of central-smoothing HGNNs has the following form:

$$X^{(l+1)} = \sigma(\tilde{X}^{(l+1)} \Theta^{(l)}), \tag{3.14}$$

where $\tilde{X}^{(l+1)} = [\tilde{x}_1^{(l+1)}, ..., \tilde{x}_K^{(l+1)}]$ and $\tilde{x}_k^{(l+1)} = \tilde{D}_k^{-1/2} \tilde{A}_k \tilde{D}_k^{-1/2} x_k^{(l)}$, $\Theta^{(l)} \in \mathbb{R}^{K \times K}$ is the parameters for the transformation from layer (l) to layer $(l+1)$, and σ is an activation function.

3.4.4 Predicting Drug-Drug Interactions

Assuming that $X^{*\mathsf{T}} \in \mathbb{R}^{|V| \times K}$ is the final node representation with learned weights $W^* = \{W_k^* | k = 1...K\}$. For all $e = (u, v, t)$, t is predicted to be a

side effect of u and v if the representation of t is close enough to the weighted midpoint of the two drug node representations (computed by score function $p(e, X^*, W^*)$). First, we compute smoothness measures $ssa(e, X^*, W^*)$ of (u, v, t) on all dimensions:

$$ssa(e, X^*, W^*) = \sum_{k=1}^{K} W^*_{k,t}(\frac{X^*_{k,u} + X^*_{k,v}}{2} - X^*_{k,t})^2. \tag{3.15}$$

Then, the prediction score is defined to be:

$$p(e, X^*, W^*) = \frac{1}{1 + ssa(e, X^*, W^*)}. \tag{3.16}$$

If $p(e, X^*, W^*) > h$, a predefined threshold, then t is predicted to be a side effect of u and v.

3.4.5 Objective Function of CentSmoothie

Let $\bar{E} = V_D \times V_D \times V_S \setminus E$ be the complement of the hyperedge set. The objective function to train CentSmoothie is to maximize the score $p(e, X^*, W^*)$ of the known hyperedges and minimize the score of the complement set $\bar{E}^*$. Then the objective function can be defined as:

$$\min_{W^* \geq 0, X^*} f(X^*, W^*) = \sum_{e \in E} (1 - p(e, X^*, W^*))^2 \tag{3.17}$$

$$+ \lambda \sum_{e \in \bar{E}} p(e, X^*, W^*)^2, \tag{3.18}$$

where λ is a hyperparameter.

In practice, as $|\bar{E}|$ is too large, we randomly sample a subset of $\Omega \subset \bar{E}$, $|\Omega| = |E|$ to replace $\bar{E}$ in the objective function to reduce the computational cost (A CentSmoothie implementation is available at https://github.com/anhnda/CentSmoothieCode.). To keep the non-negative constraint on W^*, we used a projected gradient descent [Lin, 2007].

3.5. Experiments

We conducted experiments to evaluate the performance of our proposed method, CentSmoothie, a hypergraph neural network with a central-smoothing assumption, in two scenarios: (i) a synthetic dataset and (ii) three real DDI datasets. On the synthetic dataset, we aimed to validate that CentSmoothie could achieve higher performances than traditional hypergraph neural networks, by using the data generated from the central-smoothing assumption. On the real DDI datasets, we examined the performance of CentSmoothie in comparison with baseline models, to prove that the central-smoothing assumption is suitable for DDI data.

For both scenarios, we used 20-fold cross-validation using the mean AUC (area under the ROC curve) and the mean AUPR (area under the precision-recall curve) with standard deviations, to validate the prediction performances [Zitnik et al., 2018].

For graph and hypergraph neural networks, the numbers of layers and the embedding sizes were in [1, 2, 3] and [10, 20, 30], respectively. The activation function was rectified linear unit (ReLu). The hyperparameter λ was fixed: 0.01. The results obtained were the highest performances with the number of layers of 2 and the embedding size of 20 for all methods. All experiments were run on a computer with Intel Core I7-9700 CPU, 8 GB GeForce RTX 2080 GPU, and 32 GB RAM.

3.5.1 Synthetic Data

Generation

The idea to generate synthetic data is that each drug has several groups of features and the combination of two groups of features leads to a side effect of the drugs. The generation process consists of three steps:

- Step 1: Generating groups of features and their combinations. Suppose that there were n groups of features: $G = \{g_1, ..., g_n\}$. There are maximally $\frac{n(n-1)}{2}$ group combinations: $P = \{(g_i, g_j) | i = 1...n, j = i + 1...n\}$.

Each group combination $p_i \in P, i = 1...|P|$ is assigned with a side effects s_i.

- Step 2: Generating drug features. Let a be the number of features in a group, D be the number of drugs, and m be the maximum number of groups of features for each drug.

  For each drug i, we first uniformly sampled the number of groups $1 \leq n_i \leq m$ and then sample n_i groups from G. Let $G_i \in G$ be the sampled groups of drug i. Let the binary vector $\mathbf{b}_i \in \mathbb{R}^{a.n}$ indicated the existence of features for drug d_i that $\mathbf{b}_i(j) = 1$ if $\lfloor j/a \rfloor \in G_i$, otherwise $\mathbf{b}_i(j) = 0$.

  The feature vector of drug i was sampled from a Gaussian distribution with mean $\mathbf{b}_i$ and variance σ: $\mathbf{f}_i = Gaussian(\mathbf{b}_i, \sigma)$.

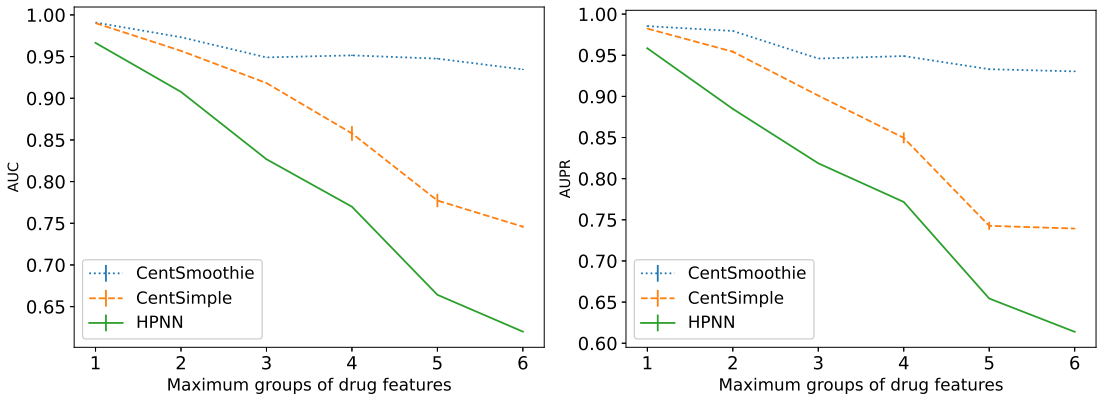
- Step 3: Generating triples of drug-drug and side effects. For each pair of two drugs generated from Step 2, we matched the group combinations of the two drugs with the corresponding side effects from Step 1. For a pair of two drugs i and j with corresponding groups G_i and G_j, let $P_{ij} = G_i \times G_j$ and $S_{ij} = \{s_t | p_t \in P_{ij}\}$, we generated the triples: $E_{ij} = \{(d_i, d_j, s_t) | s_t \in S_{ij}\}$.

By going through all pairs of drugs, we obtained the synthetic data set with the drug feature vectors $F = \{f_i | i = 1...n\}$ and the triples of drug-drug-side effect $E = \cup_{i=1...n, j=i+1...n} E_{ij}$.

We set the number of groups $n = 10$, the number of features in each group $a = 3$, the variance $\sigma = 0.01$, and the number of drugs $D = 500$. We changed m in the range of $[1, 2, \cdots 6]$.

Comparing Methods

For the synthetic dataset, we used the central-smoothing hypergraph neural networks CentSmoothie, the non-weighted central-smoothing hypergraph neural networks CentSimple, and the existing spectral based hypergraph neural network, HPNN [Feng et al., 2019].



(a) (b)

Figure 3.2: Synthetic data performance comparison: (a) AUC and (b) AUPR.

Results

Fig. 3.2 shows the AUC and AUPR of each compared method, obtained by changing the maximum number of groups of features for drugs. We could easily see that CentSmoothie achieved the highest AUC and AUPR scores for all values of x-axis, followed by CentSimple and then HPNN. In particular, the AUC scores of CentSmoothie were always higher than 0.95, while those of HPNN decreased when drugs are more complex with larger numbers of groups of drug features. This clearly showed that CentSmoothie could correctly capture the patterns generated by the central smoothing assumption, particularly for larger numbers of groups of drug features. Similarly, the AUC scores of CentSimple decreased with higher maximum numbers of groups of features, e.g. around 0.75 at 6. The pattern for AUPR scores was also similar to that of AUC scores. This result showed that CentSmoothie could learn different side effects for drug pairs more effectively than CentSimple, implying the significance of using a weight for each side effect in CentSmoothie.

3.5.2 Real Data

Data description

We used three real DDI datasets: TWOSIDES, CADDDI, and JADERDDI. TWOSIDES is a public dataset for DDI extracted from the FDA adverse event reporting system (US database) [Tatonetti et al., 2012]. To our knowledge, TWOSIDES is the largest and commonly used benchmark dataset for DDI [Rohani and Eslahchi, 2019, Xu et al., 2019, Zitnik et al., 2018]. In a similar manner as in [Tatonetti et al., 2012] of TWOSIDES, we used significant tests to generate two new DDI datasets: CADDDI from Canada vigilance adverse reaction report (Canada database, from 1965 to Feb 2021) [Canada Vigilance Program , 2021] and JADERDDI from The Japanese Adverse Drug Event Report (Japanese database, from 2004 to March 2021) [Pharmaceutical and Medical Devices Agency , 2021].

The detail of the new dataset extraction is as follows:

For Canada vigilance adverse and JADERDDI from The Japanese Adverse Drug Event Report, each database consists of reports such that each report contains drugs and the corresponding observed side effects of a patient.

The extraction from these databases was that for each drug pair, we divided the reports into two groups: an exposed group for the reports having the drug pair and a nonexposed group for the reports not having the drug pair. Then, for each side effect, Fisher's exact test with the threshold p-value of 0.05 was used to check if the occurrence rate of the side effect in the exposed group was significantly higher than in the nonexposed group.

Finally, we obtained a set of significant triples of drug-drug-side effects for each database.

Regarding the overlapping of the datasets, between TWOSIDES and CADDDI, there is 24.8% overlapping in side effect names and 59.8% overlapping in drug names. For JADERDDI, we used Google service to translate Japanese drug names to English, mostly written in Katakana, which are more reliable to translate. The overlapping in drug names of TWOSIDES and JADERDDI is 15%. We did not calculate the overlapping of side effects in JADDERDDI since the side effect names were not translated.

We only selected small molecular drugs appearing in DrugBank [Wishart et al., 2018]. Each drug feature vector was a binary vector with a size of 2,329, indicating the existence of 881 substructures and 1,448 interacting proteins [Nguyen et al., 2021]. The statistics of the final datasets is shown in Table 3.1.

Table 3.1: Statistics of the three real datasets.

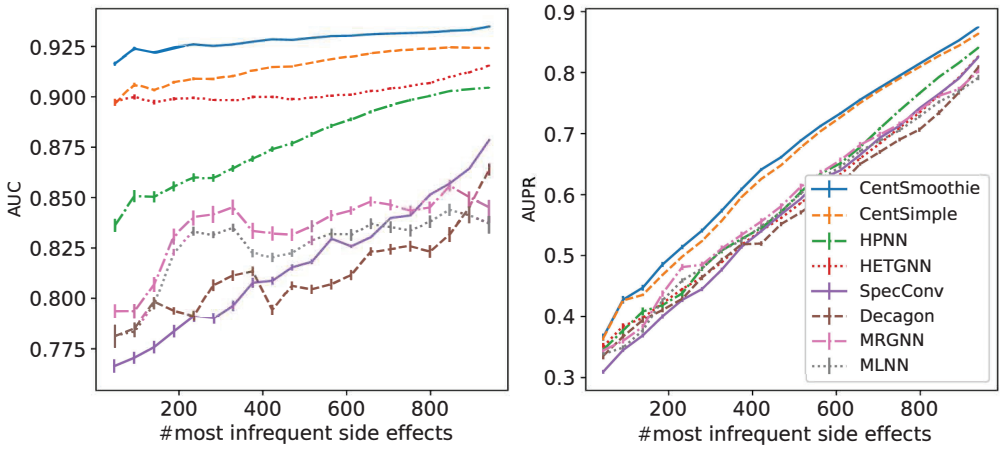
Dataset	#drugs	#side effects	#drug-drugs	#drug-drug-side effects	Avg. side effects/ drug-drugs	drug-drugs/ side effects		
						Min	*Max*	*Avg*
TWOSIDES	557	964	49,677	3,606,046	72.58	288	22,520	3740.7
CADDDI	587	969	21,918	373,976	17.06	89	3288	385.9
JADERDDI	545	922	36,929	222,081	6.01	60	1922	240.9

Table 3.2: Comparison of performances of the methods on the real DDI datasets.

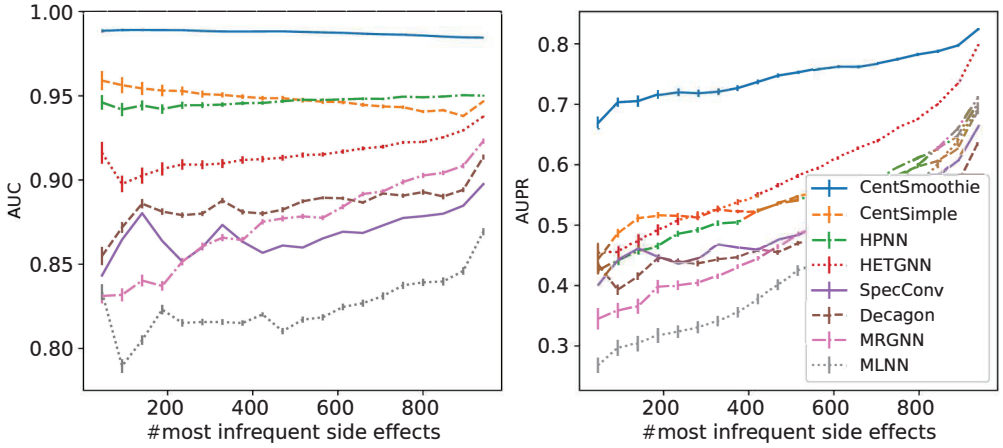
Method	TWOSIDES		CADDDI		JADERDDI	
	AUC	*AUPR*	*AUC*	*AUPR*	*AUC*	*AUPR*
MLNN	0.8372 ± 0.0050	0.7919 ± 0.0041	0.8689 ± 0.0021	0.6927 ± 0.0082	0.8578 ± 0.0015	0.3789 ± 0.0020
MRGNN	0.8452 ± 0.0036	0.8029 ± 0.0039	0.9226 ± 0.0015	0.7113 ± 0.0031	0.9049 ± 0.0009	0.3698 ± 0.0019
Decagon	0.8639 ± 0.0029	0.8094 ± 0.0024	0.9132 ± 0.0014	0.6338 ± 0.0029	0.9099 ± 0.0012	0.4710 ± 0.0027
SpecConv	0.8785 ± 0.0025	0.8256± 0.0022	0.8971 ± 0.0055	0.6640 ± 0.0014	0.8862 ± 0.0025	0.5162 ± 0.0047
HETGNN	0.9113 ± 0.0004	0.8267 ± 0.0005	0.9371 ± 0.0004	0.7974 ± 0.0011	0.8989 ± 0.0007	0.5618 ± 0.0012
HPNN	0.9044 ±0.0003	0.8410 ± 0.0007	0.9495 ± 0.0004	0.7020 ± 0.0018	0.9127 ± 0.0004	0.5198 ± 0.0016
CentSimple	0.9242 ± 0.0003	0.8638 ± 0.0011	0.9584 ± 0.0005	0.6890 ± 0.0016	0.9239 ± 0.0007	0.5349 ± 0.0021
CentSmoothie	**0.9348** ± 0.0002	**0.8749** ± 0.0013	**0.9846** ± 0.0001	**0.8230** ± 0.0019	**0.9684** ± 0.0004	**0.6044** ± 0.0025

Comparing Methods

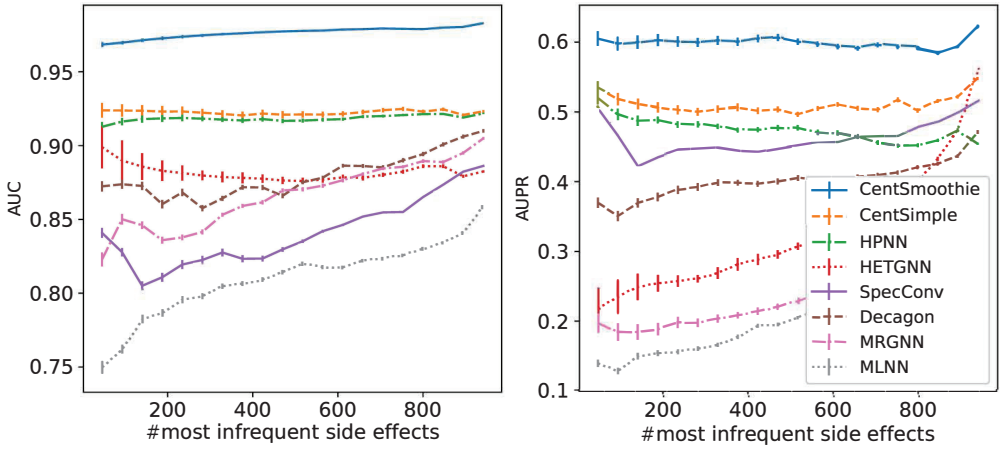
On the real datasets, we compared our proposed methods to baselines: none-graph based, graph based, and hypergraph based methods. For the none-graph based method, we used a multi-layer feedforward neural network (MLNN) [Rohani and Eslahchi, 2019]. For graph neural networks, on the drug molecular graphs, we used MRGNN [Xu et al., 2019] with the recommended hyperparameter settings. On the DDI graph, we used Decagon [Zitnik et al., 2018], a spatial convolution, SpecConv (a spectral convolution graph neural networks) [Kipf and Welling, 2016], and HETGNN (a heterogeneous graph neural network) [Zhang et al., 2019a]. For hypergraph neural networks, we used the existing spectral convolution hypergraph neural network, HPNN [Feng et al., 2019]. We also showed the results of CentSimple to see the effect of central-smoothing without having weights for side effects.



(a)



(b)



(c)

Figure 3.3: Performance comparison (AUC (left) and AUPR (right)) on (a) TWOSIDES, (b) CADDDI and (c) JADDERDDI.

Results

Table 4.2 shows the AUC scores and AUPR scores of all methods. We could see that again CentSmoothie achieved the highest AUC and AUPR scores in all three datasets. For TWOSIDES, CentSmoothie achieved 0.9348 in AUC and 0.8749 in AUPR, followed by CentSimple (0.9242 and 0.8638), HPNN (0.9044 and 0.8410), HETGNN (0.9113 and 0.8267), SpecConv (0.8785 and 0.8256), Decagon (0.8639 and 0.8094), MRGNN (0.8452 and 0.8029), and MLNN (0.8372 and 0.7919).

For CADDDI and JADERDDI, CentSmoothie had the highest performances with AUC and AUPR: (0.9845 and 0.8230) and (0.9684 and 0.6044), respectively. The second and third best methods were CentSimple and HPNN, respectively.

In particular, in AUC, there existed two clear performance gaps. The first one was between hypergraph based methods (CentSmoothie, CentSimple, and HPNN) and non-hypergraph based methods (HETGNN, SpecConv, Decagon, MRGNN, and MLNN). The second one was between CentSmoothie and (CentSimple and HPNN). The first gap showed the advantage of using hypergraph based method for predicting drug-drug interaction. The second gap showed the advantage of central smoothing over regular smoothing. In addition, we could see the importance of learning weights for each side effect to improve the prediction performance.

In AUPR, there was a clear gap between CentSmoothie and the remaining methods. This again showed the advantage of learning weights under the central smoothing assumption for predicting DDI.

CentSmoothie can learn the representations of side effects together with drugs to leverage the relationships of side effects (see the supplement for representation visualization of side effects). These side effect representations might be useful for infrequent side effects which are harder to predict due to the scarcity of positive training data. Fig. 3.3 showed the AUC (left) and AUPR (right) scores of the methods on the subset of most infrequent side effects, obtained by starting with the most infrequent side effect and adding the next infrequent side effects to the subset. From both AUC and AUPR scores in Fig. 3.3, we could see that CentSmoothie achieved the best performances for all values of x-axis (the rightmost point of x-axis corresponds to using all side effects), being followed by CentSimple and HPNN.

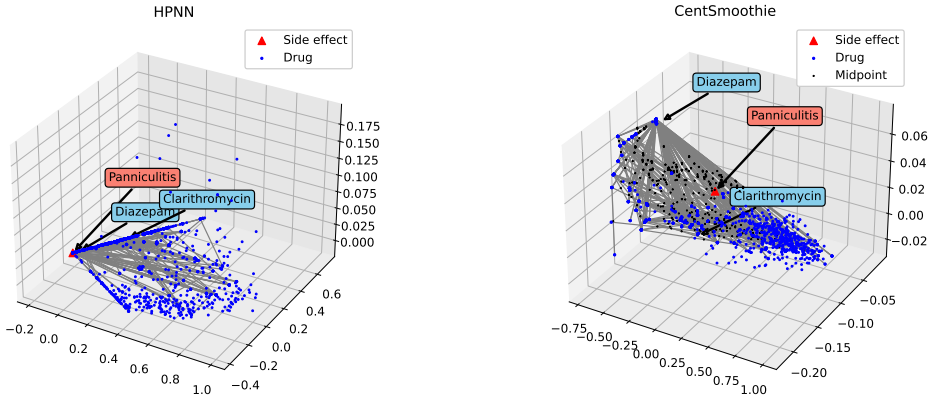
Case studies for predicting unknown drug pairs on infrequent side effects

Drug pair	Rank (Score)			Literature
	CentSmoothie	HPNN	Decagon	
Ranitidine, Pioglitazone	1(0.94)	10(0.53)	-	✓
Diazepam, Clarithromycin	2(0.94)	7(0.57)	139(0.27)	✓
Folic Acid, Metoclopramide	3(0.89)	12(0.50)	62(0.40)	-
Fexofenadine, Furosemide	4(0.88)	6(0.58)	34(0.47)	✓
Metronidazole, Salbutamol	5(0.87)	5(0.59)	1(0.61)	✓
Zolpidem, Warfarin	6(0.85)	1(0.66)	91(0.34)	✓
Salbutamol, Warfarin	7(0.85)	2(0.66)	-	✓
Sertraline, Hydrochlorothiazide	8(0.85)	4(0.62)	130(0.29)	-
Warfarin, Tolterodine	9(0.84)	17(0.45)	-	✓
Acetaminophen, Amoxicillin	10(0.82)	13(0.48)	61(0.40)	✓

Table 3.3: Predictions of unknown drug pairs for an infrequent Panniculitis side effect, top-ranked by CentSmoothie (trained with TWOSIDES) with prediction scores and the literature support.

We showed sampled results obtained by CentSmoothie trained with the largest dataset (TWOSIDES), for predicting unknown drug pairs of the Panniculitis side effect, where the drug pairs with the side effect shown here are not in the current drug-drug interaction data [Tatonetti et al., 2012]. Our focus was on infrequent side effects, which were thought to be harder to predict. Also, we confirmed the biological validity of the predicted drug pairs by finding relevant biomedical articles by searching the biological literature using keywords of the predicted drug pair and the side effect.

Table 3.3 shows the result for the Panniculitis side effect (randomly selected from the top 5% infrequent side effects), which contains ten unknown pairs with the highest prediction scores by CentSmoothie. Also for each drug pair, the score obtained by HPNN (also Decagon) and the rank according to the score are shown if they were in the top 200 predictions. The last column showed the article relevant to each predicted drug pair. For 8 of the 10 predictions, we could find evidence (biomedical articles) by literature survey, implying the prominence of the findings by CentSmoothie. Comparing with the ranks (top ten) by CentSmoothie, those by HPNN were larger. Meanwhile, those ranks by Decagon were very large, where some ranks were out of the top 200, meaning that CentSmoothie and Decagon have different prediction preferences.



(a) (b)

Figure 3.4: Visualization of representations of drugs and side effects ((a-b) Panniculitis learned from HPNN and CentSmoothie trained with TWOSIDES.

Visualizing representations

Side effects and drug pairs

We visualized the representations of drugs and side effects learned by CentSmoothie and HPNN using TWOSIDES dataset to examine the difference between the central-smoothing assumption and the traditional smoothing assumption. We used the same four side effects as those we showed in Section 3.5.2.

Fig. 3.4 shows the visualization obtained by applying principal component analysis (PCA) to the resultant representation by each of the two methods, where for each side effect, drugs (blue dots) and the side effect (red triangle) are shown in the three-dimensional (3D) space. (For CentSmoothie, the representations on the subspace corresponding to the side effect were fed into PCA). We drew (gray) lines for drug pairs with side effects. For CentSmoothie, we further showed the midpoint of each drug pair (with a side effect) by a black dot, to see if the midpoint is close to the representation of the side effect. We could easily see that for each side effect, the representations of side effects tended to be located around the mean point among all midpoints (black dots). However, for HPNN, it was difficult to interpret the representations (of side effects) learned by HPNN among the representations of drugs. Also by using these visualizations, we could easily understand how each pair of drugs and the side effect

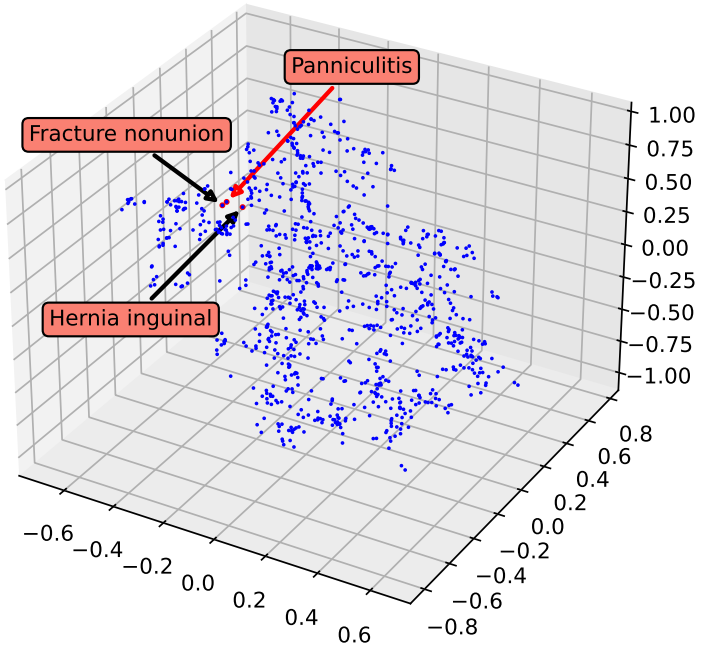


Figure 3.5: Visualization of side effect representations.

are positioned in the space. Particularly, by checking if the side effect is located nearby the midpoint of the corresponding drug pair, we can guess that the side effect might be caused.

Side effects relationships

We visualized the representations of all side effects learned by CentSmoothie on TWOSIDES dataset to see the relationships of side effects. Fig. 3.5 shows the visualization of side effects in a three-dimensional space. We could see that side effects are grouped into some small clusters. We highlighted an infrequent side effect: Panniculitis and two of its nearest neighbors: Fracture nonunion and Hemia inguinal. Furthermore, we could find evidence for the occurrence of Panniculitis with Fracture nonunion and Hemia inguinal [Ogden et al., 1960, Stieger et al., 2015].

3.6. Discussion

In this chapter, we presented CentSmoothie, a hypergraph neural network, for predicting drug-drug interactions, to learn representations of side effects together with drug representations in the same space. A unique feature of CentSmoothie is a new central-smoothing formulation, which can be incorporated into the hypergraph Laplacian, to model drug-drug interactions. Our extensive experiments using both synthetic and three real datasets confirmed clear performance advantages of CentSmoothie over existing hypergraph and graph neural network methods, indicating that CentSmoothie could learn representations of drugs and side effects simultaneously with the central-smoothing assumption. Furthermore, CentSmoothie kept high performance on the infrequent side effects for which the performances of other methods dropped significantly, indicating that CentSmoothie allows leveraging the relationships among side effects to help the difficult cases of less frequent side effects. For future work, it is interesting to extend the central-smoothing assumption into more general cases not limited to 3-uniform hypergraphs. In addition, learning adaptive ratios to replace the constraint of the midpoint might be considered.

Chapter 4

SPARSE: Learning multiple combinations of drug properties with sparsity control for improving prediction performances of drug-drug interactions

4.1. Introduction

In the previous chapter, we proposed a state-of-the-art generalization of a DDI graph can be a DDI hypergraph, which can capture higher-order relationships, where drugs and side effects are both nodes, and each hyperedge is a triple of a side effect with two interacting drugs. On the DDI hypergraph, we proposed a novel hypergraph neural network, namely CentSmoothie [Nguyen et al., 2022a], to learn the representations of drugs and side effects altogether. In DDIs, two drugs with totally different properties can still interact with each other, hence the traditional hypergraph neural networks using similarity assumption on node representations are not suitable [Feng et al., 2019]. Instead, CentSmoothie, assumes that each side effect is caused by a unique combination of latent features of the corresponding interacting drugs. However, in real life,

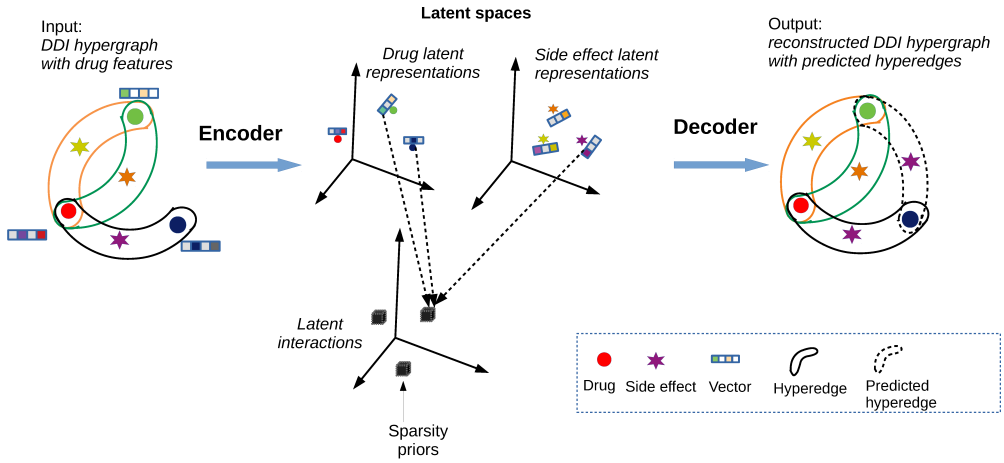


Figure 4.1: A schematic illustration of the procedure in the proposed model, SPARSE.

each side effect might have many different mechanisms [Suleyman et al., 2010] that cannot be reflected in a single combination of drug latent features. Hence it is necessary to learn different types of combinations of drug latent features for each side effect. This is the first problem **(P1)**, which we would like to address for further improvement.

To solve P1, we borrow one idea of stochastic block models (SBMs) on hypergraphs such that each node (e.g. drug or side effect) has one or several latent features [Anandkumar et al., 2013, Pal and Zhu, 2021] and there exist interactions (associations) of latent features. This method can learn different types of combinations of drug latent features for each side effect, at once. In addition, to improve the quality of learned latent features, input node features also can be used [Zhang et al., 2019b]. However, transformations from input node features and node relationships in the hypergraphs to latent features might be complex and, especially, non-linear. This is the second problem **(P2)**, which has not been addressed in existing SBMs and we address in this chapter.

Moreover, DDI data is sparse (for example, in the largest DDI dataset, 97.6% of all triples of drug-drug-side effects are not a DDI), suggesting that the model for learning DDIs also should be sparse. However, recent work on DDIs has not used this sparsity of the data [Nguyen et al., 2022a, Zitnik et al., 2018], which might potentially impair model performance. This is the third problem (**P3**),

which we address in this chapter.

We propose SPARSE, a new model for DDI prediction, to solve the above three problems. For P1, we assume that there exist drug and side effect latent features with latent interactions so that each side effect latent feature interacts with several pairs of drug latent features. For P2, we encode drug features and the DDI hypergraph altogether in the latent representations using a suitable hypergraph neural network. For P3, we guide the model to preserve the sparsity of the data using a suitable sparsity control. Fig. 4.1 schematically illustrates these ideas of our model. That is, the model consists of two parts: (i) an encoder and (ii) a decoder. The encoder encodes the input of the DDI hypergraph (for example, three hyperedges in Fig. 4.1) with drug features into latent spaces of drug and side effect latent representations, and interactions of latent features. The decoder reconstructs from the latent spaces the DDI hypergraph with new DDI predictions (for example, the dotted hyperedge in Fig. 4.1). Finally, a sparsity prior (horseshoe priors in our model) is used to control the sparsity of the latent interactions.

Our extensive experiments first validated the advantage of SPARSE in terms of prediction performance by using both synthetic and real-world datasets. Throughout all experiments on prediction performance, SPARSE achieved better prediction performances than competing methods, such as CentSmoothie and SBM. For example, in the experiment using the largest real DDI dataset, called TWOSIDES, SPARSE achieved AUC (area under the ROC curve) of 0.9524 and AUPR (area under the precision-recall curve) of 0.882, while CentSmoothie achieved AUC of 0.9348 and AUPR of 0.8749 and SBM achieved AUC of 0.9337 and AUPR of 0.8583. Similarly when using JADERDDI, another DDI dataset, SPARSE achieved AUC of 0.9698 and AUPR of 0.7348, while CentSmoothie was AUC of 0.9684 and AUPR of 0.6044 and SBM was AUC of 0.9428 and AUPR of 0.5963.

We then examined the top prediction obtained by SPARSE, which is trained by using the whole TWOSIDES. That is, we checked the number of overlaps between the top 400 predictions by one method and DDIs in drugs.com [Drugs.com, 2021, Thelwall et al., 2017], which is a commonly used online web checker for DDI. We found 98 DDIs in drugs.com out of the top 400 predictions, while by

using the same procedure, CentSmoothie found only 71 DDIs out of the top 400 predictions, implying that SPARSE can find new DDIs more than competing methods.

Finally, we validated the prediction results by characterizing the top predictions obtained by SPARSE. In more detail, we checked the biological properties, such as target proteins, of the top ten triples of drug-drug-side effect, predicted by SPARSE, by using latent features connected to these top ten predictions. We then found that top predictions can be associated with some biological mechanisms and particularly with responsible proteins/pathways. These results indicate that our model, SPARSE, can provide high predictive performances as well as latent biological knowledge beneficial to understand the background behind predicted DDIs.

4.2. Related Work

Recalling from Chapter 3, a DDI hypergraph is a generalization of DDI graphs to allow learning representations of drug and side effect nodes altogether in latent spaces instead of fixing side effects as one-hot labels [Nguyen et al., 2022a]. In detail, with DDI hypergraph representation, DDI is considered as high-order relationships of drug-drug-side effects in the form of a hypergraph where nodes are both drugs and side effects, and each hyperedge is a triple of two interacting drugs and a side effect caused by the drugs. There are two types of hypergraph neural networks models on the DDI: similarity based and non-similarity based. The similarity based models, for example, traditional spectral based hypergraph neural networks, assume that interacting drugs should have similar representations [Fan et al., 2021, Feng et al., 2019]. However, in DDI, two interacting drugs are not necessarily similar. For non-similarity models, the current state-of-the-art method is CentSmoothie [Nguyen et al., 2022a] which assumes that the representation of a side effect can be represented by a combination of latent features of two drugs causing the side effect. However, CentSmoothie cannot deal with multiple combinations of latent features at the same time.

In order to deal with multiple combinations of latent features, one possible

approach is to use the idea of stochastic block models (SBMs), which can be applied to hypergraphs, with each node belonging to several latent features (groups) and associations of latent features (groups) [Anandkumar et al., 2013]. However this has not been applied to DDI hypergraphs, and more importantly, SBM is based on linear assumption, while DDI can be generated through more complex relations to be represented by non-linearlity.

Many studies have shown the benefits of sparsity regularization, which is a commonly used method to achieve sparsity of models, especially on noisy and sparse data [Carvalho et al., 2009, Tibshirani, 1996]. In a Bayesian viewpoint, sparsity regularization can be understood as a result of using sparse prior distributions. A state-of-the-art method for sparsity regularization is to use horseshoe priors [Carvalho et al., 2009, Piironen and Vehtari, 2017]. It shows an advantage in comparison with traditional Laplace prior (Lasso regularization) [Tibshirani, 1996] in that the horseshoe prior allows to shrink in both directions: no shrinkage for important features and complete shrinkage for non-important (noise) features. A comparable shrinkage prior with the horseshoe prior is the spike and slab prior [Hoeting et al., 1999]. However, the spike-and-slab prior is a discrete prior that requires the Markov chain Monte Carlo sampling for optimization, which is not effective for large-scale datasets like DDI.

4.3. Materials and Methods

4.3.1 Background

We give definitions for horseshoe priors and n-mode tensor products for 3-dimensional tensors, which will be used later.

Horseshoe priors

We summarize the horseshoe prior [Carvalho et al., 2009], a state-of-the-art prior for sparsity control, for a non-negative 3-dimensional tensor: $\mathbf{B} = \{\mathbf{B}_{i,j,k}\} \in \mathbb{R}_{0+}^{K_1 \times K_2 \times K_3}$. The idea of the horseshoe prior is that each $\mathbf{B}_{i,j,k}$ follows a normal distribution with the same zero mean and a different variance. Each variance has two parts: one is a global parameter sharing among all variances

to decide the sparsity of $\mathbf{B}$ and one is a local parameter to decide the magnitude of each variance by using a heavy tail distribution with the Half-Cauchy distribution. In more detail:

$$\mathbf{B}_{i,j,k} \sim N(0, \tau^2 \Lambda^2_{i,j,k}) \tag{4.1}$$

$$\Lambda_{i,j,k} \sim C^+(0,1) \tag{4.2}$$

where τ is a global parameter for sparsity, and $C^+(0,1)$ is a Half-Cauchy distribution defined by: $p(\Lambda_{i,j,k}) = \frac{2}{\pi} \frac{1}{1+\Lambda^2_{i,j,k}}$ for $\Lambda_{i,j,k} \geq 0$.

Both the horseshoe prior and Laplace prior (for Lasso regularization) are shrinkage priors such that by using priors, values of features tend to be shrunk [Piironen and Vehtari, 2017]. Let $\hat{\mathbf{B}}_{i,j,k}$ be the optimal values without priors, then the optimal values having priors have the form: $\bar{\mathbf{B}}_{i,j,k} = (1 - \kappa_{i,j,k})\hat{\mathbf{B}}_{i,j,k}$, where $0 \leq \kappa_{i,j,k} \leq 1$ is a shrinkage factor depending on the priors. With Laplace prior (Lasso regularization), the density of $\kappa_{i,j,k}$ tends to be a constant near 1 and disappears near 0, meaning that it always shrinks all features, containing important ones. In contrast, the density of $\kappa_{i,j,k}$ with the horseshoe prior has two peaks at 0 and 1, meaning that the horseshoe prior allows two kinds of shrinkage: no shrinkage to maintain important features and complete shrinkage to remove unimportant features.

N-mode tensor product

The n-mode tensor product can be understood as a generalization of the matrix dot product in high-dimension that the product is processed at the n^{th} dimension. Considering in the 3-dimensional space with a tensor: $\mathbf{B} \in \mathbb{R}^{K_1 \times K_2 \times K_3}$ and a matrix $\mathbf{H} \in \mathbb{R}^{T \times K_n}, n \in \{1,2,3\}$, the n-mode product of $\mathbf{B}$ and $\mathbf{H}$ is denoted by $\mathbf{B} \times_n \mathbf{H}$ and is defined for each of $n = 1, 2$ and 3, as follows:

$$(\mathbf{B} \times_1 \mathbf{H})_{t,j,k} = \sum_{i=1}^{K_1} \mathbf{B}_{i,j,k} \mathbf{H}_{t,i} | t = 1...T, j = 1...K_2, k = 1...K_3 \tag{4.3}$$

$$(\mathbf{B} \times_2 \mathbf{H})_{i,t,k} = \sum_{j=1}^{K_2} \mathbf{B}_{i,j,k} \mathbf{H}_{t,j} | t = 1...T, i = 1...K_1, k = 1...K_3 \tag{4.4}$$

$$(\mathbf{B} \times_3 \mathbf{H})_{i,j,t} = \sum_{k=1}^{K_3} \mathbf{B}_{i,j,k} \mathbf{H}_{t,k} | t = 1...T, i = 1...K_1, j = 1...K_2 \tag{4.5}$$

4.3.2 Problem formulation

We recall the formulation of the DDI prediction problem as follows.

Input: Given a DDI hypergraph: $G = (V, E), V = V_D \cup V_S, E \subset V_D \times V_D \times V_S$, where V_D is a set of drug nodes, V_S is a set of side effect nodes[1]. The drug node features are $F_D \in \mathbb{R}_{0+}^{|V_D| \times K_0}$ and the side effect node features are one-hot vectors: $F_S \in \mathbb{R}_{0+}^{|V_S| \times |V_S|}$

Output: For $e = (u, v, t) \in V_D \times V_D \times V_S$, calculate a prediction score for interaction $m(e)$.

4.3.3 Proposed model

We propose SPARSE: a sparse model for learning multiple types of latent combinations of side effects and drugs to predict DDIs. Our model follows an auto-encoder framework with two parts: an encoder and a decoder. The encoder encodes the DDI hypergraph with drug node features to latent spaces with latent representations of drugs and side effects (**H**), and interactions of latent features (**B**). The decoder aims to reconstruct the DDI hypergraph with new predicted hyperedges from **H** and **B**. In the following parts, we first present our latent interaction assumption with sparsity for the interactions of drugs and side effects, and then we describe the encoder and decoder.

Latent interaction assumption

To model DDIs, we suppose that there exist latent spaces with drug latent features and side effect latent features where drug-drug interactions occur. The latent interaction assumption is that two interacting drugs cause a side effect if there exist a pair of drug latent features of the two drugs that interact with a latent feature of the side effect.

In detail, the formulation for the latent interaction assumption can be described as follows. Let $L_D = \{1, ..., K_D\}$ and $L_S = \{1, ..., K_S\}$ be the sets of indices of latent features of drugs and side effects with K_D and K_S be the numbers of latent features. Let $\mathbf{B} \in \mathbb{R}_{0+}^{K_D \times K_D \times K_S}$ be a 3-dimensional tensor representing

[1]Given $u, v \in V_D$, $t \in V_S$, two triples (u, v, t) and (v, u, t) are the same.

interactions of latent features of drugs and side effects. The set of interacting latent features is: $A = \{(i,j,k) \in L_D \times L_D \times L_S | \mathbf{B}_{i,j,k} > 0\}$.

Considering a triple of two drugs and one side effect $e = (u, v, t) \in V_D \times V_D \times V_S$. Let $\mathbf{h}^d(u), \mathbf{h}^d(v) \in \mathbb{R}_{0+}^{K_D}$, $\mathbf{h}^s(t) \in \mathbb{R}_{0+}^{K_S}$ be the vectors representing the presence of latent features of the two drugs and the side effect, respectively. Let $g_u = \{i \in L_D | \mathbf{h}^d(u)_i > 0\}$, $g_v = \{i \in L_D | \mathbf{h}^d(v)_i > 0\}$ and $g_t = \{i \in L_S | \mathbf{h}^s(t)_i > 0\}$ be the sets of latent features of u, v, and t, respectively.

Under the latent interaction assumption, u interacts with v to cause t if:

$$g_u \times g_v \times g_t \cap A \neq \emptyset \tag{4.6}$$

or with tensor product formulation:

$$\mathbf{B} \times_1 \mathbf{h}^d(u) \times_2 \mathbf{h}^d(v) \times_3 \mathbf{h}^s(t) > 0 \tag{4.7}$$

In practice, we can change the value 0 on the right side of Eq. (4.7) to a positive threshold. Eq. (4.6) will be used to generate synthetic data in the experimental section. Eq. (4.7) will be used in the decoder of the model.

Sparsity property

We first define formulations for sparsity measures of the DDI data and the latent interactions using the percentages of non-interactions. Let s_d be the sparsity of the hypergraph G:

$$s_d = 1 - \frac{2|E|}{|V_D|(|V_D| - 1)|V_S|} \tag{4.8}$$

The sparsity of the latent interactions s_l is defined as the percentage of the number of non-interacting triples of the latent features per the total number of all triples of the latent features.

$$s_l = 1 - \frac{2|A|}{|L_D|^2|L_S|} \tag{4.9}$$

DDI data is sparse as per statistics in Table 4.1. It is shown that 97.6% and 99.87% of all triples are non-interacting in TWOSIDES and JADERDDI, respectively.

The motivation for us to use sparse models is that sparse models, according to statistical learning theory, are usually more reliable models if they could fit

the training data well [Hastie, 2015]. As our sparse models have sparse interactions among latent features, we will prove that they tend to generate sparse data and are suitable for DDI data. We show a relationship between the sparsity of the models and the sparsity of data generated by the models, which are the ones that best fit the models, as follows.

Property 1: Assume that the DDI data is generated from the true generation model according to formula (4.7). Assuming that each drug and side effect has exactly n_u and n_t nonzero latent features, respectively. Then, there exists a relationship between the sparsity of the model and the expected sparsity of the generated data as follows:

$$\mathbb{E}(s_d) = 1 - (1 - s_l)\frac{n_u^2 n_t}{K_D^2 K_S} \tag{4.10}$$

Proof:

For a pair of drug u, v to cause side effect t, then $\mathbf{B} \times_1 \mathbf{h}^d(u) \times_2 \mathbf{h}^d(v) \times_3 \mathbf{h}^s(t) > 0$. This means that there is at least one nonzero entry of B corresponding to latent features of u, v, and t. Since there are exactly $n_u^2 n_t$ possible entries of B corresponding latent features of u, v and t, then the probability of a uniform sampling of entries of B to corresponding to these latent features is $p_1 = \frac{n_u^2 n_t}{K_D^2 K_S}$. This is the probability of having an interaction among the features (that generates a side effect data point).

Since entries of B are assumed to be randomly sampled according to a uniform distribution, the number of interactions when $\mathbf{B}$ have $|\mathbf{B}|_0 = (1 - s_l)K_D^2 K_S$ nonzero entries follows a binomial distribution $Binomial(|\mathbf{B}|_0, p_1)$.

With the assumption that the hypergraph is generated from this generative process, the expected number of nonzero data points (the number of hyperedges) becomes $|\mathbf{B}|_0.p_1 = (1 - s_l).n_u^2 n_t$. The expected sparsity of the hypergraph becomes $\mathbb{E}(s_d) = 1 - \frac{(1-s_l)n_u^2 n_t}{K_D^2 K_S} = 1 - (1 - s_l)p_1$.

This result leads to $\mathbb{E}(s_d) > s_l p_1$. It shows a relationship between the sparsity of the model (s_l) and the expected sparsity of the data generated by the model ($\mathbb{E}(s_d)$). It shows that the model can be sparse but cannot be as sparse as we want. It can be a hint on setting the sparsity of the model in learning processes.

Encoder

For the encoder, we use a hypergraph neural network with message passing [Yadati, 2020] to encode the input hypergraph and node features into latent spaces with node latent representations $\mathbf{H}$ and latent interactions $\mathbf{B}$ [2].

$$\mathbf{H} = (\mathbf{H^d}, \mathbf{H^s}) = g_{w_0}(G, F) \in \mathbb{R}_{0+}^{|V_D| \times K_D} \times \mathbb{R}_{0+}^{|V_S| \times K_S} \tag{4.11}$$

$$\mathbf{B} = f_{w_1}(G, F) \in \mathbb{R}_{0+}^{K_D \times K_D \times K_S} \tag{4.12}$$

where g_{w_0} and f_{w_1} are hypergraph neural networks based on message passing [Yadati, 2020] with parameters to learn w_0, w_1, $\mathbf{H^d} = \{\mathbf{h}^d(u) \in \mathbb{R}_{0+}^{K_D} | u \in V_D\}$ (node representations of drugs) and $\mathbf{H^s} = \{\mathbf{h}^s(t) \in \mathbb{R}_{0+}^{K_S} | t \in V_S\}$ (node representations of side effects). The formulation of each message passing layer has the following form:

$$\mathbf{h}^{(l+1)}(a) = \sigma\left(\mathcal{T}\left(\left\{M^{(l)}\left(a, \mathbf{h}^{(l)}(a), \left\{(b, \mathbf{h}^{(l)}(a))\right\}_{b \in e}\right)\right\}_{e \in N_a}\right)\right) \tag{4.13}$$

where $\mathbf{h}^{(l)}(a)$ is the representation of node $a \in V_D \cup V_S$ at layer (l), σ is an activation function, $\mathcal{T}$ is an aggregation function (for example, an average function), $N_a = \{e \in E | a \in e\}$ and $M^{(l)}$ is a message passing function at layer (l) to pass information from neighbor nodes in hyperedge e to a:

$$M^{(l)}\left(a, \mathbf{h}^{(l)}(a), \left\{(b, \mathbf{h}^{(l)}(b))\right\}_{b \in e}\right) = \tag{4.14}$$

$$\sum_{b \in e} \mathcal{M}^{(l)}(c(a), c(b), \mathbf{h}^{(l)}(a), \mathbf{h}^{(l)}(b)), \tag{4.15}$$

where $\mathcal{M}^{(l)}$ is a two layer feedforward neural network, $c(b) = 1$ if $b \in V_D$ and $c(b) = -1$ if $b \in V_S$ are the node types.

Decoder

The reconstruction of the hypergraph is from the latent interaction assumption. The likelihood to reconstruct each triple $e = (u, v, t) \in V_D \times V_D \times V_E$ follows a Gaussian distribution:

$$p(e | \mathbf{B}, \mathbf{H}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{i(e) - m_{w_0, w_1}(e))}{\sigma}\right)^2\right) \tag{4.16}$$

[2] For simplicity, $\mathbf{B}$ can be considered as a free parameter to learn.

where $i(e) = 1$ if $e \in E$, $i(e) = 0$ if $e \in \bar{E} = V_D \times V_D \times V_S / E$, and $m_{w_0,w_1}(e)$ is the mean value for the latent interaction of e:

$$m_{w_0,w_1}(e) = \mathbf{B} \times_1 \mathbf{h}^d(u) \times_2 \mathbf{h}^d(v) \times_3 \mathbf{h}^s(t) \tag{4.17}$$

Eq. (4.17) is also the score for the interactions of triples (u, v, t) used for prediction. The likelihood for the decoder is:

$$p(G|\mathbf{B}, \mathbf{H}) = \prod_{e=(u,v,t) \in V_D \times V_D \times V_E} p(e|\mathbf{B}, \mathbf{H}) \tag{4.18}$$

Objective function

The objective function of our method is to maximize the posterior of the model. The objective function consists of two parts: one for the log-likelihood of the model and one for the prior for sparsity control. Let $\mathbf{\Lambda} \in \mathbb{R}_{0+}^{K_D \times K_D \times K_S}$ be the horseshoe prior parameter for $\mathbf{B}$ and τ be the hyperparameter for the global sparsity of the horseshoe prior. We have the following objective function:

$$\underset{\mathbf{B},\mathbf{H},\mathbf{\Lambda} \geq 0}{\operatorname{argmax}} \underbrace{\log p(G|\mathbf{B}, \mathbf{H})}_{\text{log likelihood}} + \underbrace{\log p(\mathbf{B}|\mathbf{\Lambda}, \tau) + \log p(\mathbf{\Lambda})}_{\text{log of horseshoe prior}}, \tag{4.19}$$

where $\log p(G|\mathbf{B}, \mathbf{H})$ is the log likelihood of Eq. (4.18) with $\mathbf{H}$ in Eq. (4.11) and $\mathbf{B}$ in Eq. (4.12), and $\log p(\mathbf{B}|\mathbf{\Lambda}, \tau) + \log p(\mathbf{\Lambda})$ is the logarithm of the horseshoe prior:

$$\log p(\mathbf{B}|\mathbf{\Lambda}, \tau) = \sum \frac{-1}{2} \left(\frac{\mathbf{B}_{i,j,k}}{\tau \mathbf{\Lambda}_{i,j,k}}\right)^2 + \sum \log \mathbf{\Lambda}_{i,j,k}^{-1} + const \tag{4.20}$$

$$\log p(\mathbf{\Lambda}) = \sum \log \frac{1}{1 + \mathbf{\Lambda}_{i,k,j}^2} \tag{4.21}$$

We then use stochastic gradient descent libraries in the PyTorch framework for optimizing Eq. (4.19). An implementation of SPARSE is available at https://github.com/anhnda/SPARSE.

We also consider two other variants of SPARSE: SPARSE_O for not using any sparsity prior and SPARSE_L for using Laplace prior (Lasso regularization), to examine the effect of using the horseshoe prior.

4.4. Experimental results

We validated SPARSE in two scenarios: synthetic data and real data. On the synthetic data, assuming that the data is generated from the latent interactions, we examined if SPARSE can recover the latent interactions under changing hyperparameters of data: the number of latent features, sparsity, and amount of noise. On real data, we checked the prediction performance of SPARSE in comparison with state-of-the-art DDI prediction methods by using three real-world DDI datasets. Additionally, we evaluated if the top unknown predictions by SPARSE can be related to biological phenomena like functions and mechanisms.

For all experiments, we used 20-fold cross-validation by dividing hyperedges into 20 folds, keeping the same number of hyperedges (side effects) in each fold. We reported the mean and standard deviation of the two commonly used measures AUC (area under the ROC curve) and AUPR (area under the precision-recall curve). Also, all reported results were the highest performances through grid searches of hyperparameters. There were three hyperparameters for grid searches for SPARSE: 1) latent feature sizes. The tested values were 30, 40, 50, and 60. We set the same size for all layers. 2) global sparsity τ. The tested values were 0.01, 0.02, 0.03, 0.05, and 0.1, and 3) the numbers of neural layers. The tested values were 1, 2, and 3. The hyperparameter values obtained were 50 for the latent feature size, $\tau = 0.02$ for TWOSIDES, and $\tau = 0.01$ for CADDDI and JADERDDI, and the number of neural layers was 2. All experiments were run in a computer with Intel Core I7-9700 CPU, 8 GB GeForce RTX 2080 GPU, and 32 GB RAM.

4.4.1 Synthetic data

Data generation

The generation process for synthetic data consists of two steps: 1) generating latent interactions and 2) generating triples of interacting drug-drug-side effects from the latent interactions, as follows.

1. Generating latent interactions. Given sets of indices of drug latent fea-

tures: $L_D = \{1, 2, ..., K_D\}$ and side effect latent features: $L_S = \{1, 2, ..., K_S\}$.

(a) Initialize a set of latent interactions $A = \emptyset$.

(b) For each $k \in L_S$:

i. Sample the number of drug latent feature pairs: $n_k = RandomInteger(M)$, where M is the maximum number of pairs.

ii. Sample n_k pairs $(i, j) \in L_D \times L_D$. For each pair (i, j): $A = A \cup \{(i, j, k)\}$.

2. Generating drug interactions:

(a) Generate drug and side effect latent features. Assume that there are V_D drugs and V_S side effects.

i. For each drug $u \in V_D$:

i) Sample the number of drug latent features: $n_u = RandomInter(N_1)$, where N_1 is the maximum number of drug latent features.

ii) Sample $g_u \subset L_D, |g_u| = n_u$. For drug feature vectors F: $m_u \in \mathbb{R}_{0+}^{K_D \times c}, m_u \leftarrow 0, m_u[i] = 1$ if $\lfloor i/c \rfloor \in g_u, f_u = Gaussian(m_u, \delta)$. $F = F \cup f_u$.

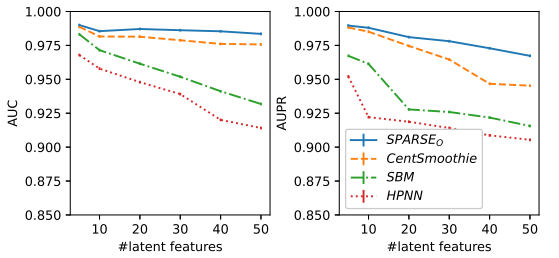
ii. For each side effect $t \in V_S$, sample the number of side effect latent feature $n_t = RandomInter(N_2)$ and Sample $g_t \subset L_S, |g_t| = n_t$.

(b) Generating true triples E^*. Initialize $E^* \rightarrow \emptyset$. For $(u, v, t) \in V_D \times V_D \times V_S$, if $g_u \times g_v \times g_t \cap A! = \emptyset$ then (u, v, t) is a true triple: $E^* = E^* \cup (u, v, t)$.

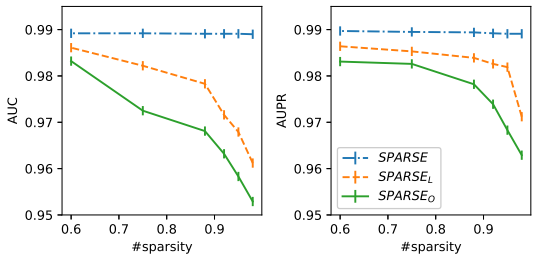
(c) Adding noise:

i. For each $e \in E^*$, replace e by a random sample $e' \in \bar{E^*} = V_D \times V_D \times V_S / E^*$ with probability r. The final set of triples of drug-drug-side effects is E.

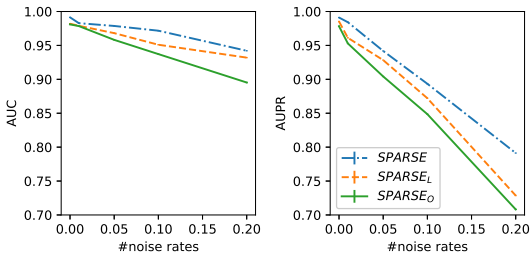
Finally, we have a synthetic data set with triples of drug-drug-side effects E and drug feature vectors F.



(a) Changing #latent features.



(b) Changing sparsity.



(c) Changing noise rates.

Figure 4.2: Performances on synthetic data, when changing (a) #latent features, (b) sparsity, and (c) amount of noise.

Experiments

The synthetic data has five hyperparameters: the number of drugs, the number of side effects, the number of latent interactions, data sparsity, and the amount of noise (noise rate). We evaluated our methods by changing one hyperparameter and fixing the other four. The hyperparameters changed are 1) numbers of latent features, 2) data sparsity, and 3) noise rate.

1) Changing the number of latent features

Setting: $V_D = 400, V_S = 300$, noise rate $r = 0.01$. We changed $K_D = K_S \in \{5, 10, 20, 30, 40, 50\}$. For each (K_D, K_S), we selected N_1, N_2 and M such that the sparsity of the generated data is kept at 0.98.

Compared methods: We compared four methods: SPARSE$_O$ (no sparsity control), CentSmoothie [Nguyen et al., 2022a], a similarity-based hypergraph neural network, HPNN [Feng et al., 2019] and stochastic block model on hypergraph (SBM) [Anandkumar et al., 2013].

Results: Fig. 4.2a shows the results, where SPARSE$_{\mathrm{O}}$ achieved the highest performances among the compared methods in all cases. We had the following two findings:

1) For the small number of latent features, the performance of CentSmoothie was close to SPARSE$_{\mathrm{O}}$ (both AUC and AUPR were around 0.99 under $K_D = K_S = 5$). However, by increasing the number of latent features, the performance gap between SPARSE$_{\mathrm{O}}$ and CentSmoothie also increased (gaps in AUC and AUPR were around 0.01 and 0.03, respectively, when $K_D = K_S = 50$). This result implies that CentSmoothie was unable to distinguish latent interactions clearly for a large number of latent interactions, while SPARSE$_{\mathrm{O}}$ worked better for capturing multiple latent interactions.

2) The performances of SBM were lower than both CentSmoothie and SPARSE$_{\mathrm{O}}$ since SBM did not use the node features, which decreased the performance. HPNN, a similarity based hypergraph neural network, had the lowest performance since the two drugs of a DDI do not necessarily have similarity in the data generated from latent interactions. Overall, these results indicated that SPARSE$_{\mathrm{O}}$ can recover the latent interactions better than the other methods.

2) Changing data sparsity

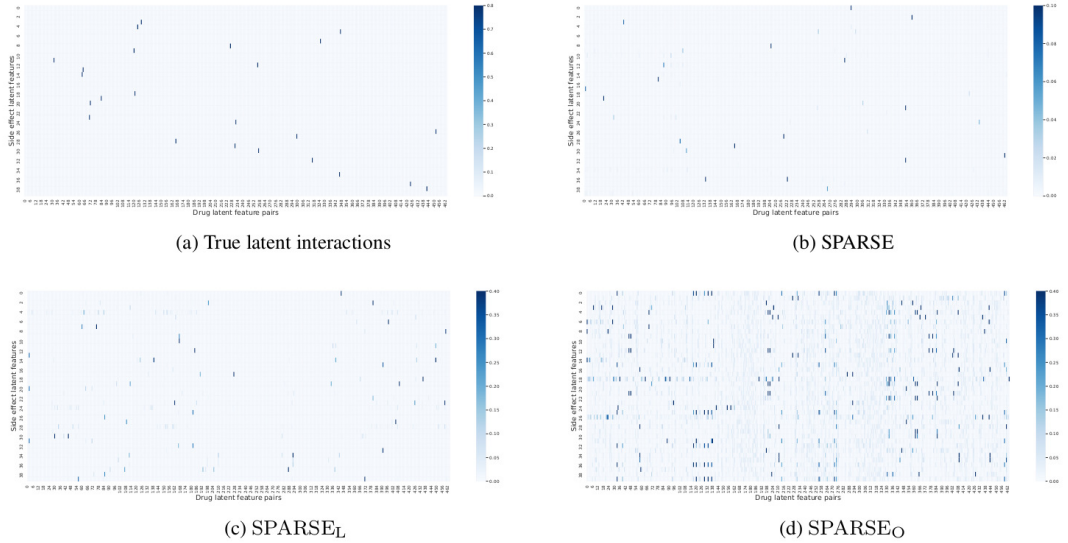
Setting: $V_D = 400, V_S = 300, K_D = K_S = 50, N_1 = N_2 = 4$ and $r = 0.01$. We changed M in $\{50, 40, 30, 20, 10, 5\}$, resulting in data sparsity in $\{0.6, 0.75, 0.88, 0.92, 0.95, 0.98\}$, respectively.

Compared methods: Since in the previous experiment, SPARSE$_{\mathrm{O}}$ outperformed the compared methods already, we compared SPARSE and two variants SPARSE$_{\mathrm{L}}$ and SPARSE$_{\mathrm{O}}$ (please see the end of Section 3.3.4) to check the effect of sparse priors.

Results: Fig. 4.2b shows the results, where SPARSE achieved the highest performance, followed by SPARSE$_{\mathrm{L}}$ and SPARSE$_{\mathrm{O}}$. In particular, the performance advantage by SPARSE using sparsity control was clearer with higher sparsity. These results indicate that the horseshoe prior is suitable for learning sparse data.

3) Changing the amount of noise

Setting: $V_D = 400, V_S = 300, K_D = K_S = 50, N_1 = N_2 = 4, M = 1$ (keeping



(a) True latent interactions (b) SPARSE

(c) SPARSE_{L} (d) SPARSE_{O}

Figure 4.3: Illustrations of learned latent interactions of SPARSE (and variants) on synthetic data.

the data sparsity of 0.98). We changed noise r in $[0, 0.01, 0.05, 0.10, 0.20]$.

Compared methods: We again compared SPARSE with two variants SPARSE_{L} and SPARSE_{O} to examine the effectiveness of the sparse priors to deal with noise.

Results: Fig. 4.2c shows the results, where again SPARSE achieved the highest performances among the three methods for all different amounts of noise. When there is no noise, the performances of the three methods were very close to each other. However, as the amount of noise is increased, the advantage of SPARSE over the other two methods became clearer. For example, when the amount of noise is 20%, the gap between SPARSE and SPARSE_{L} reached around 0.07, and the gap between SPARSE and SPARSE_{O} was around 0.1. These results suggest that the horseshoe prior could deal with noise better than the Laplace prior and the case with no sparsity prior.

Illustrations of learned latent interactions on synthetic data.

We visualized the learned latent interactions on the synthetic data in Fig. 1, where the x-axis is for pairs of drug latent features, the y-axis is for the side

effect latent feature, and the latent interactions are shown as dots. We used the data with a sparsity of 0.98.

Fig. 1 (a) is the true latent interactions that were used to generate sparse (only a few number of) DDIs. Fig. 1 (b), 1 (c), and 1 (d) are learned latent interactions of SPARSE, SPARSE$_L$, and SPARSE$_O$, respectively. We can see that the learned latent interactions of SPARSE were the closest ones to the true latent interactions (Fig. 1 (a)). On the other hand, the other methods captured non-significant interactions also. Hence, these results show that SPARSE with the horseshoe prior was suitable to deal with sparse data.

Sensitivity of SPARSE by changing the global sparsity hyperparameter τ

We examined the sensitivity of SPARSE by changing global sparsity hyperparameter τ in (10^{-10}, 10^{-5}, 0.001, 0.01, 0.02, 0.03, 0.05, 0.1, 0.5,1, 10, 100, 1000, 10^5 and 10^{10}). The results are in Fig. 4.4. The x-axis is $\log_{10}$ of τ, and the y-axis is AUPR.

We can see that SPARSE achieved the highest performance with $\log_{10}(\tau)$ of around -3 and 0 (τ from 0.001 to 1) and decreased as $\tau \to 0$ or $\tau \to \infty$. This is a reasonable, expected result, since as $\tau \to 0$, the horseshoe regularization term becomes stronger and as $\tau \to \infty$, the horseshoe regularization term becomes weaker (and eventually no regularization).

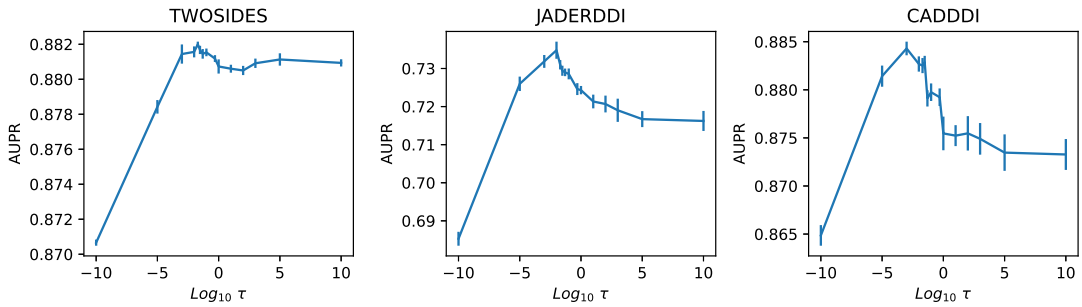


Figure 4.4: Sensitivity of SPARSE by changing the global sparsity hyperparameter τ.

Table 4.1: Statistics of three real datasets.

Dataset	#drugs	#side effects	#drug-drug pairs	#drug-drug-side effects (DDIs)	Avg. #side effects/ #drug-drug pairs	Sparsity
TWOSIDES	557	964	49,677	3,606,046	72.58	97.6%
CADDDI	587	969	21,918	373,976	17,06	99.77%
JADERDDI	545	922	36,929	222,081	6.01	99.83%

Table 4.2: Comparison of performances of the methods on the real DDI datasets.

Method	TWOSIDES		CADDDI		JADERDDI	
	AUC	AUPR	AUC	AUPR	AUC	AUPR
MRGNN	0.8452 ± 0.0036	0.8029 ± 0.0039	0.9226 ± 0.0015	0.7113 ± 0.0031	0.9049 ± 0.0009	0.3698 ± 0.0019
Decagon	0.8639 ± 0.0029	0.8094 ± 0.0024	0.9132 ± 0.0014	0.6338 ± 0.0029	0.9099 ± 0.0012	0.4710 ± 0.0027
SpecConv	0.8785 ± 0.0025	0.8256± 0.0022	0.8971 ± 0.0055	0.6640 ± 0.0014	0.8862 ± 0.0025	0.5162 ± 0.0047
HPNN	0.9044 ±0.0003	0.8410 ± 0.0007	0.9495 ± 0.0004	0.7020 ± 0.0018	0.9127 ± 0.0004	0.5198 ± 0.0016
SBM	0.9337 ± 0.0002	0.8583 ± 0.0004	0.9588 ± 0.0006	0.8170 ± 0.0008	0.9428 ± 0.0006	0.5963 ± 0.0018
CentSmoothie	0.9348 ± 0.0002	0.8749 ± 0.0013	0.9846 ± 0.0001	0.8230 ± 0.0019	0.9684 ± 0.0004	0.6044 ± 0.0025
$SPARSE_O$	0.9511 ± 0.0002	0.8811 ± 0.0001	0.9824 ± 0.0009	0.8773 ± 0.0014	0.9692 ± 0.0007	0.7230 ± 0.0008
$SPARSE_L$	0.9517 ± 0.0001	0.8815 ± 0.0002	**0.9859** ± 0.0007	0.8797 ± 0.0010	0.9694 ± 0.0011	0.7276 ± 0.0017
SPARSE	**0.9524** ± 0.0001	**0.8820** ± 0.0002	0.9837 ± 0.0010	**0.8843** ± 0.0012	**0.9698** ± 0.0008	**0.7348** ± 0.0018

4.4.2 Real data

Data description

We used three real-world datasets for DDI, namely TWOSIDES [Tatonetti et al., 2012], CADDDI, and JADERDDI. To our knowledge, TWOSIDES is the largest benchmark dataset for DDI. The other two datasets, i.e. CADDDI and JADERDDI, were generated from Canada Vigilance Adverse Reaction Reports and Japanese Adverse Drug Event Reports, respectively, in the same manner as the way that TWOSIDES was generated from the adverse events reported to U.S. Food and Drug Administration (FDA) [Nguyen et al., 2022a]. For all datasets, we only chose small molecular drugs, which can be found in DrugBank. Also, we focused on drugs appearing in more than five interactions (hyperedges) in each dataset. For each drug, we used a feature (binary) vector, with a size of 2,329, consisting of 881 substructures and 1,448 interacting proteins. Table 4.1 shows summary statistics of the three real benchmark datasets, TWOSIDES, CADDDI, and JADERDDI.

Table 4.3: Number of overlaps with DDIs in drugs.com for the top 400 predictions.

Method	#overlaps
SPARSE	98
CentSmoothie	71
HPNN	48

Predictive performance experiments

Compared methods: For our method, we used SPARSE and two variants SPARSE_{O} and SPARSE_{L}. We further used five methods as competing methods against SPARSE. These competing methods were CentSmoothie [Nguyen et al., 2022a], the traditional similarity-based hypergraph neural network (HPNN) [Feng et al., 2019], two DDI graph-based graph neural networks: Decagon [Zitnik et al., 2018] and SpecConv [Kipf and Welling, 2016], and, a molecular graph-based graph neural network, MRGNN [Xu et al., 2019]. Decagon and CentSmoothie provide available codes, and we ran them with the recommended settings. For MLNN, MGRNN, SpecConv, HPNN, and SBM, we implemented them and did a grid search for finding the best hyperparameter values.

Results – Cross-validation predictive performance: Table 4.2 shows AUC and AUPR results of all competing methods. From this table, SPARSE and two variants (SPARSE_{L} and SPARSE_{O}) achieved the highest performances, followed by CentSmoothie, SBM, and HPNN. On the other hand, the performances of SpecConv, Decagon, and MRGNN were significantly lower. Amazingly, SPARSE_{O} (SPARSE without any sparsity prior) achieved still better performance over CentSmoothie, particularly in AUPR. There was only one case (CADDDI), where the AUC of SPARSE was slightly smaller than that of CentSmoothie. We then ran a t-test over the prediction results of these two methods, to examine the significance of the difference between CentSmoothie and SPARSE. The resultant p-value of t-test was 0.057, indicating that the performance advantage of CentSmoothie over SPARSE was NOT significant, under the regular significance level of 0.05. Also, it has to be noted that AUPR is more useful than AUC for imbalanced data [Saito and Rehmsmeier, 2015], which can be often seen practically. We emphasize that DDI is a typical example of this situation. In

Table 4.4: Top 10 new (unknown) predictions with potentially associated latent features of proteins and extracted proteins.

No.	Drug A	Drug B	Side effect	Observable features associated with latent features	Extracted proteins of drugs from DrugBank	References
1	ciprofloxacin	mefenamic acid	abdominal distension	cytochrome enzymes	-	Venkataraman et al. [2014]
2	naratriptan	oxycodone	abnormal ECG	serotonin transporters and receptors	-	Baldo [2018], Ritter et al. [2019]
3	naratriptan	tramadol	abnormal ECG	serotonin transporters and receptors	-	Baldo [2018]
4	naratriptan	sertraline	abnormal ECG	serotonin transporters and receptors	5-hydroxytryptamine receptor 1B (and 1D) and sodium-dependent serotonin transporter	Ritter et al. [2019]
5	naratriptan	paroxetine	abnormal ECG	serotonin transporters and receptors	5-hydroxytryptamine receptor 1B (and 1D) and sodium-dependent serotonin transporter	Ritter et al. [2019]
6	trihexyphenidyl	thiothixene	abnormal EEG	dopamine receptors	-	Ritter et al. [2019]
7	carisoprodol	orphenadrine	abnormal vision	Not clear	-	Downs et al. [2019]
8	buspirone	orphenadrine	abnormal vision	Not clear	-	Ritter et al. [2019]
9	oxycodone	orphenadrine	abnormal vision	Not clear	-	Ritter et al. [2019]
10	carisoprodol	zaleplon	abnormal vision	Not clear	-	Fagiolini et al. [2004]

fact, the AUPR performance gap between SPARSE_{O} and CentSmoothie reached around 1%, 5% and 12% in TWOSIDES, CADDDI and JADERDDI, respectively. The performance gap in JADERDDI is especially sizable. This might be caused by the high sparsity of JADERDDI (see Table 4.1).

These results suggest that the latent interaction assumption in SPARSE is more reasonable and suitable for DDI prediction than CentSmoothie and the other competing methods. Among SPARSE, SPARSE_{L} and SPARSE_{O}, SPARSE achieved the highest performance. Note that the performance gap between SPARSE and SPARSE_{L} in AUPR became clearer for more sparse data: for example, only around 0.1 % for TWOSIDES, while the gap reached around 1% for CADDDI and JADERDDI. Hence, we can see that with more sparse data, the horseshoe prior had the advantage over Laplace prior and also the case with no sparsity prior.

Algorithm 1 Extracting potentially associated drug features

Input: Learned parameters $\mathbf{B} \in \mathbb{R}_{0+}^{K_D \times K_D \times K_S}$, $\mathbf{H}^d = \{\mathbf{h}^d(u)\} \in \mathbb{R}_{0+}^{|V_D| \times K_D}$, $\mathbf{H}^s = \{\mathbf{h}^s(u)\} \in \mathbb{R}_{0+}^{|V_S| \times K_S}$, drug features matrix $\mathbf{F}^d = \{\mathbf{f}^d(u)\} \in \mathbb{R}_{0+}^{|V_D| \times K_O}$, a predicted triple (u, v, t), hyperparameter T

Output: Associated drug features for the triple

//Extract drug features for each latent feature

for $k \in 1...K_D$ **do**

$a_k = \{j | \text{Correlation}(\mathbf{H}^d_{:,k}, \mathbf{F}^d_{:,j}) \text{in top T}\}$

end for

//Calculate non-zeros latent interactions. $\odot$ *is the pairwise dot product,* $\otimes$ *is the outer product.*

$ss = \mathbb{B} \odot (\mathbf{h}^d(u) \otimes \mathbf{h}^d(v) \otimes \mathbf{h}^s(t)$

$tt = \{(i, j, k) | ss_{i,j,k} > 0\}$

//Extract potentially associated drug features for the triple

$Re \leftarrow \emptyset$

for $(i, j, k) \in tt$ **do**

$Re \leftarrow Re \cup \{(\text{Non-zero features of } \mathbf{f}^d(u) \in a_i,$ Non-zero features of $\mathbf{f}^d(v) \in a_k)\}$

end for

Return Re

Results – Unknown DDI prediction performance: We evaluated the predictive ability of unknown DDIs. That is, we first trained a model by using the whole TWOSIDES data (the largest dataset), then predicted the scores of unknown triples (drug-drug-side effect), and finally sorted the predicted triples in the descending order of the scores.

We focused on the top 400 predictions of each method and checked the overlap with the DDIs stored in drugs.com [Drugs.com, 2021, Thelwall et al., 2017], a commonly used web checker for DDIs. Table 4.3 shows the number of overlaps between the DDIs in drugs.com and the top 400 predictions. SPARSE found 98 overlapped DDIs with drugs.com, this number being the highest and followed by CentSmoothie with 71 and HPNN with 48.

Case studies: interpretation of top 10 unknown predictions

SPARSE is an SBM with latent features for drugs, side effects, and interactions. In particular, the model has connections between latent drug features and latent interactions. Thus from the trained model, we can extract the drug features which are most associated with each drug latent feature and further extract the drug features most associated with each latent interaction through the corresponding latent drug feature. This means that we can retrieve drug features of a DDI if we can connect the DDI with the latent interactions. Algorithm 1 shows the pseudocode of this procedure (with T=20 in our cases). SPARSE is a sparse model, which allows only a limited number of latent interactions and eventually allows to extract only a limited number of drug features. This is a sizable advantage of SPARSE for understanding the biological / chemical background behind predicted DDIs.

For case studies, we extracted drug features (such as protein / pathway names) of the top unknown DDI predictions by using SPARSE, which was trained by the entire TWOSIDES. Table 4.4 shows the top 10 predictions (out of the 400 predictions in the experiment of the previous section) with the observable features associated with latent drug features (5th column from the right-hand side. In this column, "*Not clear*" means that to our current understanding of the potential DDI mechanisms, we could not explain the corresponding low-level (molecular level) background, although our algorithm could find associated drug features.), the target protein of the corresponding drug using DrugBank (6th column), and the corresponding reference to each DDI (7th column). The top predictions are likely to be similar to each other, since the similar triples are likely to have similar scores. In fact the top predictions in Table 4.4 have large overlaps, but from the table, we could find the following four points:

1) The 4th and 5th predictions show the cases, where SPARSE could specify target proteins precisely, confirming the high credibility of these predictions and more importantly, approving the high ability of SPARSE for detecting unknown DDIs.

2) The 1st, 2nd, 3rd, and 6th predictions show the cases, where SPARSE could identify possible interacting protein groups (4th column), not necessarily directly associated with the drugs, indicating that SPARSE allows suggesting

novel interactions as well as potential target proteins.

3) The validity of the 7th, 8th, 9th, and 10th predictions might be understood by high-level views, like the connection between vision and dizziness/sedation. This result implies that SPARSE can predict probable interactions, which however cannot be straightforwardly inferred from low-level data.

4) Entirely, we could find relevant references for all top 10 predictions [Baldo, 2018, Fagiolini et al., 2004, Rho et al., 1997, Venkataraman et al., 2014], giving plausibility of these predictions and at the same time an additional layer of evidence for the usefulness of SPARSE in practical settings. We discuss below a case for a potential biological mechanism extracted from SPARSE:
Naratriptan, Sertraline, and abnormal ECG: Sertraline belongs to the selective serotonin reuptake inhibitor class antidepressants. Members of this class inhibit the reuptake of the neurotransmitter serotonin into cells [Ritter et al., 2019]. Through this inhibition, sertraline increases serotonin levels outside of the cells and allows serotonin to remain longer at its site of action. Naratriptan is known to cause heart-related side effects through serotonin receptor agonism at serotonin type 1 receptors [Dodick et al., 2004, Ritter et al., 2019]. Therefore, the predicted side effect can be a direct consequence of sertraline increasing the level of endogenous serotonin and naratriptan acting at serotonin receptors in the heart, with the resulting changes visible in electrocardiogram recordings.

4.5. Discussion

In this chapter, we have proposed SPARSE to learn the latent representations of drugs, side effects, and interactions, through hypergraph neural networks. SPARSE addresses three important issues of state-of-the-art DDI prediction which have not been addressed by any other methods. Extensive empirical validation using both synthetic and real data showed that SPARSE outperformed all current, cutting-edge methods for DDI prediction, verifying the effectiveness of multiple types of latent interaction assumptions and the sparsity control setting of SPARSE.

Chapter 5

Concluding remarks and Future directions

5.1. Summary

We provided a systematic survey for data resources with corresponding tasks and methods for drug side effect studies and established novel models for predicting side effects of drug-drug interactions with high prediction performances.

The classifications for data resources in drug side effects were presented in Chapter 2. Basically, these data resources can be divided into clinical and non-clinical data. Clinical data contains important personal context information such as drug side effects, diseases, dosages of treatments, and demographic information. Non-clinical data contains more detailed information about drugs and biological systems with chemical, and physical properties of drugs, drug-protein interactions, and biological pathways. We also summarized the commonly used drug descriptors to represent drug properties from the data resources. In addition to traditional physical and chemical descriptors, biological descriptors of drugs were also used to better describe drug information.

There were three main tasks in drug side effect studies: creating drug side effect benchmark data, drug side effect prediction, and drug side effect mechanism analysis. We did comparisons on the existing methods for predicting the side effects of single drugs, the results showed the prominence of deep learning

models with high prediction performance.

A new deep learning model for predicting drug-drug interactions was established in Chapter 3. Existing work normally considered drug-drug interaction data in the form of a graph with only drug nodes but lacked the side effect relationship expression. We for the first time proposed a new representation of drug-drug interaction data in the form of a hypergraph with both drug and side effect nodes and each hyperedge is a triple of two interacting drugs with a corresponding side effect to leverage the side effect relationships. We then proposed a deep learning model to learn drugs and side effects altogether with an assumption that the side effect representation is close to the combination of the two corresponding drug properties, which is reflected by the midpoint. The assumption was formulated in the CentSmoothie framework that outperformed existing cutting-edge methods in terms of prediction accuracy.

We further developed a model namely SPARSE in Chapter 4 that could learn multiple combinations of drug properties while CentSmoothie can learn only one combination. In addition, SPARSE could handle sparsity data of drug-drug interactions by using a suitable sparsity control. The empirical experimental results showed that SPARSE outperformed CentSmoothie and achieved the highest prediction in comparison with all other methods. Moreover, SPARSE could extract relevant proteins for explaining the predicted drug-drug interactions, implying the prominent for supporting the safety of the drug development process.

5.2. Future directions

We describe some potential directions to improve our models.

Higher-order of drug interactions prediction

In our current work, we only consider the side effects of single drugs or drug-drug interactions. However, side effects can be caused by the interactions of more than two drugs. Generalizing SPARSE to address these high-order drug interactions is a possible future work. A remaining challenge is the lim-

itation of the known high-order drug interactions, which is to our knowledge, there is no available benchmark data for drug interactions with at least three drugs. The lack of data is a barrier to applying computational methods, especially deep learning models.

Personalized drug-drug interaction prediction

The actual outcomes of medications depend on each individual. The general prediction model might capture some common patterns of the populations, but when applied to each individual, the actual result might be different. Therefore, an important objective of medication is to personalize the treatments. Developing a personalized drug-drug interaction prediction is a challenge to overcome. By integrating personal data, for example, genotype and phenotype profiles, there are chances to develop such personalized prediction models. However, due to the high cost of collecting large-scale genetic data and the strict policy in using and publishing, such a direction still needs a lot of effort.

Bibliography

Canada Vigilance Program . Canada vigilance adverse reaction online database. https://www.canada.ca/en/health-canada/services/drugs-health-products/medeffect-canada/adverse-reaction-database.html, 2021. Online; accessed 25 May 2021.

Pharmaceutical and Medical Devices Agency . The japanese adverse drug event report. https://www.pmda.go.jp/safety/info-services/drugs/adr-info/suspected-adr/0003.html, 2021. Online; accessed 15 March 2021.

Alan Agresti. A survey of exact inference for contingency tables. *Statistical science*, 7(1):131–153, 1992.

Paola Alberti and G Cavaletti. Management of side effects in the personalized medicine era: chemotherapy-induced peripheral neuropathy. In *Pharmacogenomics in Drug Discovery and Development*, pages 301–322. Springer, 2014.

Anima Anandkumar, Rong Ge, Daniel Hsu, and Sham M Kakade. A tensor approach to learning mixed membership community models. arxiv. org, 2013.

Song Bai, Feihu Zhang, and Philip HS Torr. Hypergraph convolution and hypergraph attention. *arXiv preprint arXiv:1901.08150*, 2019.

Brian A Baldo. Opioid analgesic drugs and serotonin toxicity (syndrome): mechanisms, animal models, and links to clinical effects. *Archives of toxicology*, 92(8):2457–2473, 2018.

Juan M Banda, Lee Evans, Rami S Vanguri, Nicholas P Tatonetti, Patrick B Ryan, and Nigam H Shah. A curated and standardized adverse drug event resource to accelerate drug safety research. *Scientific data*, 3:160026, 2016.

Yujia Bao, Zhaobin Kuang, Peggy Peissig, et al. Hawkes process modeling of adverse drug reactions with longitudinal observational data. In *Machine learning for healthcare conference*, volume 2017, pages 177–190. Proceedings of Machine Learning Research, 2017.

François Belleau, Marc-Alexandre Nolin, Nicole Tourigny, Philippe Rigault, and Jean Morissette. Bio2rdf: towards a mashup to build bioinformatics knowledge systems. *Journal of biomedical informatics*, 41(5):706–716, 2008.

Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

Evan E Bolton, Sunghwan Kim, and Stephen H Bryant. Pubchem3d: conformer generation. *Journal of cheminformatics*, 3(1):4, 2011.

Mei-Chun Cai, Quan Xu, Yan-Jing Pan, Wen Pan, Nan Ji, Yin-Bo Li, Hai-Jing Jin, Ke Liu, and Zhi-Liang Ji. Adrecs: an ontology database for aiding standardization and hierarchical classification of adverse drug reaction terms. *Nucleic acids research*, 43(D1):D907–D913, 2014.

Aurel Cami, Alana Arnold, Shannon Manzi, and Ben Reis. Predicting adverse drug events using pharmacological network models. *Science translational medicine*, 3(114):114ra127–114ra127, 2011.

D.S Cao, N Xiao, Y.J Li, et al. Integrating multiple evidence sources to predict adverse drug reactions based on a systems pharmacology model. *CPT: pharmacometrics & systems pharmacology*, 4(9):498–506, 2015.

Carlos M Carvalho, Nicholas G Polson, and James G Scott. Handling sparsity via the horseshoe. In *Artificial Intelligence and Statistics*, pages 73–80. PMLR, 2009.

T-H Hubert Chan and Zhibin Liang. Generalizing the hypergraph laplacian via a diffusion process with mediators. *Theoretical Computer Science*, 806:416–428, 2020.

Andy W Chen. Predicting adverse drug reaction outcomes with machine learning. *International Journal Of Community Medicine And Public Health*, 5(3):901–904, 2018.

Xin Chen, Zhi Liang Ji, and Yu Zong Chen. Ttd: therapeutic target database. *Nucleic acids research*, 30(1):412–415, 2002.

Xiujie Chen, Xiangqiong Liu, Xiaodong Jia, Fujian Tan, Ruizhi Yang, Sheng Chen, Lei Liu, Yunfeng Wang, and Yuelong Chen. Network characteristic analysis of adr-related proteins and identification of adr-adr associations. *Scientific reports*, 3:1744, 2013.

Yun Gu Chen, Yin Ying Wang, and Xing Ming Zhao. A survey on computational approaches to predicting adverse drug reactions. *Current topics in medicinal chemistry*, 16(30):3629–3635, 2016.

Xu Chu, Yang Lin, Yasha Wang, Leye Wang, Jiangtao Wang, and Jingyue Gao. Mlrda: A multi-task semi-supervised learning framework for drug-drug interaction prediction. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4518–4524. AAAI Press, 2019.

Kathryn Corrie and Jonathan G Hardman. Mechanisms of drug interactions: pharmacodynamics and pharmacokinetics. *Anaesthesia & Intensive Care Medicine*, 12(4):156–159, 2011.

Patrizia Crivori, Gabriele Cruciani, Pierre Alain Carrupt, et al. Predicting blood- brain barrier permeation from three-dimensional molecular structure. *Journal of medicinal chemistry*, 43(11):2204–2216, 2000.

Gabriele Cruciani, Emanuele Carosati, Benoit De Boeck, Kantharaj Ethirajulu, Claire Mackie, Trevor Howe, and Riccardo Vianello. Metasite: understanding metabolism in human cytochromes from the perspective of the chemist. *Journal of medicinal chemistry*, 48(22):6970–6979, 2005.

Behrooz Davazdahemami and Dursun Delen. A chronological pharmacovigilance network analytics approach for predicting adverse drug events. *Journal of the American Medical Informatics Association*, 25(10):1311–1321, 2018.

Allan Peter Davis, Cynthia G Murphy, Cynthia A Saraceni-Richards, Michael C Rosenstein, Thomas C Wiegers, and Carolyn J Mattingly. Comparative toxicogenomics database: a knowledgebase and discovery tool for chemical–gene–disease networks. *Nucleic acids research*, 37(suppl_1):D786–D792, 2008.

Sanjoy Dey, Heng Luo, Achille Fokoue, Jianying Hu, and Ping Zhang. Predicting adverse drug reactions through interpretable deep learning framework. *BMC bioinformatics*, 19(21):1–13, 2018.

Giovanna Maria Dimitri and Pietro Lió. Drugclust: a machine learning approach for drugs side effects prediction. *Computational biology and chemistry*, 68:204–210, 2017.

David W Dodick, Vincent T Martin, Timothy Smith, and Stephen Silberstein. Cardiovascular tolerability and safety of triptans: a review of clinical data. *Headache: The Journal of Head and Face Pain*, 44:S20–S30, 2004.

Anthony M Downs, Xueliang Fan, Christine Donsante, HA Jinnah, and Ellen J Hess. Trihexyphenidyl rescues the deficit in dopamine neurotransmission in a mouse model of dyt1 dystonia. *Neurobiology of disease*, 125:115–122, 2019.

Drugs.com. Drug interactions checker. 2021. Online; accessed 25 Dec 2021.

Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. Reoptimization of mdl keys for use in drug discovery. *Journal of chemical information and computer sciences*, 42(6):1273–1280, 2002.

Ehsan Emadzadeh, Abeed Sarker, Azadeh Nikfarjam, and Graciela Gonzalez. Hybrid semantic analysis for mapping adverse drug reaction mentions in tweets to medical terminology. In *AMIA Annual Symposium Proceedings*, volume 2017, page 679. American Medical Informatics Association, PubMed Central, 2017.

Michela Fagiolini, Jean-Marc Fritschy, Karin Löw, Hanns Möhler, Uwe Rudolph, and Takao K Hensch. Specific gabaa circuits for visual cortical plasticity. *Science*, 303(5664):1681–1683, 2004.

Haoyi Fan, Fengbin Zhang, Yuxuan Wei, Zuoyong Li, Changqing Zou, Yue Gao, and Qionghai Dai. Heterogeneous hypergraph variational autoencoder for link prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3558–3565, 2019.

Yue-Hua Feng, Shao-Wu Zhang, and Jian-Yu Shi. Dpddi: A deep predictor for drug-drug interactions. *BMC bioinformatics*, 21(1):1–15, 2020.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

Peter J Goodford. A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. *Journal of medicinal chemistry*, 28(7):849–857, 1985.

Francesca Grisoni, Davide Ballabio, Roberto Todeschini, et al. Molecular descriptors for structure–activity applications: A hands-on approach. In *Computational Toxicology*, pages 3–53. Springer, 2018.

Stefan Günther, Michael Kuhn, Mathias Dunkel, Monica Campillos, Christian Senger, Evangelia Petsalaki, Jessica Ahmed, Eduardo Garcia Urdiales, Andreas Gewiess, Lars Juhl Jensen, et al. Supertarget and matador: resources for exploring drug-target relationships. *Nucleic acids research*, 36(suppl_1): D919–D922, 2007.

Lowell H Hall and Lemont B Kier. Electrotopological state indices for atom types: a novel combination of electronic, topological, and valence state information. *Journal of Chemical Information and Computer Sciences*, 35(6):1039–1045, 1995.

Trevor Hastie. *Statistical learning with sparsity : the lasso and generalizations*. Chapman and Hall/CRC monographs on statistics and applied probability ; 143. CRC Press, Boca Raton, FL, 2015. ISBN 9781498712163.

Tu Bao Ho, Ly Le, Dang T Thai, et al. Data-driven approach to detect and predict adverse drug reactions. *Current pharmaceutical design*, 22(23):3498–3526, 2016.

Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. Bayesian model averaging: a tutorial (with comments by m. clyde, david draper and ei george, and a rejoinder by the authors. *Statistical science*, 14(4): 382–417, 1999.

Brooke E Hoots, Likang Xu, Mbabazi Kariisa, et al. 2018 annual surveillance report of drug-related risks and outcomes–united states. *CDC National Center for Injury Prevention and Control*, 2018.

George Hripcsak, Jon D Duke, Nigam H Shah, Christian G Reich, Vojtech Huser, Martijn J Schuemie, Marc A Suchard, Rae Woong Park, Ian Chi Kei Wong, Peter R Rijnbeek, et al. Observational health data sciences and informatics (ohdsi): opportunities for observational researchers. *Studies in health technology and informatics*, 216:574, 2015.

GJ Huba, Joseph A Wingard, and Peter M Bentler. A comparison of two latent variable causal models for adolescent drug use. *Journal of Personality and Social Psychology*, 40(1):180, 1981.

Trung Huynh, Yulan He, Alistair Willis, et al. Adverse drug reaction classification with deep neural networks. In *Proceedings of COLING*. Coling, COLING, 2016.

Md Jamiul Jahid and Jianhua Ruan. An ensemble approach for drug side effect prediction. In *Bioinformatics and Biomedicine (BIBM), 2013 IEEE International Conference on*, volume 2013, pages 440–445. IEEE, IEEE, 2013.

Zhi Liang Ji, Lian Yi Han, Chun Wei Yap, Li Zhi Sun, Xin Chen, and Yu Zong Chen. Drug adverse reaction target database (dart). *Drug safety*, 26(10):685–690, 2003.

Yanping Jiang, Yizhou Li, Qifan Kuang, Ling Ye, Yiming Wu, Lijun Yang, and Menglong Li. Predicting putative adverse drug reaction related proteins based on network topological properties. *Analytical Methods*, 6(8):2692–2698, 2014.

Minoru Kanehisa and Susumu Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30, 2000.

TS Keshava Prasad, Renu Goel, Kumaran Kandasamy, Shivakumar Keerthikumar, Sameer Kumar, Suresh Mathivanan, Deepthi Telikicherla, Rajesh Raju, Beema Shafreen, Abhilash Venugopal, et al. Human protein reference database—2009 update. *Nucleic acids research*, 37(suppl_1):D767–D772, 2008.

Sunghwan Kim, Paul A Thiessen, Evan E Bolton, Jie Chen, Gang Fu, Asta Gindulyte, Lianyi Han, Jane He, Siqian He, Benjamin A Shoemaker, et al. Pubchem substance and compound databases. *Nucleic acids research*, 44(D1): D1202–D1213, 2015.

Sunghwan Kim, Paul A Thiessen, Evan E Bolton, Jie Chen, Gang Fu, Asta Gindulyte, Lianyi Han, Jane He, Siqian He, Benjamin A Shoemaker, et al. Pubchem substance and compound databases. *Nucleic acids research*, 44(D1): D1202–D1213, 2016.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Justin Klekota and Frederick P Roth. Chemical substructures that enrich for biological activity. *Bioinformatics*, 24(21):2518–2525, 2008.

Hugo Kubinyi. Comparative molecular field analysis (comfa). *The encyclopedia of computational chemistry*, 1:448–460, 1998.

Michael Kuhn, Monica Campillos, Ivica Letunic, Lars Juhl Jensen, and Peer Bork. A side effect resource to capture phenotypic effects of drugs. *Molecular systems biology*, 6(1):343, 2010.

Michael Kuhn, Ivica Letunic, Lars Juhl Jensen, et al. The sider database of drugs and side effects. *Nucleic acids research*, 44(D1):D1075–D1079, 2015.

Kathy Lee, Ashequl Qadir, Sadid A Hasan, Vivek Datla, Aaditya Prakash, Joey Liu, and Oladimeji Farri. Adverse drug event detection in tweets with semi-supervised convolutional neural networks. In *Proceedings of the 26th International Conference on World Wide Web*, pages 705–714. International World Wide Web Conferences Steering Committee, WWW, 2017.

Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

Jiao Lin, Qifan Kuang, Yizhou Li, and et al. Prediction of adverse drug reactions by a network based external link prediction method. *Analytical Methods*, 5 (21):6120–6127, 2013.

Christopher A Lipinski, Franco Lombardo, Beryl W Dominy, et al. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced drug delivery reviews*, 23 (1-3):3–25, 1997.

Mei Liu, Yonghui Wu, Yukun Chen, Jingchun Sun, Zhongming Zhao, Xue-wen Chen, Michael Edwin Matheny, and Hua Xu. Large-scale prediction of adverse drug reactions using chemical, biological, and phenotypic properties of drugs. *Journal of the American Medical Informatics Association*, 19(e1):e28–e35, 2012.

Tiqing Liu, Yuhmei Lin, Xin Wen, Robert N Jorissen, and Michael K Gilson. Bindingdb: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic acids research*, 35(suppl_1):D198–D201, 2006.

Lara Magro, Ugo Moretti, and Roberto Leone. Epidemiology and characteristics of adverse drug reactions caused by drug–drug interactions. *Expert opinion on drug safety*, 11(1):83–94, 2012.

R Mahadevan and CH Schilling. The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metabolic engineering*, 5(4): 264–276, 2003.

Ronald D Mann and Elizabeth B Andrews. *Pharmacovigilance*. John Wiley & Sons, 2007.

Jean-Louis Montastruc, Agnès Sommet, Haleh Bagheri, and Maryse Lapeyre-Mestre. Benefits and strengths of the disproportionality analysis for identification of adverse drug reactions in a pharmacovigilance database. *British journal of clinical pharmacology*, 72(6):905–908, 2011.

Emir Muñoz, Vít Nováček, and Pierre-Yves Vandenbussche. Using drug similarities for discovery of possible adverse reactions. In *AMIA Annual Symposium Proceedings*, volume 2016, page 924. American Medical Informatics Association, 2016.

Emir Muñoz, Vít Nováček, and Pierre-Yves Vandenbussche. Facilitating prediction of adverse drug reactions by using knowledge graphs and multi-label learning models. *Briefings in bioinformatics*, 20(1):190–202, 2017.

Duc Anh Nguyen, Canh Hao Nguyen, and Hiroshi Mamitsuka. A survey on adverse drug reaction studies: data, tasks and machine learning methods. *Briefings in bioinformatics*, 22(1):164–177, 2021.

Duc Anh Nguyen, Canh Hao Nguyen, and Hiroshi Mamitsuka. Centsmoothie: Central-smoothing hypergraph neural networks for predicting drug-drug interactions. *arXiv preprint arXiv:2112.07837*, 2022a.

Duc Anh Nguyen, Canh Hao Nguyen, Peter Petschner, and Hiroshi Mamitsuka. Sparse: a sparse hypergraph neural network for learning multiple types of latent combinations to accurately predict drug-drug interactions. *Bioinformatics*, 38(Supplement_1):i333–i341, 2022b.

Hao Canh Nguyen and Hiroshi Mamitsuka. Learning on hypergraphs with sparsity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

William W Ogden, Donald M Bradburn II, and James D Rives. Panniculitis of the mesentery. *Annals of surgery*, 151(5):659, 1960.

Soumik Pal and Yizhe Zhu. Community detection in the sparse hypergraph stochastic block model. *Random Structures & Algorithms*, 2021.

Edouard Pauwels, Véronique Stoven, and Yoshihiro Yamanishi. Predicting drug side-effect profiles: a chemical fragment-based approach. *BMC bioinformatics*, 12(1):169, 2011.

Juho Piironen and Aki Vehtari. Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, 11(2): 5018–5051, 2017.

Aleksandar Poleksic and Lei Xie. Predicting serious rare adverse reactions of novel chemicals. *Bioinformatics*, 1:8, 2018.

Naresh Poloju and Purushotham Muniganti. Adverse drug reaction detection using data mining approaches: A survey. *Internatinal journal of recent trends in engineering and research*, 2018, 2018.

Dilli Ram Poudel, Prakash Acharya, Sushil Ghimire, Rashmi Dhital, and Rajani Bharati. Burden of hospitalizations related to adverse drug events in the usa: a retrospective analysis from large inpatient database. *Pharmacoepidemiology and drug safety*, 26(6):635–641, 2017.

Hossein Rahmani, Gerhard Weiss, Oscar Méndez-Lucio, et al. Arwar: A network approach for predicting adverse drug reactions. *Computers in biology and medicine*, 68:101–108, 2016.

Jong M Rho, Sean D Donevan, and Michael A Rogawski. Barbiturate-like actions of the propanediol dicarbamates felbamate and meprobamate. *Journal of Pharmacology and Experimental Therapeutics*, 280(3):1383–1391, 1997.

Michael J Rieder. Mechanisms of unpredictable adverse drug reactions. *Drug Safety*, 11(3):196–212, 1994.

Johannes Ring and Knut Brockow. Adverse drug reactions: mechanisms and assessment. *European surgical research*, 34(1-2):170–175, 2002.

James Ritter, Rod J Flower, Graeme Henderson, Yoon Kong Loke, David J MacEwan, and Humphrey P Rang. Rang and dale's pharmacology. 2019.

Narjes Rohani and Changiz Eslahchi. Drug-drug interaction predicting by neural network using integrated similarity. *Scientific reports*, 9(1):1–11, 2019.

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

Itay Shaked, Matthew A Oberhardt, Nir Atias, Roded Sharan, and Eytan Ruppin. Metabolic network prediction of drug side effects. *Cell systems*, 2(3): 209–213, 2016.

Shawn E Simpson, David Madigan, Ivan Zorych, Martijn J Schuemie, Patrick B Ryan, and Marc A Suchard. Multiple self-controlled case series for large-scale longitudinal observational databases. *Biometrics*, 69(4):893–902, 2013.

Paul E Stang, Patrick B Ryan, Judith A Racoosin, J Marc Overhage, Abraham G Hartzema, Christian Reich, Emily Welebob, Thomas Scarnecchia, and Janet Woodcock. Advancing the science for active surveillance: rationale and design for the observational medical outcomes partnership. *Annals of internal medicine*, 153(9):600–606, 2010.

Christoph Steinbeck, Yongquan Han, Stefan Kuhn, Oliver Horlacher, Edgar Luttmann, and Egon Willighagen. The chemistry development kit (cdk): An open-source java library for chemo-and bioinformatics. *Journal of chemical information and computer sciences*, 43(2):493–500, 2003.

Marco Stieger, Jean-Paul Schmid, Nikhil Yawalkar, and Thomas Hunziker. Extracorporeal shock wave therapy for injection site panniculitis in multiple sclerosis patients. *Dermatology*, 230(1):82–86, 2015.

Halis Suleyman, Abdulmecit Albayrak, Mehmet Bilici, Elif Cadirci, and Zekai Halici. Different mechanisms in formation and prevention of indomethacin-induced gastric ulcers. *Inflammation*, 33(4):224–234, 2010.

Nicholas P Tatonetti, P Ye Patrick, Roxana Daneshjou, and Russ B Altman. Data-driven prediction of drug effects and interactions. *Science translational medicine*, 4(125):125ra31–125ra31, 2012.

Bernard Testa and Lemont B Kier. The concept of molecular structure in structure–activity relationship studies and drug design. *Medicinal research reviews*, 11(1):35–48, 1991.

Bernard Testa, Giulia Caron, Patrizia Crivori, Sébastien Rey, Marianne Reist, and Pierre Alain Carrupt. Lipophilicity and related molecular properties as determinants of pharmacokinetic behaviour. *CHIMIA International Journal for Chemistry*, 54(11):672–677, 2000.

Daylight. Daylight theory: Fingerprint, 2018. `http://www.daylight.com/dayhtml/doc/theory/theory.finger.html` (10 Sep 2019, date last accessed).

FDA. Questions and answers on fda's adverse event reporting system (faers), 2019. `https://www.fda.gov/drugs/surveillance/fda-adverse-event-reporting-system-faers`, 10 Sep 2019, date last accessed.

Openeye scientific software. OMEGA, 2018. `https://www.eyesopen.com/omega` (10 Nov 2018, date last accessed).

WHO. Definitions, 1972. `http://www.who.int/medicines/areas/quality_safety/safety_efficacy/trainingcourses/definitions.pdf` (10 Sep 2019, date last accessed).

WHO. Guidelines for atc classification and ddd assignment, 2019. `https://www.whocc.no/filearchive/publications/2019_guidelines_web.pdf`, (10 Sep 2019, date last accessed).

Mike Thelwall, Kayvan Kousha, and Mahshid Abdoli. Is medical research informing professional practice more highly cited? evidence from ahfs di essentials in drugs. com. *Scientometrics*, 112(1):509–527, 2017.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Roberto Todeschini and Viviana Consonni. *Handbook of molecular descriptors*, volume 11. John Wiley & Sons, 2008.

Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 613–622. IEEE, IEEE, 2006.

Han van de Waterbeemd and Manfred Kansy. Hydrogen-bonding capacity and brain penetration. *CHIMIA International Journal for Chemistry*, 46(7-8):299–303, 1992.

Harini Venkataraman, Michiel W Den Braver, Nico PE Vermeulen, and Jan NM Commandeur. Cytochrome p450-mediated bioactivation of mefenamic acid to quinoneimine intermediates and inactivation by human glutathione s-transferases. *Chemical research in toxicology*, 27(12):2071–2081, 2014.

Izhar Wallach, Navdeep Jaitly, and Ryan Lilien. A structure-based approach for mapping adverse drug reactions to the perturbation of underlying biological pathways. *PloS one*, 5(8):e12063, 2010.

Chi-Shiang Wang, Pei-Ju Lin, Ching-Lan Cheng, Shu-Hua Tai, Yea-Huei Kao Yang, and Jung-Hsien Chiang. Detecting potential adverse drug reactions using a deep neural network model. *Journal of medical Internet research*, 21(2): e11016, 2019.

Juan Wang, Zhi-xin Li, Cheng-xiang Qiu, Dong Wang, and Qing-hua Cui. The relationship between rational drug design and drug side effects. *Briefings in bioinformatics*, 13(3):377–382, 2011.

Xiujuan Wang, Bram Thijssen, and Haiyuan Yu. Target essentiality and centrality characterize drug side effects. *PLoS computational biology*, 9(7):e1003119, 2013.

AJ Weiss, A Elixhauser, J Bae, et al. Origin of adverse drug events in us hospitals, 2011. *HCUP Statistical Brief*, 158, 2013.

Camille Georges Wermuth. *The practice of medicinal chemistry*. Academic Press, 2011.

David S Wishart, Craig Knox, An Chi Guo, Dean Cheng, Savita Shrivastava, Dan Tzur, Bijaya Gautam, and Murtaza Hassanali. Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic acids research*, 36 (suppl_1):D901–D906, 2007.

David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, et al. Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic acids research*, 46(D1):D1074–D1082, 2018.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Cao Xiao, Ping Zhang, W Art Chaowalitwongse, Jianying Hu, and Fei Wang. Adverse drug reaction prediction with symbolic latent dirichlet allocation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI Press, 2017.

Nuo Xu, Pinghui Wang, Long Chen, Jing Tao, and Junzhou Zhao. Mr-gnn: Multi-resolution and dual graph neural network for predicting structured entity interactions. *arXiv preprint arXiv:1905.09558*, 2019.

Naganand Yadati. Neural message passing for multi-relational ordered and recursive hypergraphs. *Advances in Neural Information Processing Systems*, 33, 2020.

Yoshihiro Yamanishi, Edouard Pauwels, and Masaaki Kotera. Drug side-effect prediction based on the integration of chemical and biological spaces. *Journal of chemical information and modeling*, 52(12):3284–3292, 2012.

Fan Yang, Xiaohui Yu, and George Karypis. Signaling adverse drug reactions with novel feature-based similarity model. In *Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on*, pages 593–596. IEEE, IEEE, 2014.

Rodney C Young, Robert C Mitchell, Thomas H Brown, C Robin Ganellin, Robin Griffiths, Martin Jones, Kishore K Rana, David Saunders, and Ian R Smith. Development of a new physicochemical model for brain penetration and its application to the design of centrally acting h2 receptor histamine antagonists. *Journal of medicinal chemistry*, 31(3):656–671, 1988.

Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 793–803, 2019a.

Wen Zhang, Feng Liu, Longqiang Luo, et al. Predicting drug side effects by multi-label learning and ensemble learning. *BMC bioinformatics*, 16(1):365, 2015.

Wen Zhang, Yanlin Chen, Shikui Tu, Feng Liu, and Qianlong Qu. Drug side effect prediction through linear neighborhoods and multiple data source integration. In *2016 IEEE international conference on bioinformatics and biomedicine (BIBM)*, pages 427–434. IEEE, 2016.

Yun Zhang, Kehui Chen, Allan Sampson, Kai Hwang, and Beatriz Luna. Node features adjusted stochastic block model. *Journal of Computational and Graphical Statistics*, 28(2):362–373, 2019b.

Huiru Zheng, Haiying Wang, Hua Xu, Yonghui Wu, Zhongming Zhao, and Francisco Azuaje. Linking biochemical pathways and networks to adverse drug reactions. *IEEE transactions on nanobioscience*, 13(2):131–137, 2014.

Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, 2018.