Learning sparse representations for predicting drug side effects, disease genes and customer preferences

Diego A. Galeano

a dissertation
presented to the Department of Computer Science
of
Royal Holloway, University of London
in candidacy for the degree
of Doctor of Philosophy

Supervisor: Prof. Alberto Paccanaro

September 2019

Declaration of Authorship

I Diego A. Galeano G. hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

_____

Signature

_____

Date

Para Felipa, Ramona, Fátima Galeano y en memoria de Albino.
Muchas gracias por su apoyo.
Aguyje.

Acknowledgments

I AM FOREVER GRATEFUL to all the fellows that contributed to my ability to complete this dissertation. First, the content of this dissertation is the product of collaborative research. Thank you to my co-authors Shantao Li, Cheng Ye, Rubén Jiménez, Mark Gerstein, and Alberto Paccanaro. Second, my research would not have been possible without the supportive funding from the BECAL Scholarship from Paraguay. When I ask for help to visit Prof. Mark Gerstein's lab at Yale University for three months in 2017, I received the generous funding from the BECAL Travel Award, the Santander Travel Award and the Royal Holloway Travel Award; including a generous fee waiver covered by Prof Gerstein.

Four years ago, I would not have thought possible to embark in this long journey of pursuing a doctoral degree. It takes the convincing words of Alberto Paccanaro. Alberto has been an amazingly supportive and dedicate mentor throughout these years. He is always encouraging me to work on important problems with high impact in healthcare and society, which has shaped the way I think about research. I always value the great freedom that he has given me to pursue my research interests and ideas. Soon after I join the lab in 2015, I mentioned to Alberto that I want it to work on the field of system pharmacology, and he provided great guidance as I started on this new field.

I would like to thank all the folks from PaccanaroLab that has made this journey certainty more joyful. Special thanks to Mateo, Juan, Ruben, Horacio, Victor, Jessica, Miki, Santiago, Philip, and our dear friends Cristina and Ed.

My family has been immense support, even from the other side of the world. Thank you to my mother Felipa and my sisters Ramona and Fátima, and my larger family in Paraguay. Special thanks to my aunt Brinda, thanks for all your help.

I would like to also thank Prof. Michael M. Bronstein, pioneer of geometric deep learning, that invited me to collaborate on their drug repositioning project. Also, thanks to Prof. Martin Wilkins, Vice Dean of the Faculty of Medicine at Imperial College London, who suggested that we should look at the predictions of the severe side effects that led to drug withdrawal from the market. Lastly, thanks to Prof. Sir Munir Pirmohamed for the conversation on his insights into these predictions.

Abstract

Computational prediction methods that operate on pairs of objects are fundamental tools for understanding and modelling complex systems in biology, chemistry, and customer preference in recommender systems. I present four sparse matrix completion models to learn a sparse representation of objects from data consisting of associations between pairs of objects. The main goal of my models is to be able to generalise, that is, to predict new relationships between a pair of objects. This thesis addresses the following problems: (1) drug-side effect frequency prediction; (2) drug-side effect prediction; (3) disease-gene prediction; and (4) user preference prediction in top-*N* recommender systems. I show how my sparse matrix completion models can be effectively used to predict missing relationships in the data; better than other state-of-the-art methods. My models are designed to favour interpretability. On the task of predicting the frequencies of drug side effects, I show a new algorithm for non-negative matrix factorisation that learns parts of the human anatomical system. On the task of predicting the presence/absence of drug side effects, I show a new algorithm that learns sparse self-representation of objects such that a given object, e.g. a side effect is represented by the linear combination of few other objects. In addition, my models naturally integrate structure knowledge in the form of graph networks, adding strong relational inductive biases without requiring well-defined heuristics or hand-crafted features.

Contents

List of Tables

Listing of figures

Is perception of the whole based on perception of its parts? There is psychological and physiological evidence for parts-based representations in the brain, and certain computational theories of object recognition rely on such representations. But little is known about how brains or computers might learn the parts of objects.

Daniel Lee and Sebastian Seung, Nature, 1999

0

Sparse Completion thinking in Matrix Completion

In many application areas in biology, chemistry and medicine, relationships between pair of objects are by nature experimentally determined. For instance, whether a

given drug is associated to a specific side effect is determined in randomised controlled trials in humans, or whether a given mutation in a gene is associated to a suspected genetic disorder is determined using genome-wide association studies (GWAS). In the past decade, several computational prediction methods have been developed to operate on pairs of objects by considering features of each[1]. Prominent examples are protein-protein interaction (PPI)[2], protein-drug interaction[3,4], protein-RNA interaction[5], drug-side effect[6] and drug-indication[7] prediction methods. Typically, rich patterns emerge when these objects are arranged in an incomplete matrix $X \in \mathbb{R}^{n \times m}$, where the row elements represent one set of objects and the column elements represent the other set of objects and an entry x_{ij} represents the measured relationship between the object i and the object j. This representation of the data appears in a wide variety of disciplines and this thesis presents new methods that have been developed to address the following problems:

- *Predicting the frequencies of drug side effects.* In terms of X, a row element is a drug and a column element is a side effect. A measured association x_{ij} between a drug i and a side effect j correspond to a frequency class — a natural number in the set $x_{ij} \in \{1, 2, 3, 4, 5\}$ — encoding the experimentally obtained frequencies of side effects in randomised controlled trials (very rare $= 1$, rare $= 2$, infrequent $= 3$, frequent $= 4$ and very frequent $= 5$). The remaining associations in X are filled in with zeros. A $x_{ij} = 0$ means that either drug i does not cause side effect j, or that it does, but it could not be detected. The problem is to predict the frequencies of missing drug side effect associations. To the best of my knowledge, this is the first attempt to predict frequencies of drug side effects in the literature. Previous approaches have focused on predicting the presence/absence of a drug side effect association, but not

its frequency.

- *Predicting the presence/absence of drug side effects.* In terms of X, a row element is a drug and a column element is a side effect. An association x_{ij} between a drug i and a side effect j correspond to a binary value — $x_{ij} \in \{0, 1\}$ — encoding whether a drug has been associated to a side effect. The problem is to predict missing drug side effect associations. This problem differs from the previous problem in that its main goal is to detect unknown side effects associated to a drug; regardless of its frequency. In some cases, when a lethal side effect is detected, new clinical studies needs to be designed to measure the frequency of the side effect in the relevant clinical cohort.

- *Predicting disease-gene associations.* In terms of X, a row element is a gene and a column element is a genetic disorder. An association x_{ij} between a gene i and a disease j correspond to a binary value — $x_{ij} \in \{0, 1\}$ — encoding whether a gene has been associated to a disease. The problem is then to predict genes associated with diseases.

- *Predicting user preference in recommendation systems.* In terms of X, a row element is a user and a column element is an item, e.g. a movie. An association x_{ij} between an user i and an item j correspond to a binary value — $x_{ij} \in \{0, 1\}$ — encoding whether an user has a preference towards an item. The problem is then to predict items likely to be preferred by the user.

The description of these problems illustrates how a matrix, consisting of row and column elements and associations between them, is a natural mathematical description for bipartite relationships between objects. Two common characteristics of the above problems are that: (i) only a small number of entries in X is observed and that; (ii) missing entries in X

are represented with zero values. The problem of "completing" a partially observed matrix have been extensively studied in the literature of matrix completion[8,9,10,11,12,13,14,15]. However, a standard matrix completion model does not represent missing values with zero values but rather performs the learning of the model based on the observed entries only. This typically means taking as input a sparse matrix X and returning as the output of the model a dense matrix $\hat{X} \in \mathbb{R}^{n \times m}$ such that $\hat{x}_{ij} = x_{ij}$ for $(i,j) \in \Omega$, where Ω is the set of indices of the observed entries. This is how the famous Netflix competition problem was framed in 2006[16], where the goal was to predict user ratings on films with rating values in the set $x_{ij} \in \{1,2,3,4,5\}$. However, in the problems that I addressed in this thesis, a "non-association" or the lack of association between two objects is a possible outcome, which I had represented with $x_{ij} = 0$. For example, a drug might not cause a side effect, a disease might not be associated with a gene and a user might never watch a certain film. I refer to this idea as the *sparse completion* assumption. Its immediate consequence is that even the ideal complete matrix $\hat{X}$ should remain sparse, with a possible large number of zero values. This thesis provides new computational prediction models that I called *sparse matrix completion* operating under the sparse completion assumption.

At the heart of standard matrix completion models there are two core ideas. Firstly, there is the assumption that datapoints (row or column elements in X) lie in a lower dimensionality $k \ll \min(n,m)$. Secondly, there is the idea that domain-specific knowledge can be used as a complementary information to improve the prediction.

In this context, consider that chemically similar drugs tend to have similar side effect profiles[17], or that like-minded users tend to like the same films[18] and that genetic disorders with similar (patho)phenotypes tend to be associated to the same genes in the DNA[19]. These are

examples illustrating how datapoints can be already intrinsically related when building X. These intrinsic relationships often lead to linear dependencies between datapoints and thus is natural to assume a *low-rank model*[*]. These models, which are at the core of matrix completion, typically assign a low-dimensional feature vector to each row element and a low-dimensional feature vector to each column element such that the measured relationship between a row and a column element is modelled by the dot-product of the two feature vectors. In mathematical terms, this can be formulated as the following matrix decomposition model: $\hat{X} = WH$, where $W \in \mathbb{R}^{n \times k}$ (each row is a feature vector) and $H \in \mathbb{R}^{k \times m}$ (each column is a feature vector). The rank of $\hat{X}$ is k — the number of features assigned to each row or column element.

Low-rank models can capture well the structure of datapoints that lie in a single low-dimensional subspace but they are not effective at capturing the structure of datapoints that lie in the union of low-dimensional subspaces. This latter has been addressed in recent years using self-expressive models[8]. A self-expressive model aims to represent each datapoint as a linear combination of few other datapoints. It aims to learn a sparse zero-diagonal matrix of coefficients $C \in \mathbb{R}^{m \times m}$ such that $\hat{X} = XC$ with the constraint that $\text{diag}(C) = 0$. This model, which often results in a high/full rank matrix, it is known as *high-rank model*. Although low- and high-rank matrix completion model are different in their mathematical formulation, their goal is the same, which is to better capture the intrinsic structure of datapoints to predict missing associations between row and column elements in X. In this thesis, I develop new models using both low- and high-rank matrix completion models.

[*]The rank of a matrix X is the number of linearly independent columns in the matrix.

The distribution of entries for each row and column element in X is typically not uniformly distributed. In fact, it has been noticed that the distribution of ratings in the Netflix dataset follows a long-tailed distribution with about 80% of the entries involving only 10% of the movies[20]. If we imagine a bipartite graph where row elements in X are one set of nodes and column elements in X are another set of nodes and an entry x_{ij} is the weight that is assigned to the edge connecting node i with node j in the network, then the long-tailed distribution observed in the Netflix dataset will resemble the preferential attachment principle of scale-free networks[21]. Rich nodes in the network, e.g. Star Wars, will always get richer by gaining links from newcomers. The main implication of this uneven distribution of the entries in X is that, for a large number of row and column elements, there is little information in the matrix to obtain a good representation for the prediction. As a matter of fact, in many cases, there is no information at all about a row or a column element of interest — this is the case of an isolated node in our bipartite graph. For instance, we might aim to predict the side effects of a novel compound that have not undergone clinical trials in human, or we might aim to recommend to users a recently released movie. In the recommendation system literature, this scenario is typically known as the *cold-start* problem[22].

Addressing the cold-start problem often requires domain-specific knowledge to establish relationships between datapoints that cannot be inferred otherwise. For instance, a gene associated with a molecularly uncharacterised disease can be predicted by exploiting the similarities in (patho)phenotypes between diseases and the connectivity of protein interaction networks. This complementary information, which is often presented in the form of graph networks, can incorporate relevant relational prior[23] between datapoints to either learn a better low-dimensional representation in a low-rank model, or a better self-representation

in a high-rank model. In this thesis, I showcase how my prediction models can integrate heterogeneous biological networks that can handle the cold-start scenario. The importance of integrating graph networks — instead of hand-crafted features or well-defined heuristics — in my models is inspired by the recent trend on representation learning on graphs[24] and geometric deep learning[25,26].

It has been recently reported that black-box machine learning models are being used to make high-stake decisions in society, including in the domains of healthcare and criminal justice[27,28]. Understandably, clinicians, patients and regulators would like to understand, for instance, how a binary classifier concluded that a drug causes a certain lethal side effect, e.g. stroke. In this thesis, I have developed sparse matrix completion models that are inherently interpretable. To favour interpretability, I imposed non-negative constraints on the learned sparse representations. In a low-rank model, $W, H \geq 0$, while in a high-rank model, $C \geq 0$. This was mainly motivated by the seminal work of Lee and Seung[29] on how non-negative constraints on matrices learns a parts-based representation of objects.

1 Contributions

My approach to tackling each problem was *problem-centred*. That is, I studied the problem, analysed the data — distribution, patterns, etc. — and then I formulated a hypothesis about the problem that I considered reasonable within its context. The reader may certainly find distinct interpretations of very similar models in different chapters but under the light of a different context. Therefore, each chapter in this thesis is self-contained and divided by problem. I summarised my contributions on each problem in Table 1.

Problem addressed	Existing computational approaches	Contributions
Predicting frequencies of drug side effects	n/a	General framework, objective function and multiplicative algorithm. Pharmacological interpretation of the model's representations and its links to drug routes of administration [30]. Chapter 1
Drug side effect prediction	Network-based [6,31], Label propagation and random walks [32,33], low-rank matrix decomposition [34,35].	Regularised low-rank matrix decomposition model [36]. Geometric high-rank sparse matrix completion model: objective function, algorithm, and pharmacological interpretation of the model. Prediction explainability [37]. Chapter 2
Disease gene prediction	DIAMonD [38], Prodige [39], PRINCE [40], Cardigan [41].	Graph regularised high-rank sparse matrix completion: objective function, algorithm. Prediction on molecularly characterised and uncharacterised diseases. Chapter 3
Top-N recommender systems	Neighbourhood models [42], low-rank matrix factorization [18], Sparse Linear Model [43].	Multiplicative algorithms for sparse linear methods: global optima convergence, covariance-driven regularisation, model interpretability in terms of popularity and novelty. Chapter 4

Table 1: Contributions of this thesis. Columns represent the problem addressed in each chapter, the existing computational approaches to address those problems, and my contributions to each problem, respectively. n/a stands for not available.

1.1 Chapter 1 - Predicting the frequencies of drug side effects

In this chapter, I focused on the problem of predicting the frequencies of drug side effects. Drug side effects are a leading cause of morbidity and mortality in health care, with an annual cost in the billions of dollars [44,45]. A wide range of computational approaches has been proposed to detect side effects of drugs in both pre-market [46,47,48,49,50,51,52,53,54,55,56] and post-

market[57,58,59] stages. However, we still lack computational approaches to predict the frequencies of drug side effects. The frequency of a drug side effect can be critical for the drug risk-benefit assessment and inaccurate estimations of side effect frequencies represent a potential risk of drug withdrawal from the market. For instance, in 2004, the arthritis drug Vioxx produced by the pharmaceutical Merk & Co. was withdrawn from the worldwide market because new data from clinical trials found an increased risk of heart attacks and stroke. This, and many other such cases[60], could have been avoided with accurate methods that predicts the frequency of specific drug adverse events.

Currently, in the pharmaceutical industry, the frequencies of drug side effects are measured experimentally in randomised controlled trials, but it is well recognised that these trials have numerous limitations and might fail to identify important drug side effects. In this chapter, I present the first purely computational method that can successfully predict the frequencies of drug side effects. Earlier computational approaches[6,32] focused on the problem of predicting the presence/absence of drug side effects, but could not predict its frequency. The computational framework I present here further strengthens the collection of tools available to drug safety professionals taking high-stakes decisions regarding the risk-benefit of any existing drug.

My contributions to this problem are the following:

- I framed the problem as that of simultaneously predicting frequency classes and the presence/absence of the associations. From the machine learning standpoint, my model aims to learn regression on the frequency classes $x_{ij} \in \{1, 2, 3, 4, 5\}$, and to correctly classify `true` from `false` drug side effect associations, where a `true` class is any $x_{ij} > 0$ and a `false` class is a $x_{ij} = 0$. This is important because our model

generalises over the classic drug side effect prediction[6], in which the goal is to classify `true` from `false` associations. In my model, however, the binary classification is as important as the correct estimation of the frequency class of the drug side effect.

- I proposed a non-negative matrix decomposition model (low-rank model) and developed a new objective function that considers different levels of uncertainty in the entries of X. The assumption is that the zero entries in the matrix X have higher uncertainty than measured associations. Earlier matrix decomposition methods (e.g. SVD or NMF), do not explicitly account for different levels of uncertainties in the data.

- I developed a novel multiplicative learning algorithm to solve the objective function. My algorithm does not require to set a learning rate nor applying projective functions and I proved that it satisfies the Karush-Khun-Tucker (KKT) complementary conditions of convergence.

- I showed that my method can predict the frequencies of drug side effects in a cross-validation setting. Through further experiments, I found that my method was able to predict that post-marketing side effects — those detected after the drug has been marketed — are very rare (1 in 10,000 cases) in the population. This is in fact a common belief in clinical medicine[61]. For selected case studies, I also show that my model was able to predict the frequency of specific lethal side effects that caused drug withdrawal from the market.

- I studied the pharmacological interpretability of the learned representations – that I called signatures – in the model. I show that the similarity between drug signatures

predict clinical and molecular drug activity. Importantly, I interpreted the specific components of the signatures in the model, and found that specific components are related to distinct anatomical activities of drugs and specific drug routes of administration.

- I analysed the reproducibility of the signatures. Given that we studied the pharmacological interpretation of the signatures, it is important to analysed whether these signatures are reproducible in multiple runs of the algorithm (non-convex optimisation problem). We found that the majority of the components of the signatures (8/10) are highly reproducible.

A pre-print detailing my model, algorithm and biological interpretation can be found in bioRxiv[62]. I then extended the biological interpretation of the model to include an in-depth pharmacological analysis of each component of the signatures with the help of Dr Shantao Li and Prof. Mark Gerstein from the Department of Molecular Biophysics and Biochemistry at Yale University (presented in section 3.5). Our paper is currently under review at *Nature Communications*.

1.2 Chapter 2 - Drug side effect prediction

In this chapter, I focused on the problem of predicting the "presence/absence" of drug side effects. This is typically framed as a binary classification problem for which a wide range of computational methods has been proposed (see reviews in[63,64]). Addressing the binary problem itself is important because the frequency information is only available for about half of the drugs in SIDER (only 40% of pairs have frequency information in SIDER 4.1[65]).

Furthermore, the binary problem also allows the use of other reported or post-marketing side effects for building prediction models.

To predict drug side effects, I proposed two different models. The first model is a regularised low-rank model. The second model is a high-rank model that also integrates multiple heterogeneous information about drugs in the form of graph networks. The integration of the geometric graph structure as a regularisation in the high-rank model was inspired by the recent trend of deep learning on graphs [25,26,24].

My contributions to this problem are the following:

- As a first attempt to predict drug side effects, I proposed a regularised low-rank matrix decomposition model. I showed that this low-rank model outperforms four state-of-the-art methods for drug side effect prediction.

- My second attempt was an interpretable high-rank matrix completion model on graphs that integrates heterogeneous graph networks for drugs and side effects. This began with the observation that the drug side effect matrix has, in fact, a high-rank. I called this model Geometric Sparse Matrix Completion (GSMC). To my knowledge, this is the first high-rank matrix completion model to predict drug side effects.

- To solve GSMC, I proposed a novel objective function and developed a new multiplicative learning algorithm. I proved that my learning algorithm convergences to a globally minimum solution with a first-order convergence rate. This theoretical guarantees of convergence is desirable for biological interpretation because it guarantees that if the same high-rank model is refit to the same data, but with changes in the initial random values of the weights, there will be the same learned self-representations.

- Extensive experiments on human clinical trials data show that my GSMC model outperforms six state-of-the-art methods in drug side effect prediction, including my low-rank model.

- I also studied the biological interpretability of my GSMC model. I show that the learned self-representations are informative of the biology underlying drug activity: these make explicit the similarities between drug activities at the molecular and phenotypic level. The learned self-representation matrices can be used for predicting the shared drug clinical activity, targets of drugs, and even the anatomical/physiological relationships between side effect phenotypes.

My work on low-rank model was presented in the 2018 International Joint Conference on Neural Networks (IJCNN) in Rio de Janeiro, Brazil. This paper is indexed by IEEE Xplore[36]. My high-rank model can be found in bioRxiv[37] and it is currently under preparation for submission at *Bioinformatics*.

Furthermore, I have used my high-rank model for drug repositioning in a join work with Prof. Michael Bronstein's group at Imperial College London and USI Lugano. This latter work was accepted at the NeurIPS 2019 Workshop on Graph Representation Learning — and arxiv version can be found in[66].

1.3 Chapter 3 - Disease gene prediction

In this chapter, I focused on the problem of disease gene prediction. The elucidation of genes associated with genetic disorders is critical for our understanding of the molecular mechanisms of diseases and the development of effective therapies[67]. Yet, gene prioritisation remains a challenge: about 56% of the diseases in the Online Mendelian Inheritance in

Man (OMIM) database have a single associated gene, and for about 40% of the diseases, the molecular basis is completely unknown (molecularly uncharacterised).

My contributions to this problem are the following:

- I proposed a high-rank matrix completion model for disease-gene prediction. This began with the observation that the gene-disease matrix has a high-rank. My model has two main features, interpretability, and the ability to generate predictions for diseases with unknown molecular basis.

- My method outperforms state-of-the-art methods such as PRINCE[40], DIAMOnD[38] and the Prodige family[39] at predicting the genes for molecularly characterised and uncharacterised disorders. In the more challenging case of molecularly uncharacterised diseases, we can retrieve around 50% of the genes associated with genetic disorders in the top-100 predictions.

- I showed that my model is inherently interpretable as it learns direct associations between genes and diseases based on an aggregated guilt-by-association principle.

- To provide a more realistic scenario, I validated some of the top predictions by using a prospective evaluation approach on prediction case studies. This realistic scenario preserves the chronological order in which information becomes available. That is, having trained my model with biological data available up to the year 2017, I checked whether top prediction could be found in a most recent 2018 version of OMIM.

This work was a joint work with Dr Cheng Ye and Prof. Alberto Paccanaro. Cheng helped me to run the experiments and analysed the results. This work is currently under preparation for submission at *Scientific Reports*.

1.4 Chapter 4 - Top-N recommender systems

In this chapter, I focused on the problem of predicting user preference in top-*N* recommender systems. Accurate recommendations of products to users is critical for e-commerce and entertainment platforms such as Netflix[18]. A challenge of these platforms is that only a small number of N-items are shown to the users. In the recommender system literature, this problem is typically framed as a Top-N recommendation system[20].

My contributions to this problem are the following:

- I proposed an algorithmic framework for Top-*N* recommender systems based on high-rank matrix completion under self-expressive models. This began with the observation that several real-world datasets used in recommendation systems have a high-rank structure — including popular datasets such as Netflix. I connected my formulations to a group of models called Sparse Linear Method (SLIM)[43] and I showed that my approach can be easily extended to this family of models.

- I provided a strong theoretical foundation regarding the objective function and the optimality of the solution using novel multiplicative learning algorithms. I showed that my objective function is smooth and that my learning algorithm converges to a unique global optimum solution.

- I have tested the performance of my algorithms across several real-world datasets and found state-of-the-art performance.

- In large-scale applications, fine-tuning model parameters can be prohibited due to time and space complexity. I empirically show that my algorithms do not require

fine-tuning of the model parameters and that they can even be used as a parameter-free model without great loss in performance. This property of my algorithms can be explained by its intrinsic regularisation.

- Importantly, the recommendations produced by my models are explainable and my model is inherently interpretable. I found that the learned self-representations in my model favours novelty in the recommendations while mitigating the bias of items popularity.

Several of the algorithmic ideas that are presented in chapter 4 were foundational to the algorithms presented in Chapters 2 and 3.

This work was a joint work with Ruben Jimenez and Prof. Alberto Paccanaro. Ruben helped me to run the benchmarks on the different datasets and to set up the cluster for these experiments. This work is currently under preparation for submission at the *IEEE Transactions on Knowledge and Data Engineering*.

1.5 Additional resources

Datasets and code are publicly available for reproducibility.

- Frequency prediction of drug side effects.

  Project website: `https://paccanarolab.org/drug-signatures/`.

  GitHub repository: `https://github.com/paccanarolab/SEFrequency`.

- Geometric Sparse Matrix Completion Model for predicting drug side effects.

  GitHub repository: `https://github.com/paccanarolab/GSMC`

- High-rank matrix completion for disease gene prediction.

  Project website: https://paccanarolab.org/hrmc-gene/.

  GitHub repository: https://github.com/paccanarolab/HRMC.

- High-rank matrix completion for Top-*N* recommender systems.

  GitHub repository: https://github.com/paccanarolab/NSEM

I have not failed. I've just found 10,000 ways that won't work.

Thomas A. Edison (1847-1931)

1

Predicting the Frequencies of Drug Side Effects

Drug risk-benefit assessment[68,69] requires the experimental measurement of the frequencies of drug side effects. Currently, these frequencies are estimated using intervention

and placebo groups during randomised controlled trials. Although these trials are inherently limited by the sample size, time-frame, and lack of accrual[70], they are the standard approach to eliminate selection bias in clinical medicine[71,72]. However, it is well recognised that numerous side effects are not observed during clinical trials[73], but are identified after the drug has reached the market[74,75,76]. For this reason, drug side effects remain a leading cause of morbidity and mortality in healthcare, with an annual loss of billions of dollars[77,45,44]. Several computational approaches have been proposed for predicting side effects of a given drug[6,32,78,31,79,80,36]. Yet, the application of these methods in drug risk-benefit assessment is limited, as they can only predict the presence or absence of a drug side effect, not its frequency.

Accurate estimation of the frequencies of side effects is vital to patient care in the clinical practice, but it is also essential for pharmaceutical companies as it reduces the risk of drug withdrawal from the market[81,82], or of the costly reassessment of side effect frequencies through new clinical trials[83].

In this chapter, I present a novel approach for predicting the frequencies of drug side effects. Given a few experimentally determined side effects, my method predicts the frequencies of a broader range of unknown side effects. To the best of my knowledge, this is the first computational method that successfully addresses the problem of predicting the frequencies of drug side effects. A critical application of my approach is in the early phase of clinical trials, where computational predictions can be used as complementary hypotheses to set the direction of the risk assessment in later phases of clinical trials, or after a drug has entered the market. My method can also be useful in other aspects of clinical trial design, such as in the estimation of the cohort size required for the detection of the side effect.

My approach for predicting the frequencies of drug side effects is to use a matrix decomposition algorithm that learns a small set of latent features (or signatures) that encode the biological interplay between drugs and side effects. My model is inspired by movie recommendation systems[18,84,85] that recommend movies to users: my recommendation system recommends side effects to drugs. Importantly, I constraint my matrix decomposition model to be non-negative; it has the advantage of making explicit the parts-based representation[86] thus offering biological interpretability. In other words, drugs are characterised by a set of learned non-negative features that, when additively combined, account for the side effect frequencies across the entire repertoire of drugs. Consequently, my predictions are explainable and the individual features can be interpreted in terms of specific human anatomical systems, and I show that they are related to different routes of administration and are predictive of shared drug clinical activity, drug targets and anatomy/physiology of side effect phenotypes.

1 The matrix decomposition model

In drug clinical trials, it is common to use five side effect frequency classes to describe the occurrence of drug side effects in clinical cohorts[87]. By coding these classes with integers between 1 and 5 — very rare $=1$, rare $=2$, infrequent $=3$, frequent $=4$, and very frequent $=5$ — I assembled a $n \times m$ matrix X with $n = 759$ drugs and $m = 994$ unique side effects containing 37,441 frequency class associations obtained from SIDER 4.1[65] (see Appendix 1 for details). The remaining entries of the matrix were filled with zeros.

The average frequency value in X is 3.52, indicating that frequencies from clinical trials are biased towards frequent side effects –– this has been attributed to the limitation of clin-

ical trials at detecting side effects of rare occurrence[61]. Popular side effects, such as headache, account for most of the non-zero entries in X, indicating that specific popular side effects are reported on most drugs[87]. Indeed, my analysis of X showed that drug side effects follow a long-tailed distribution, where about 30% of the side effects are responsible for 80% of the associations (Fig. 1.1a). Figure 1.1b shows that the distribution of frequency classes in X is zero-inflated: about 95% of the associations are unobserved.

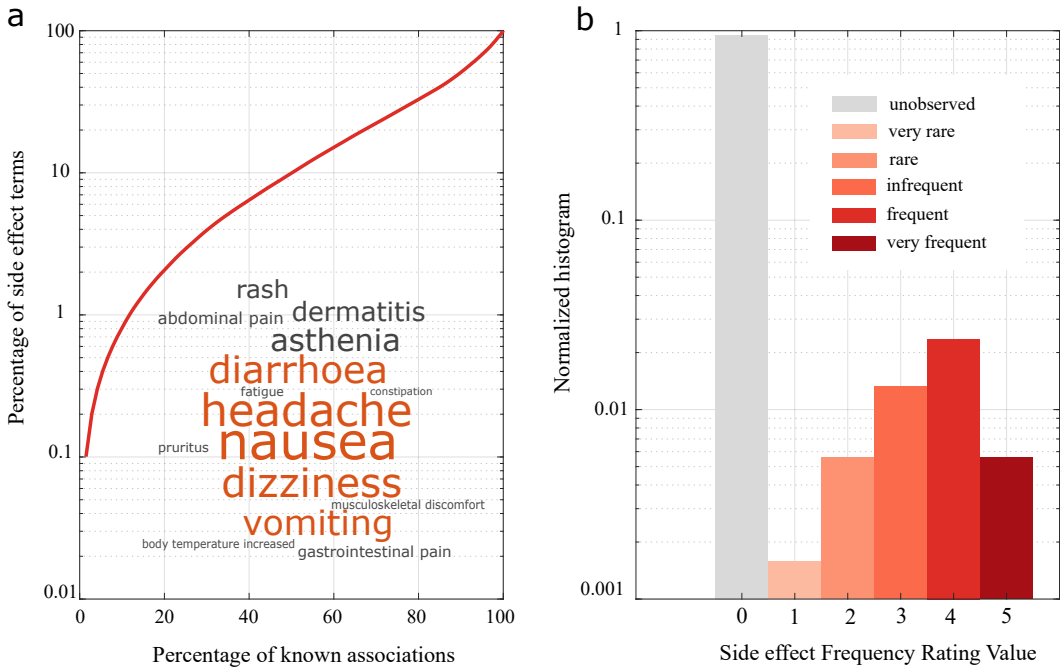

Figure 1.1: Distribution of drug side effects in our dataset. (a) Long-tailed distribution of side effects. Side effects in y-axis are ordered in decreasing order of popularity, i.e. the number of drugs in which a side effect appear. Inset. Word cloud of the fifteen most popular side effects. The size of the word is proportional to its popularity; the five most popular ones are coloured in orange (b) Histogram of side effect frequency classes. The frequency of a drug side effect in the population can be very rare (less than 1 in 10,000), rare (1 in 10,000 to 1 in 1,000), infrequent (1 in 1,000 to 1 in 100), frequent (1 in 100 to 1 in 10) or very frequent (greater than 1 in 10) – shown in shaded red bars. The remaining of the associations are unobserved (grey bar).

The long-tailed distribution of side effects resembles the distribution of the ratings previously found in movie datasets such as Netflix or Movielens[20]. Similar to our dataset, in

Movielens, about 30% most popular movies account for 80% of the ratings, and the distribution of ratings tend to be biased towards high values (Fig. 1.2). One widely studied group of methods for movie recommendation systems is based on low-rank matrix decomposition techniques[42]. Their fundamental assumption is that both users and movies can be represented as latent feature vectors in a low-dimensional space and that a rating value for a specific user-movie pair is obtained by the dot product of the corresponding feature vectors. The assumption is reasonable for movie datasets, where latent features can be thought of as modelling both movie genres and user preferences (e.g. thriller, romantic, sci-fi).

I realised that this assumption is also reasonable for our task: drugs and side effects can be represented as latent feature vectors in a low-dimensional space where the latent features might capture specific molecular or cellular mechanisms that elicit side effects[88]. Therefore, my idea is to learn a low-dimensional latent representation for each drug — that we shall call drug signature, $w \in \mathbb{R}^k$ — and a low-dimensional representation for each side effect — side effect signature, $h \in \mathbb{R}^k$ — such that the frequency of a drug-side effect pair is obtained by the dot product of the two feature vectors. This amounts to decomposing X into a product of two matrices as $X \simeq WH$, where $W \in \mathbb{R}^{n \times k}$ (each row is a drug signature), $H \in \mathbb{R}^{k \times m}$ (each column is a side effect signature) and $k \ll \min(n, m)$ is the number of latent features in the model. My matrix decomposition algorithm learns the matrices W and H by minimising the following loss function:

$$\min_{W,H} \mathcal{L}(W, H) = \frac{1}{2} \sum_{\Omega} (X_{ij} - (WH)_{ij})^2 + \frac{\alpha}{2} \sum_{\Omega^c} (WH)_{ij}^2 \tag{1.1}$$

subject to the non-negative constraints $W, H \geq 0$.

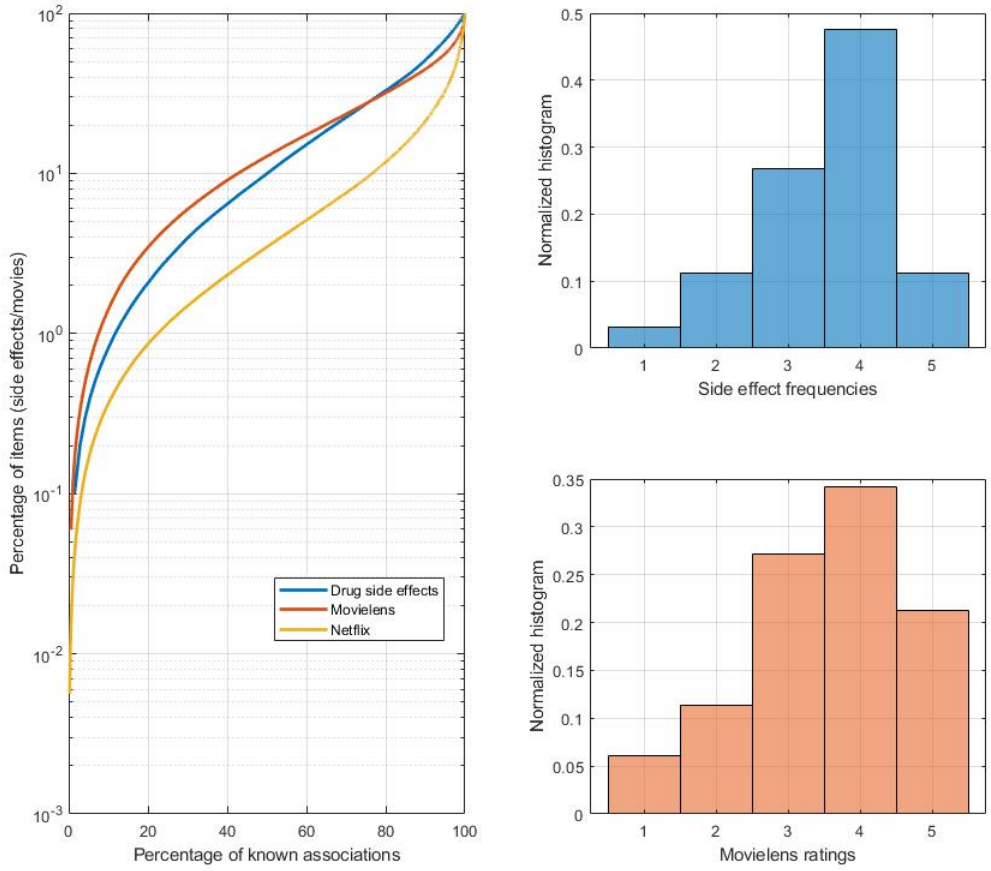

Figure 1.2: The heavy-tailed distribution of drug side effects and two popular movie datasets. *(Left)* Side effects and movies (items) are ordered according to popularity, most popular at the bottom. Side effects and movie datasets tend to have a few popular items containing more than 20% of the associations (usually known as short-head). However, most items reside in the long-tail of the distribution, populated with items (side effects or movies) with fewer associations. The Movielens dataset contains 943 users and 1682 movies with 100K associations (6.3% density). The Netflix dataset contains 480,189 users and 17,770 movies with 100M associations (1.17% density). The density of the Movielens dataset is more comparable to our dataset of drug side effects (4.96% density). *(Right)* Distribution of rating values for drug side effect frequency and the rating values in the Movielens dataset. The distribution of frequency values comes from a normal distribution (Chi-square goodness-of-fit Significance,$p < 2.23 \times 10^{-308}$) and it is very similar to the distribution of ratings in Movielens (Kolmogorov-Smirnov Significance, $p < 2.51 \times 10^{-233}$).

The first summation in my model is the *fitting constraint on the observed entries* where $\Omega = \{(i,j) \,|\, X_{ij} \in \mathcal{F}\}$ with $\mathcal{F} \in \{1,2,3,4,5\}$, which aims at reconstructing X for

the known frequency classes. The second term in Eq. 1.1 is the *fitting constraint on the zeros* where $\Omega^c = \{(i,j) \,|\, X_{ij} \in 0\}$, which aims at reconstructing the zeros found in X, and I introduced it here because our dataset is fundamentally different from the movie ratings. While in the movie rating matrix a zero entry is simply a missing value that needs to be filled in, for our problem, a zero entry indicates that a specific side effect was not detected for a given drug — which could either mean that the drug does not cause the side effect, or that it does, but it could not be detected. The parameter $\alpha \in [0,1]$ controls the relative importance of the zeros; in other words, it represents our confidence in their correctness. Finally, I impose non-negative constraints on our solution as it favours biological interpretability since only additive combinations of the latent features are allowed[29].

1.1 Data-driven regularisation

Observe that the second term in Eq. 1.1 also acts as a regularisation factor, so no additional regularisation term is required. To understand this, let first consider the matrix formulation of Eq. 1.1:

$$\begin{aligned} &\min_{W,H} \mathcal{L}(W,H) = \frac{1}{2} \parallel \mathbb{I}_\Omega \circ (X - WH) \parallel_F^2 + \frac{\alpha}{2} \parallel \mathbb{I}_{\Omega^c} \circ (WH) \parallel_F^2 \\ &\text{subject to the non-negative constraints } W, H \geq 0. \end{aligned} \tag{1.2}$$

where $\mathbb{I}_\Omega, \mathbb{I}_{\Omega^c} \in \mathbb{R}^{n \times m}$ are binary indicator matrices for the entries in Ω and Ω^c, respectively, $\circ$ is the Hadamard or element-wise product of matrices, and $\parallel \ . \ \parallel_F$ is the Frobenius norm*.

*The Frobenius norm of a matrix A is defined as $\parallel A \parallel_F = \sqrt{\mathrm{Tr}(A^T A)}$, where $\mathrm{Tr}(.)$ is the trace of A.

The goal of my objective function in Eq. 1.2 is twofold. The first term aims to fit the model to the observed entries only, whereas the second term aims to fits the model to the zero entries. The difference between these two competing constraints lies in the penalisation parameter α. Let's consider the effect of this parameter in the learned signatures. In principle, if $\alpha = 0$, the solution will be dense, populated by scores that approximates the values in $\mathcal{F} \in \{1, 2, 3, 4, 5\}$. In this case, our system predictions would mean that a given drug can be associated with all the side effects but with differences in their corresponding side effect frequencies. Of course, that is the incorrect assumption for our problem. Intuitively, a value of $\alpha > 0$ affects the sparsity in the solution by raising the importance of the zeros. A large value of α must lead to sparser signatures (lower model complexity) while a small value of α must lead to denser signatures (higher model complexity). Notice that this is also a direct consequence of a large number of zeros in X. Given that the complexity of the solution is controlled by the intrinsic structure of the data (entries in X), I called this phenomenon data-driven regularisation. This is different from other types of regularisation such as $\mathcal{L}_1$- or $\mathcal{L}_2$- norms, commonly used in other state-of-the-art matrix decomposition models[42]. While $\mathcal{L}_1$- or $\mathcal{L}_2$- norms constraint the individual latent matrices W and H to reduce model complexity, the data-driven regularisation constraints a well-defined set of entries ($\mathbb{I}_{\Omega^c}$) in WH. The sparsity of W and H is a consequence of the data-driven regularisation rather than of a direct penalisation in its entries. Data-driven regularisation is also different from the most recent nuclear-norm regularisation[11], that constraints the spectra of singular values in an SVD decomposition of X.

To minimise the loss function in Eq. 1.2 subject to the non-negative constraints, I developed a novel iterative algorithm that uses the following multiplicative update rule:

$$\begin{aligned} W &= W \circ \frac{XH^T}{(\mathbb{I}_\Omega \circ WH + \alpha\, \mathbb{I}_{\Omega^c} \circ WH)H^T} \\ H &= H \circ \frac{W^T X}{W^T(\mathbb{I}_\Omega \circ WH + \alpha\, \mathbb{I}_{\Omega^c} \circ WH)} \end{aligned} \tag{1.3}$$

this procedure does not require setting a learning rate nor applying a projection function and satisfies the Karush-Kuhn-Tucker (KKT) complementary conditions of convergence. I prove the convergence of my algorithm as follow:

Theorem 1 (Convergence). *The cost function $\mathcal{L}(W, H)$ in Equation (1.2) converges to a local minimum under the multiplicative update rule in Equation 1.3.*

Proof. From the theory of constrained optimisation[89,90], we know that we need to show that at convergence, the solution given by the multiplicative learning algorithm satisfies the well-known Karush-Khun-Tucker (KKT) complementary conditions:

$$\frac{\partial \mathcal{L}(W,H)}{\partial W} W = 0; \quad \frac{\partial \mathcal{L}(W,H)}{\partial H} H = 0 \tag{1.4}$$

The gradients of the loss function $\mathcal{L}(W, H)$ are:

$$\begin{aligned} \frac{\partial \mathcal{L}(W,H)}{\partial W} &= -(X - \mathbb{I}_{\Omega} \circ WH)H^T + \alpha(\mathbb{I}_{\Omega^c} \circ WH)H^T \\ \frac{\partial \mathcal{L}(W,H)}{\partial H} &= -W^T(X - \mathbb{I}_{\Omega} \circ WH) + \alpha W^T(\mathbb{I}_{\Omega^c} \circ WH) \end{aligned} \tag{1.5}$$

At local minimun, $W = W^*$ and $H = H^*$ must satisfy KKT condition in Eq. 1.4. Therefore, replacing Eqs. 1.5 in 1.4 and reordering we can obtain the multiplicative rules in Eq. 1.3. Therefore, the algorithm satisfies KKT conditions and converges to a local minimum. □

For the implementation, I followed the guidelines in [91]. I added a small number $\varepsilon \propto 10^{-16}$ to the denominators of in Eq. 1.3 to prevent division by zero at each iteration. I initialised W and H by sampling from an uniform distribution in the range $[0, 0.01]$. Furthermore, to avoid the well-known degeneracy [29,91] associated with the invariance WH under the transformation $W \to W\Delta$ and $H \to \Delta^{-1}H$, for a diagonal matrix $\Delta \in \mathbb{R}^{k\times k}$, I normalised H at each iteration as follow[†]

$$H_{pj} \leftarrow \frac{H_{pj}}{\|h_p\|_F}$$

where h_p denotes the vector corresponding to the pth row in H. The stopping criteria of my algorithm was based on the maximum tolerance in the change in the elements of W and H, which typically occurred in 2000 iterations.

[†]Notice that although the product $WH = W\Delta\Delta^{-1}H$ does not changed, different normalisations will lead to different signatures, that is $W\Delta_1 \neq W\Delta_2$ and $\Delta_1^{-1}H \neq \Delta_2^{-1}H$ if $\Delta_1 \neq \Delta_2$.

To predict specific frequency classes, we need a way to assign the learned scores by our model $\hat{X} = WH$ to the specific frequency classes $\mathcal{F} \in \{1, 2, 3, 4, 5\}$. Assume for now that we can estimate, from our model's predictions, the likelihood functions $P(\hat{x}|\mathcal{C}_{\mathcal{F}})$ for each of the frequency classes in $\mathcal{F}$ [‡]. Here $\mathcal{C}_F$ denotes a class in $\mathcal{F}$ and $\hat{x}$ denotes a vector of predicted scores for a set of entries in X belonging to class $\mathcal{F}$. Then, for a new drug-side effect pair (i, j) with predicted score $\hat{x}_{ij}$, we assign a class label $\hat{y} = \mathcal{C}_{\mathcal{F}}$ for some $\mathcal{F}$ as follows:

$$\hat{y} = \underset{\mathcal{F}\in\{1,2,3,4,5\}}{\arg\max}\ P(\mathcal{C}_{\mathcal{F}}|\hat{x} = \hat{x}_{ij}) \tag{1.6}$$

which is a *maximum a posteriori* (MAP) decision rule that selects the most probable class. To estimate the posterior, we used a Naive Bayes classifier such that the MAP in Eq. 1.6 can be written in terms of the likelihood functions $P(\hat{x}|\mathcal{C}_{\mathcal{F}})$ and the class prior probabilities $P(\mathcal{C}_{\mathcal{F}})$, as follows:

$$\hat{y} = \underset{\mathcal{F}\in\{1,2,3,4,5\}}{\arg\max}\ P(\mathcal{C}_{\mathcal{F}})P(\hat{x} = \hat{x}_{ij}|\mathcal{C}_{\mathcal{F}}) \tag{1.7}$$

Unfortunately, due to incomplete data and biases on the observed entries (recall that in Fig. 1.1, the observed data is biased towards frequent side effects), we cannot obtain reasonable estimates for the priors for each class, therefore, we assume uniform priors [§]. Therefore, our final MAP rule was simply based on the maximum likelihood estimation, as fol-

[‡]The procedure is detailed in section 2.2

[§]To understand this, consider the distribution of classes shown in Fig. 1.1. First, we do not have reasonable estimation of the number of `true` zeros, as these are unreported in the databases. Second, as our dataset is built from data from clinical trials, the distributions of classes based on the observed entries is biased towards the `frequent` class. Altogether, missing data and biases represent a challenge for our model.

lows:

$$\hat{y} = \underset{\mathcal{F}\in\{1,2,3,4,5\}}{\arg\max}\ P(\hat{x} = \hat{x}_{ij}|\mathcal{C}_{\mathcal{F}}) \tag{1.8}$$

1.4 Connection with standard matrix completion

The goal of matrix completion is to fully recover a matrix from its observed entries[92,11]. Matrix completion aims to learn an unknown parameter, a matrix $Z \in \mathbb{R}^{n\times m}$, with high dimensionality, based on few observations. It is typically assumed that the parameter Z lies in a lower dimentionality, which translates into a minimisation of the rank of Z as follows:

$$\begin{aligned} &\text{minimise} && \text{rank}(Z) \\ &\text{subject to} && \sum_{(i,j)\in\Omega} (X_{ij} - Z_{ij})^2 \leq \delta \end{aligned} \tag{1.9}$$

where $\delta \geq 0$ is a regularisation parameter and as before, Ω represent the observed entries in X. It is common to solve Eq. 1.9 using matrix decomposition techniques, that is, by approximating $Z \simeq WH$, where $W \in \mathbb{R}^{n\times k}$ and $H \in \mathbb{R}^{k\times m}$ with $k \ll \min(n, m)$.

My model presented in Eq. 1.2 can be also written as a matrix completion task, as follow:

$$\begin{aligned} &\text{minimise} && \text{rank}(Z) \\ &\text{subject to} && \sum_{(i,j)\in\Omega} (X_{ij} - Z_{ij})^2 \leq \delta_{\Omega}, \sum_{(i,j)\in\Omega^c} (X_{ij} - Z_{ij})^2 \leq \delta_{\Omega^c} \end{aligned} \tag{1.10}$$

where Ω^c represents the set of entries corresponding to the zeros in X and δ_{Ω} and δ_{Ω^c} are

regularisation parameters for the observed and unobserved entries, respectively. Assuming that the unobserved entries are coupled with higher level of uncertainty, we would expect $\delta_{\Omega^c} \ll \delta_{\Omega}$. Therefore, in terms of standard matrix completion, my model minimises the rank of Z while making specific assumptions about different levels of uncertainty in the entries of the data matrix X.

1.5 General remarks

Let's consider the general low-rank model loss function:

$$\mathcal{L}(W,H) = \frac{1}{2} \parallel \zeta \circ (X - WH) \parallel_F^2 + \varphi(W,H) \tag{1.11}$$

where $\zeta \in \mathbb{R}^{n\times m}$, $W \in \mathbb{R}^{n\times k}$, $H \in \mathbb{R}^{k\times m}$, $\varphi(.)$ represents an appropriate regularisation function and $\circ$ represents element-wise product between matrices.

The loss function in Eq. 1.11 represents a family of low-rank models. The first term is the *fitting constraint*, which fits the model to either the entries in X that are defined by an appropriate selection of ζ. The second term in Eq. 1.11 is a *regularisation constraint*, which is typically applied to prevent overfitting in the solution. Therefore, to better understand how my proposed model connects to the general family of low-rank models, consider the following remarks:

- ζ is problem dependent. In rating-based recommendation systems[18] where the goal is to predict the rating that a user will give to movies, ζ is defined as follow:

$$\zeta_{ij} = \begin{cases} 1 & \text{if} \quad x_{ij} \in \{1,2,3,4,5\} \\ 0 & \text{otherwise} \end{cases}$$

In this case, the optimisation is performed on the observed entries only. Conversely, in ranking-based recommendation system [20], where the goal is to predict the movie(s) that the user will watch in a shortlist of top-N recommendations, ζ is defined as follow:

$$\zeta_{ij} = 1 \qquad \forall (i,j)$$

In this case, the zeros are also considered in the learning.

My formulation to predict the frequencies of drug side effects represents a hybrid formulation between the rating-based and ranking-based recommendation system. That is, I defined ζ as follows:

$$\zeta_{ij} = \begin{cases} 1 & \text{if} \quad x_{ij} \in \{1,2,3,4,5\} \\ \sqrt{\alpha} & \text{if} \quad x_{ij} \in \{0\} \end{cases}$$

for $\alpha \in [0,1]$.

- Constraints on the learned matrices W and H. The constraints imposed on the learning of the latent representations can have a significant impact in both prediction performance and model interpretability. Data sparsity tend to cause overfitting, and it is typically avoided by using a $\mathcal{L}_2$ regularisation.

  Going beyond overfitting, there are additional constraints that can be imposed to W and H. For instance, orthogonal representations [93] for which $WW^T = I_n$ and $H^T H = I_m$ (identity matrices). The most common algorithm to achieve orthogonality is truncated singular value decomposition (TSVD) [94] that finds a decomposition $X \simeq USV^T$, where $U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{m \times k}$ are orthogonal matrices, and $S \in \mathbb{R}^{k \times k}$ is a

diagonal matrix containing the kth largest singular values.

In the seminal work of Lee and Seung on non-negative matrix factorisation [29], it has been shown that when negative weights are allowed in the learning of the latent representations, these tend to be holistic, non-interpretable representations. Conversely, only when these representations are constrained to be non-negative, these reflects a parts-based representation (e.g. parts of faces in images), as only additive combinations of the features are allowed.

2 Empirical evaluation

2.1 Datasets

I used the drug side effect frequencies from the Side effect Resource (SIDER) database version 4.1 [65]. In the database, around 40% of the pairs have frequency information, whereas, for the remaining associations, the frequency is unknown. Drugs are indexed by their PubChem IDs, and side effect terms are mapped to the Medical Dictionary for Regulatory Activities (MedDRA) taxonomy. I only considered side effect terms that were Preferred-Terms (PT) in MedDRA. I also kept only the drugs with known monotherapy Anatomical Therapeutic and Chemical (ATC) classification according to the 2018 World Health Organisation (WHO) release. Side effect frequencies were found in different formats: exact values, range of values or frequency class labels. I standardised these frequencies to frequency classes by encoding them as follow: very rare (=1), rare (=2), infrequent (=3), frequent (=4) and very frequent (=5). Our dataset contains 37,441 frequency associations that cover 759 drugs and 994 side effect terms (see Appendix 1 for details). Drug protein targets and drug

SMILES were obtained from Drugbank database v5.0.5 (3). Drug Anatomical, Therapeutic and Chemical (ATC) codes and the drugs route of administrations (Adm.R) were obtained from WHO release 2018 (see Appendix 2 for details).

2.2 Cross-validation procedure

I set apart 10% of randomly held-out associations of the observed entries in X for testing (held-out test set). I then used a standard ten-fold cross-validation procedure on the remaining 90% of the associations for the setting of the model parameters k and α. I framed the problem as simultaneously predicting the frequency classes and the presence/absence of the associations. Therefore, I used two evaluation metrics:

- *Root mean squared error* (RMSE). To assess the overall prediction performance of the frequency class values estimation. RMSE is a standard measure of regression. RMSE is a non-negative quantity bounded in the interval $[0, \infty)$.

- *Area Under the Receiver Operating Characteristic Curve* (AUROC). Due to the lack of experimentally validated zeros, I followed previous approaches for binary drug side effect prediction[6] and framed the prediction problem as a binary classification.

The overall performance of my model in the cross-validation was based on the mean RMSE and AUROC of the ten folds. To select the model parameters, I first chose α based on a good binary classification performance (AUROC) while ensuring a good RMSE.

To predict the specific frequency classes, for each validation set in the ten-fold cross-validation, I collected the frequency class values and their corresponding predicted scores. Then, for each of the five frequency classes, I fitted a normal kernel smoothing function to

the predicted scores and obtained a probability density function for each of the five classes. I obtained the following decision thresholds for the predicted scores: $1.26, 2.43, 3.25$ and 3.93. Furthermore, due to the lack of experimentally validated zero values, in order to discriminate the zero associations, I followed an approach similar to the one used by Cami et al.[6] and chose a threshold using the ROC curve at a sensitivity of 0.97 given a specificity of 0.57.

2.3 Prediction performance

In the ten-fold cross-validation, I obtained a good performance with $\alpha = 0.05$ and $k = 10$ (mean RMSE $= 1.372 \pm 0.021$ and mean AUROC $= 0.920 \pm 0.003$ – Fig. 1.3. The performance of the algorithm is robust with respect to the setting of the parameters k and α – Figures 1.4, 1.5.

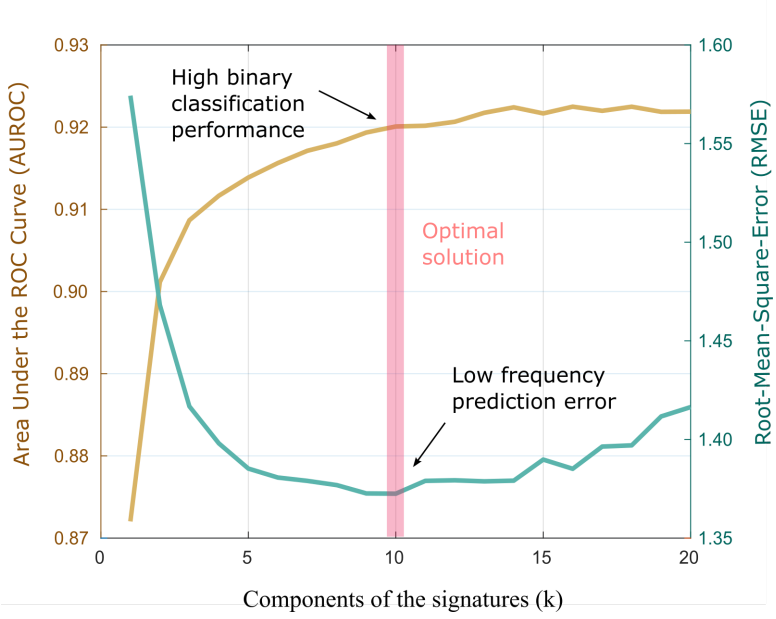

Figure 1.3: Model selection. Selection of the optimal number of latent features (or signatures components) based on the RMSE-AUROC trade-off. Here $\alpha = 0.05$.

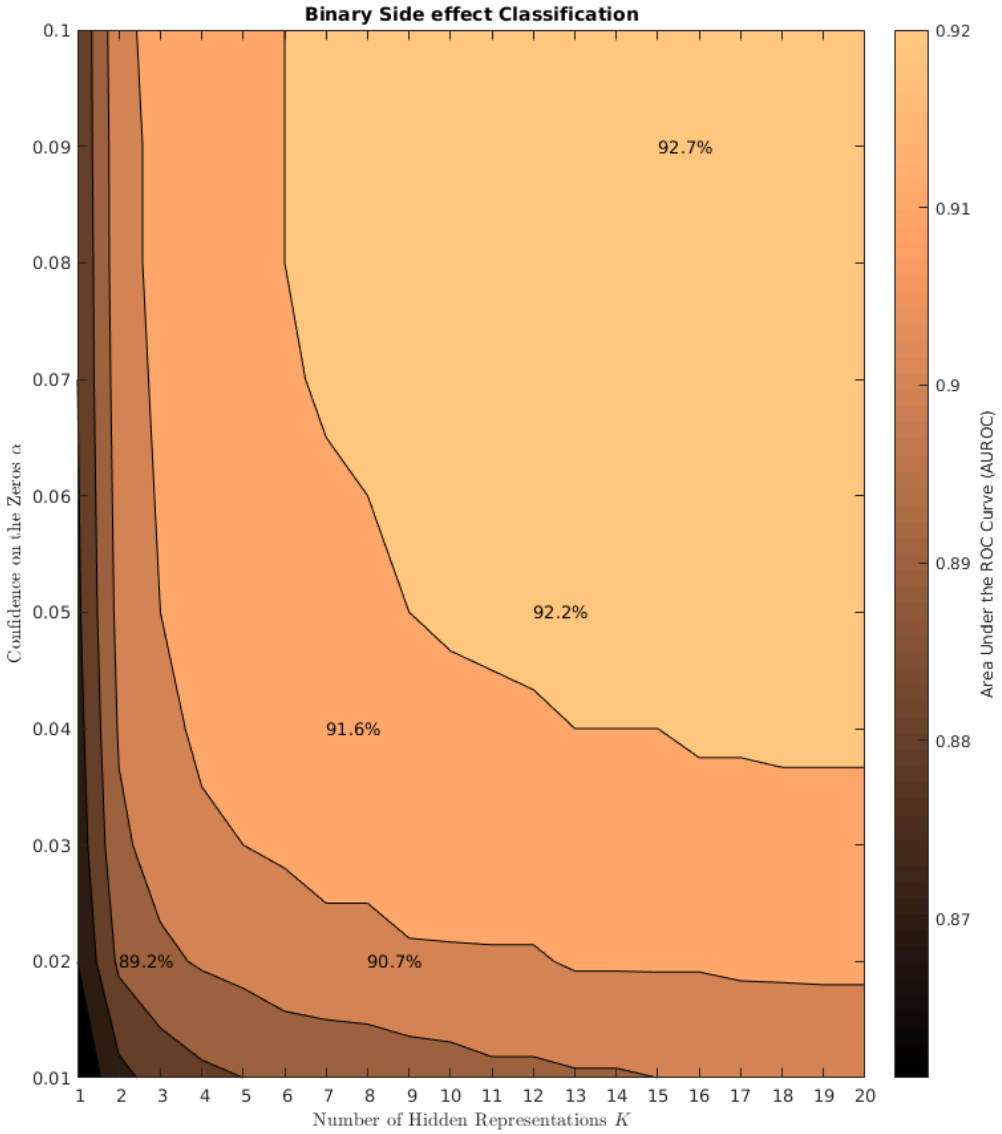

Figure 1.4: Contour plot of mean AUROC of the ten-fold cross-validation performance for the binary side effect classification problem. The higher the AUROC, the better we can correctly identify true associations. The performance is divided, for clarity, into nine contour levels for varying values of the number of representations (k) and the confidence in the zeros (α).

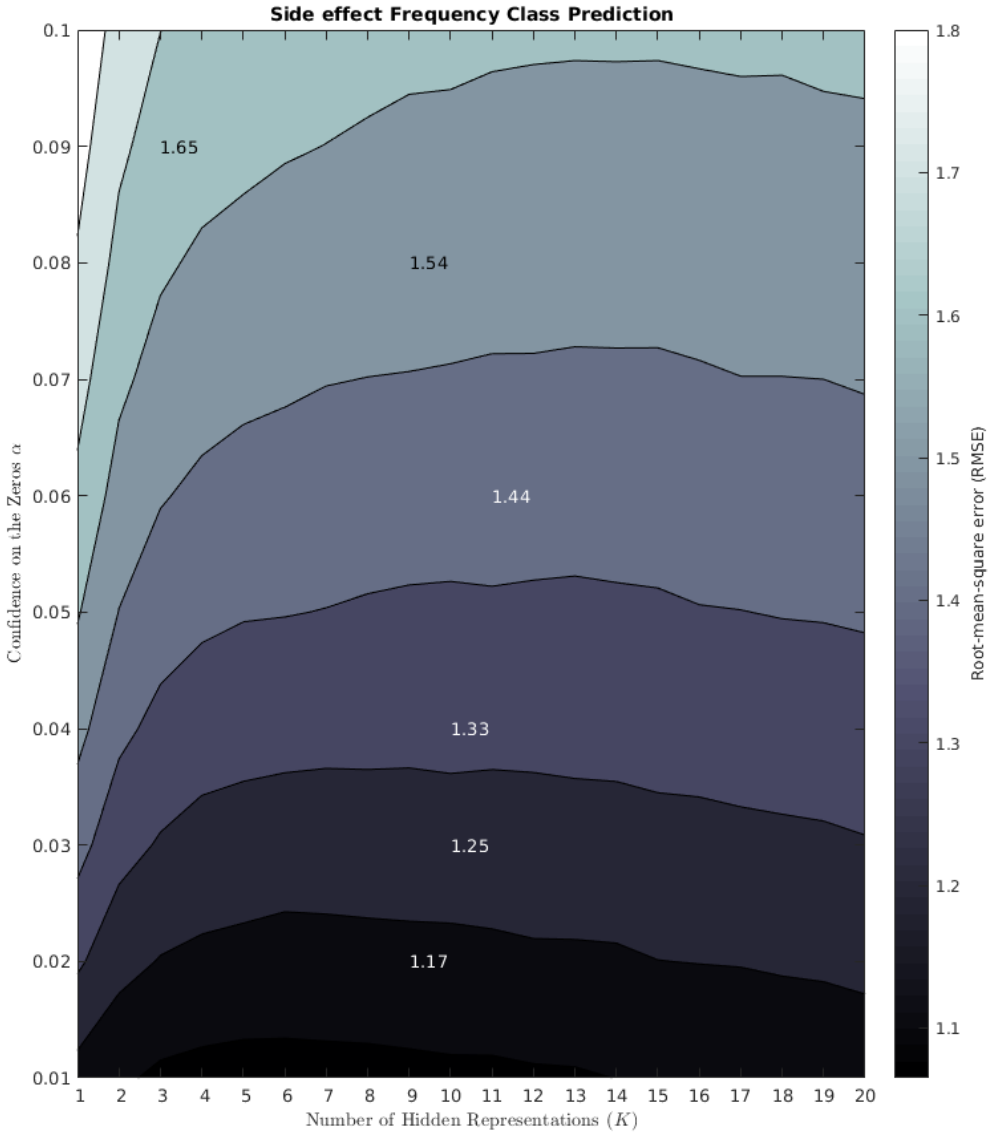

Figure 1.5: Contour plot of mean RMSE of the ten-fold cross-validation performance for the side effect frequency class value prediction problem. The smaller the RMSE, the better we can predict the true frequency value of the drug side effects. The performance is divided for clarity into nine contour levels for varying values of the number of representations (k) and the confidence in the zeros (α).

On the held-out test set, my model scored an RMSE of 1.32 and an AUROC of 0.932. Figure 1.6a shows, for each of the five frequency classes in the test set, the histogram of the values that were predicted for that class. The Pearson correlation between the predicted scores and their corresponding frequency classes was $\rho = 0.47$ (Significance, $p < 2.40 \times 10^{-209}$); the differences between the distributions of scores for the five frequency classes were statistically significant (Kruskal-Wallis One way ANOVA Significance at 1%, $p < 1.15 \times 10^{-193}$).

Figure 1.7a shows the accuracy at predicting side effect frequency classes. For any given class, the most predicted class is the correct one, and the prediction accuracy ranges from 55.2% to 75.5% when including the contiguous lower class, and 67.8% to 94% when both contiguous classes are considered. Looking at the first column in the figure, we notice how my system rarely (0.72%) fails to detect a very frequent side effect and seldom misses side effects in the frequent (2.68%), infrequent (2.52%) and rare (3.11%) classes. The number of undetected side effects only increases for the very rare class (16.94%) — probably due to the small number of known associations in this class. As illustrative examples, Figure 1.7c presents the predicted frequency scores for the anticonvulsant drug Gabapentin, a top 50 prescribed drug in the U.S.[95], and the side effect arrhythmia, critical in cardiotoxicity assessment[96].

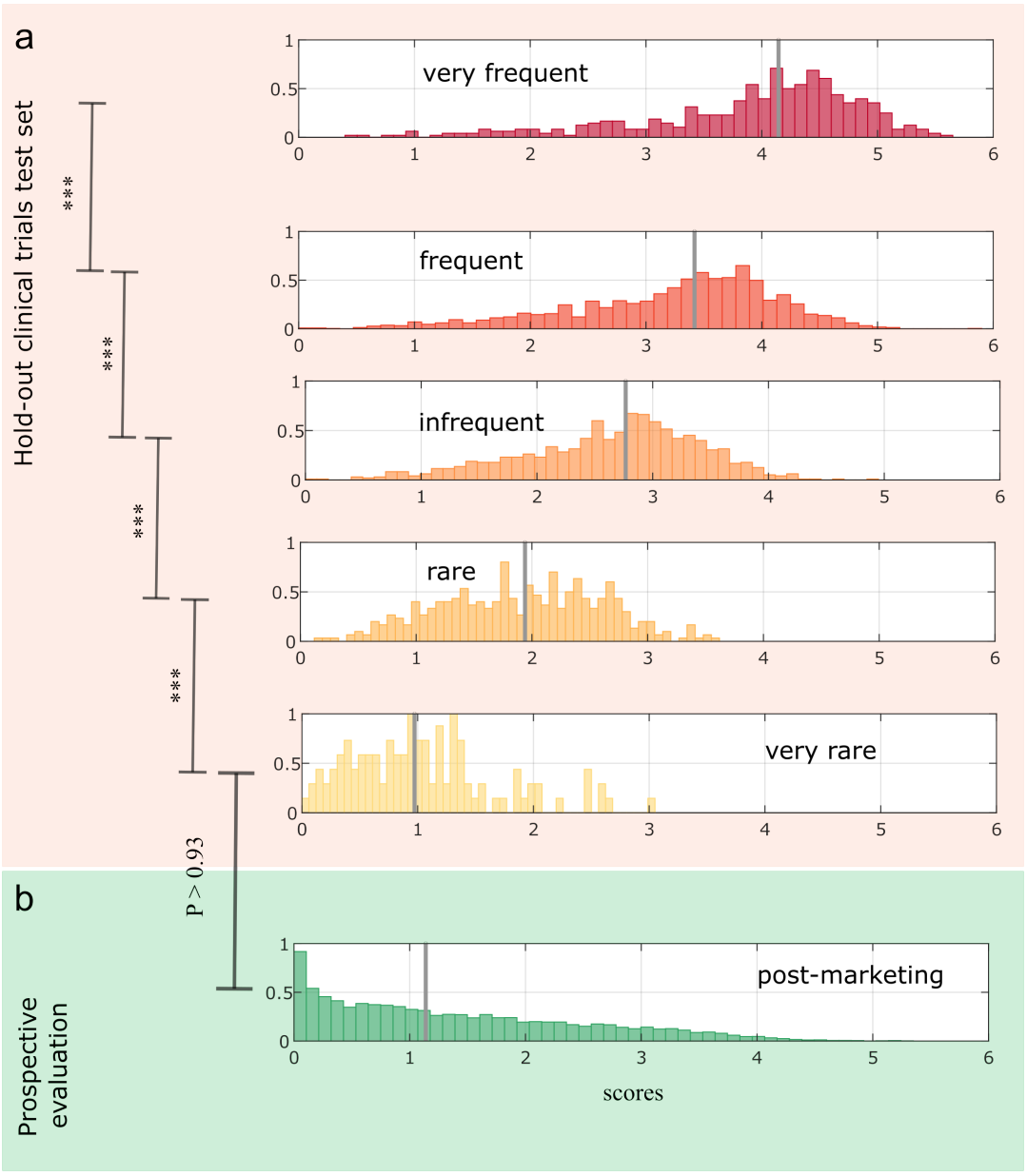

Figure 1.6: Distributions of scores for held out and post-marketing test sets. (a) Normalized histogram of scores obtained for each of the five frequency classes in the held out test set. The differences in the distributions between the classes are statistically significant. (b) Normalised histogram of scores obtained for the post-marketing test set. Significance levels between the scores are indicated with asterisks ($p \leq 0.001$, ***), ($p \leq 0.01$, **). Wilcoxon rank sum test was used in all the cases. Median values are shown as grey vertical lines.

I further tested the performance of my system at predicting the frequency of side effects that were detected after the drugs had reached the market. This amounts to a prospective evaluation where post-marketing data is used as the test set — it is a realistic scenario that preserves the order in which the information becomes available. Post-marketing side effects are typically regarded as side effects of very rare occurrence in the population[61,97]. I collected 9, 387 post-marketing associations — these had a value of zero in the corresponding entries in X used for training (Appendix 2). The statistical analysis of the distribution of scores obtained for these post-marketing associations show no significant differences with the scores obtained for the very rare class in the held-out test set (Fig. 1.6b, Wilcoxon Significance, $p > 0.936$). Fig. 1.7b shows the percentages of post-marketing associations assigned to each class by maximum likelihood: 55.5% of the post-marketing associations were predicted to be either very rare or rare, while only 2% were predicted as very frequent.

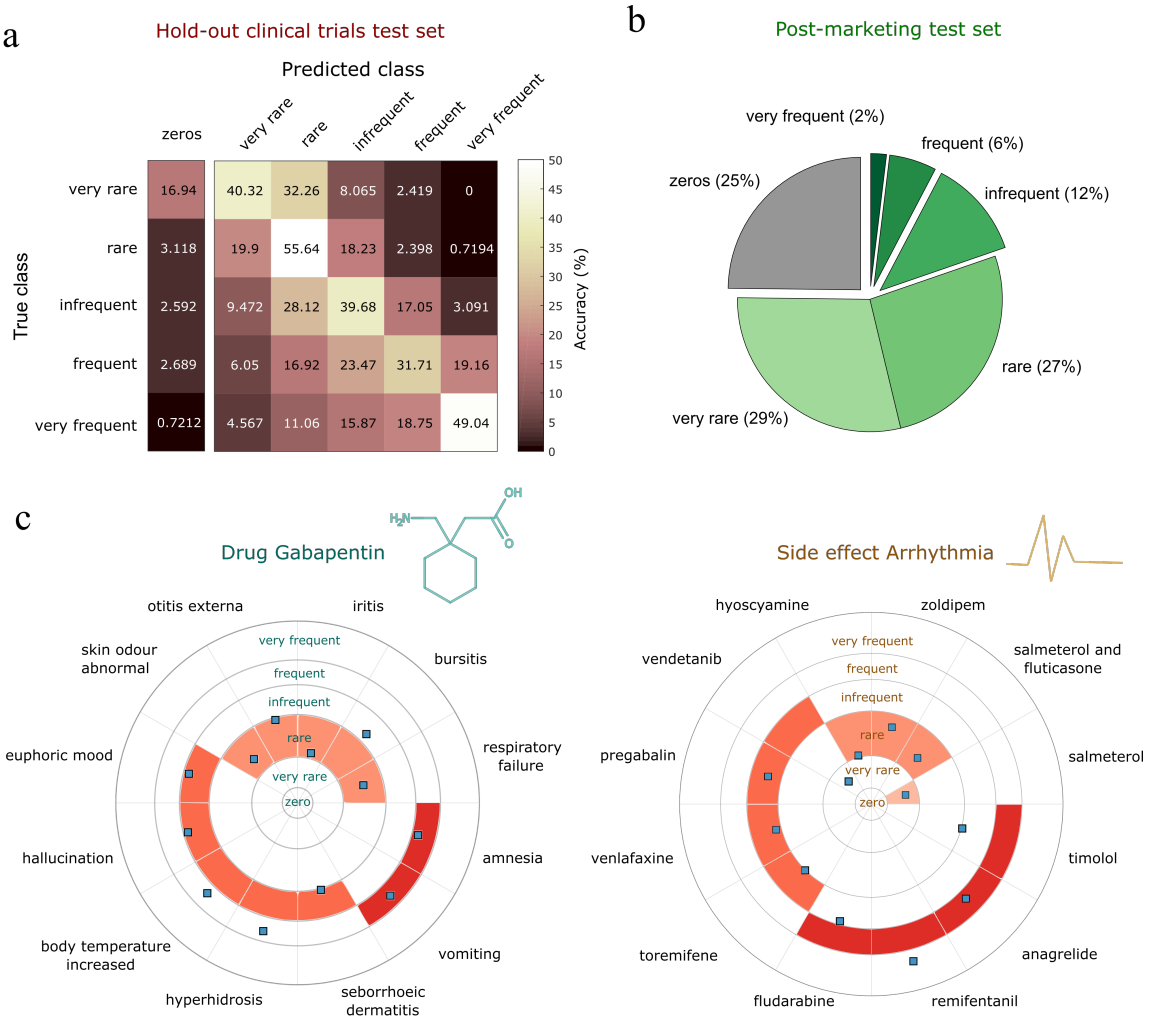

Figure 1.7: Evaluation of side effect frequency predictions. (a) Accuracy percentages for the predictions in the held-out test set. Frequency classes are predicted by maximum likelihood. Zeros, corresponding to "no side effect" prediction, are predicted for score values below 0.42 (corresponding to 0.97 sensitivity given 0.57 specificity). (b) Distribution of predicted classes assigned to post-marketing data. (c) Illustrative examples from the held-out test set. Twelve randomly-chosen predictions for the anticonvulsant drug Gabapentin (left) and the cardiovascular side effect arrhythmia (right) are shown around polar plots, each in a dedicated sector. Gray concentric circles between frequency classes correspond to thresholds learned by maximum likelihood. The correct class for each association is coloured in each circular sector, while predicted scores are shown as blue squares.

3 Biological interpretability of the model

The effectiveness of the model at predicting the frequency of side effects prompted us to analyse whether the learned signatures are informative of the biology underlying drug activity. I began by analysing whether the signatures were reproducible across independent runs. This is important to ensure that any biological interpretability arising from the latent representations is reproducible. In the next subsections, I shall present the reproducibility procedure and the pharmacological interpretation of my model.

3.1 Model reproducibility

Using all the available data in X, I followed the reproducibility procedure used by Alexandrov et al. to study cancer mutational signatures[98,99]. The reproducibility procedure is summarise in the following steps:

- *(Step 1)* Perform the decomposition for $10,000$ times using all the available data in the matrix X with the optimal parameters $k = 10$ and $\alpha = 0.05$.

- *(Step 2)* Select the best 100 solutions that minimises the cost function and aggregate them into the matrices $\mathcal{W}$ of $n \times L$, and $\mathcal{H}$ of $L \times m$, where $L = 10 \times 100 = 1,000$ latent features, $n = 759$ drugs and $m = 994$ side effects.

- *(Step 3)* Apply a partition clustering algorithm on the columns of $\mathcal{W}$ (and rows of $\mathcal{H}$) using cosine distance as the metric¶. Then, I run k-means++ algorithm[100] with $k = 10$ for $10,000$ times to find an optimal solution. The reproducibility of the

¶The cosine distance for two vectors w_1 and w_2 is defined as $d(w_1, w_2) = 1 - \frac{w_1 w_2^T}{\sqrt{(w_1 w_1^T)(w_2 w_2^T)}}$. The cosine similarity is $1 - d(w_1, w_2)$.

representation was then measured by the tightness and separation of the clusters obtained. I used the cosine similarity-based average silhouette width[101] of each cluster as a measure of reproducibility of each component in the signature. The silhouette width for each component of the signature (a column in $\mathcal{W}$ or a row in $\mathcal{H}$) is a measure of how similar that component is to components in its own cluster, when compared to components in other clusters. The silhouette width s_i for the ith component of the signature is defined as:

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)} \tag{1.12}$$

where a_i is the average distance from the ith point (i.e. component of the signature) to the other points in the same cluster as i, and b_i is the minimun average distance from the ith point to points in a different clusters, minimised over clusters. The silhouette value ranges from -1 to $+1$. A value close to $+1$ indicates that a component is very similar to other components in its cluster but very dissimilar to neighbouring clusters.

I found that eight out of the ten components of the signatures have a median reproducibility score above 80% (Figs. 1.8-1.9). Using as a reference the best solution of the 10,000 runs, the highly reproducible components on both drugs and side effect signatures are components $\{1, 2, 4, 5, 6, 7, 8, 10\}$. Hereafter, I shall report the biological and pharmacological analysis found for the best solution.

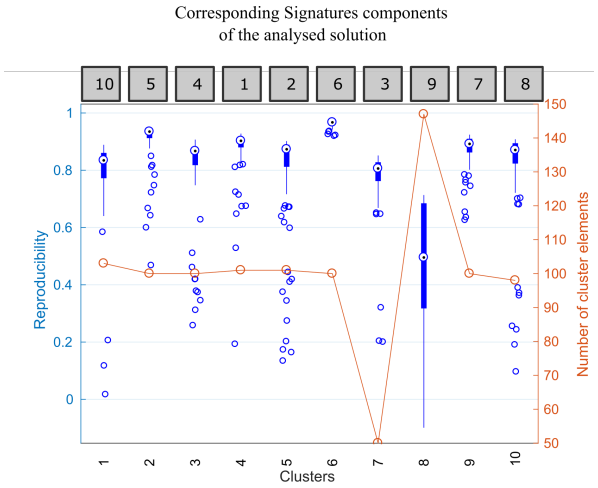

Figure 1.8: Reproducibility analysis of the drug signature components for the best 100 runs out of 10,000 runs of my decomposition algorithm. *(left axis)* Reproducibility of each k-means clusters measured using the cosine-based silhouette value. The silhouette value for each component is a measure of how similar that component is to component in its own cluster when compared to component in other clusters. *(right axis)* The number of elements in each cluster. Ideally, we would expect 100 components in each cluster.

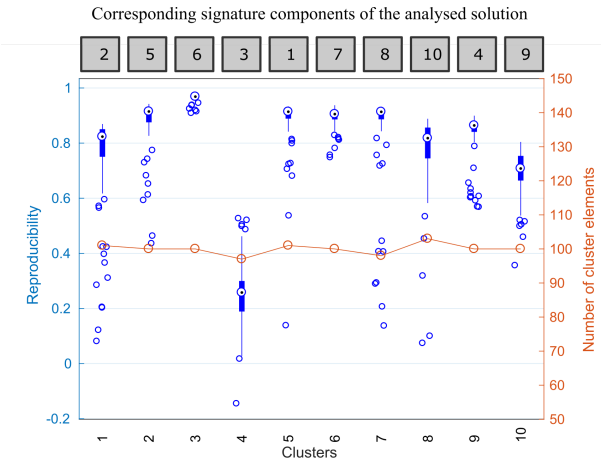

Figure 1.9: Reproducibility analysis of the side effect signatures components for the best 100 runs out of 10,000 runs of my decomposition algorithm. *(left axis)* Reproducibility of each k-means clusters measured using the cosine-based silhouette value. The silhouette value for each component is a measure of how similar that component is to component in its own cluster when compared to component in other clusters. *(right axis)* The number of elements in each cluster. Ideally, we would expect 100 components in each cluster.

Having shown that the features were highly reproducible, allowed me to investigate the link between drug signatures and drug clinical activities. I hypothesised that the signature for two drugs should be similar when they share clinical activity. Clinical activity for drugs was defined based on their main Anatomical, Therapeutic and Chemical (ATC) class level –— a five-level hierarchical organisation of terms where lower levels of the hierarchy contain more specific descriptors of clinical activity. I quantify the similarity between two drug or side effect signatures using the cosine similarity over the set of latent features. In detail, given two drug signatures $w_1 \in \mathbb{R}^k$ and $w_2 \in \mathbb{R}^k$ (rows in W), the drug signature similarity is given by the dot product of the vectors divided by the product of the norm of each vector.

$$S(w_1, w_2) = \frac{w_1 w_2^T}{\sqrt{(w_1 w_1^T)(w_2 w_2^T)}} \tag{1.13}$$

Therefore, the similarity for non-negative signatures ranges from 0 to 1.

Figure 1.10a shows that the cosine similarity between the signature of drugs within an ATC class is higher than the similarity between classes. I further checked whether the similarity between drug signatures was predictive of the clinical activity at each level of the ATC hierarchy. Following the approach by Tattoneti et al.[97], I frame it as a binary classification problem, where the scores are the drug signature similarities and we predict whether pairs of drugs share or not a given relationship. The performance is measured using the area under the receiver operating curve (AUROC). Fig.1.10b shows that the prediction performance increases when considering terms located lower in the ATC hierarchy. My findings correctly reflect the fact that drug clinical activity becomes more similar as we move to lower

(more specific) levels of the ATC hierarchy.

3.3 Drug signatures predict drug molecular activity

Encouraged by these results, I decided to test whether drug signature similarity can even be used for the prediction of drug targets. I found that drug signature similarities are predictive of shared protein targets between drugs (AUROC = 68.38%) (Fig. 1.10c) and the predictions are better than baselines previously used elsewhere[97,102], such as the 2D Tanimoto chemical similarity (AUROC = 59.26%) and the Jaccard side effect similarity (AUROC = 61.07%).

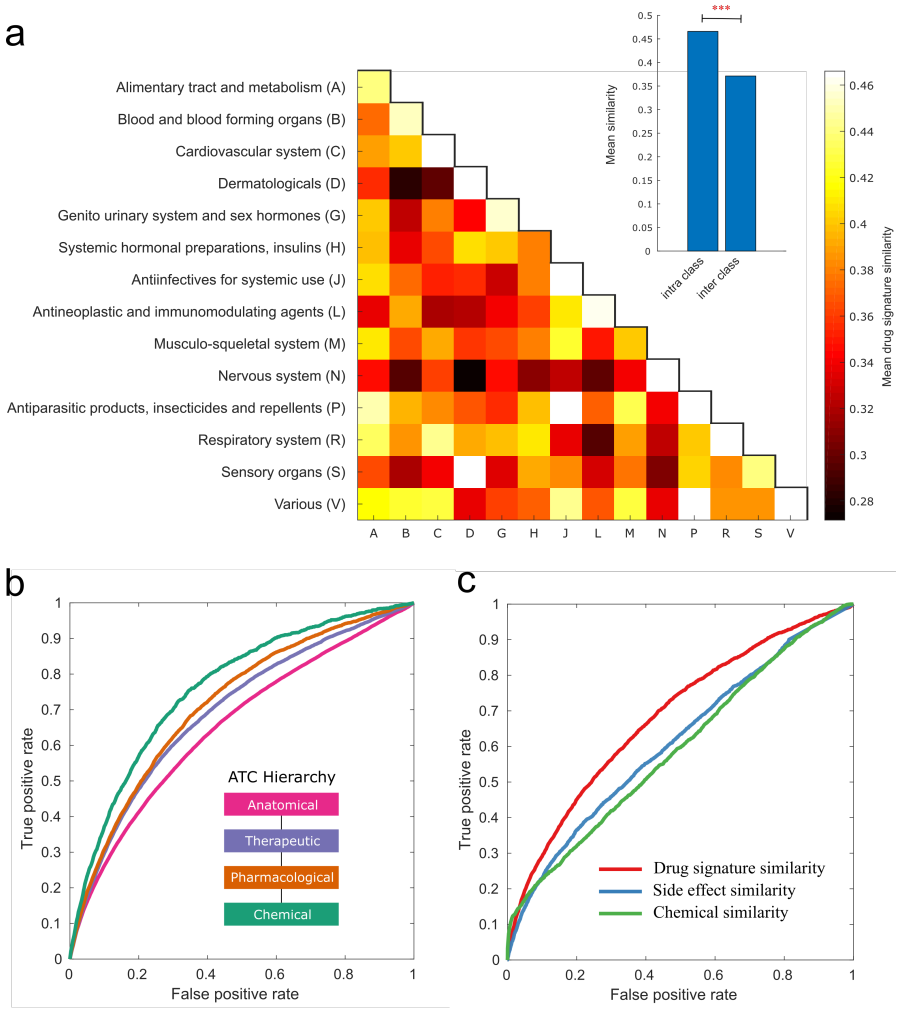

Figure 1.10: Drug signatures capture drug clinical and molecular activity. (a) Heat maps of mean drug signature similarities per anatomical class. Each (x, y) tile represents, for each main Anatomical, Therapeutic and Chemical (ATC) drug category, the mean similarity of drug pairs where one drug belongs to category x and the other to category y. The value ranges from 0.27 (Nervous system - Dermatological) to 0.55 (Nervous system- Nervous system). The colours range between the minimum mean similarity and 0.466, with all values above 0.466 (In the diagonal: 0.471 (C), 0.512 (D), 0.55 (N), 0.47 (P), 0.52 (R), 0.475 (V)) set to 0.466. *Inset*: the average intra-class similarity is significantly higher than the average inter-class similarity (t-test Significance, $p < 2.62 \times 10^{-13}$). (b) ROC curve representing the ability of the drug signature similarity to predict which pairs of drugs share Anatomical, Therapeutic and Chemical (ATC) category at each of the different levels in the ATC hierarchy. (c) ROC curve representing the ability of the drug signature similarity, side effect similarity and Tanimoto chemical similarity scores to predict which pairs of drugs share targets.

Similarly, I analysed the link between side effect signatures and the anatomy/physiology of the side effect phenotypes. Side effects were grouped based on their system organ classes according to MedDRA — the top level of the MedDRA hierarchy. I found that signatures for two side effects tend to be more similar when they are phenotypically related (Figure 1.11). Moreover, the similarity between side effect signatures is predictive of shared MedDRA category at each of the different levels of the MedDRA hierarchy, and predictions improve as we move to more specific terms in the hierarchy (Fig. 1.12).

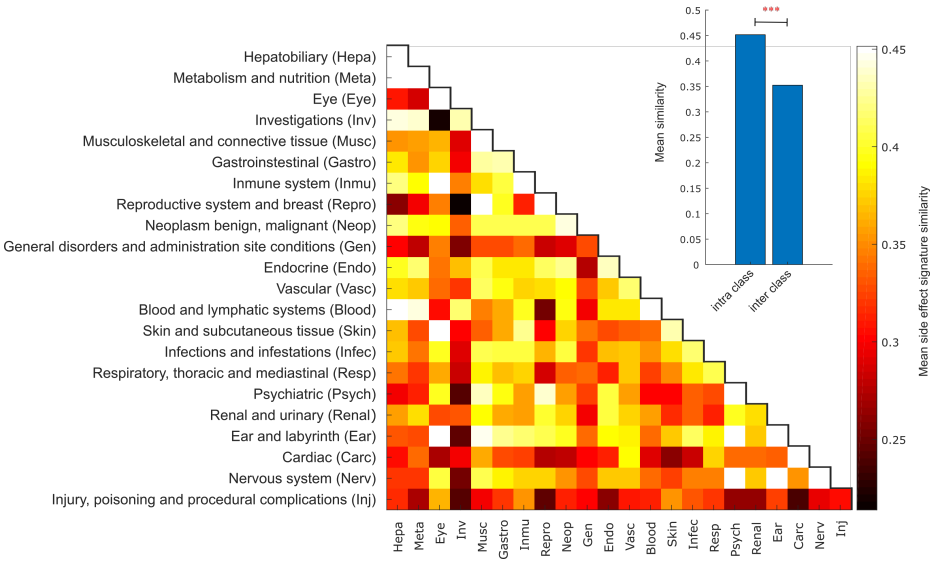

Figure 1.11: Side effect signatures encode side effect phenotypes. Each (x, y) tile represents, for each main Medical Dictionary for Regulatory Activities (MedDRA) classification of disorders, the mean similarity of side effect pairs where one side effect belong to category x and the other to category y. The value ranges from 0.21 (Reproductive systems - Investigations) to 0.58 (Psychiatric – Psychiatric). The colours range between the minimum mean similarity and 0.45, with all values above 0.45 (In the diagonal: 0.49 (Hepa), 0.55 (Eye), 0.57 (Repro), 0.49 (Blood), 0.58 (Psych), 0.54 (Carc), 0.47 (Nerv)) set to 0.45. Inset: the average intra-class similarity is significantly higher than the average inter-class similarity (t-test Significance, $p < 4.37 \times 10^{-16}$).

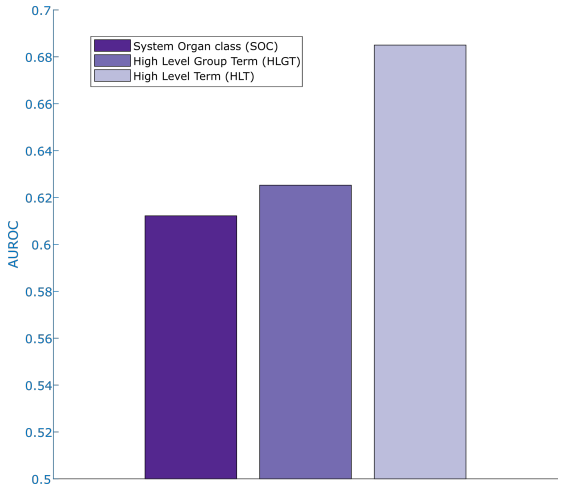

Figure 1.12: Predicting share side effect anatomical/physiological categories for different levels of the MedDRA taxonomy using side effect signatures similarity. Level 1 or System Organ class (SOC): 57,076 side effects that share and 436,445 do not. Level 2 or High-Level Group Term (HLGT): 12,097 shares and 481,424 do not. Level 3 or High-Level Term (HLT): 2,312 shares and 491,209 do not.

3.5 Pharmacological interpretation of the signature components

I have shown that the signatures of drugs and side effects, as a whole, encode meaningful biological information of drug molecular and clinical activity. A further important question is whether the individual components of the signatures are interpretable.

I grouped drugs and side effects according to their main anatomical classes, and I looked for significant activations of individual components of the signatures for each group. The groups were obtained using top-level terms in ATC and MedDRA hierarchies, respectively. I observed that, often, specific component(s) of the signatures tended to be significantly activated for drugs and side effects that were anatomically related – Table 1.1 and 1.2 summarises the correspondences that I found to be statistically significant (one-tailed Wilcoxon with Benjamini-Hochberg adjusted Significance, $p < 0.01$).

Let us analyse a few entries of the table in detail. Component 1 of the signature is sig-

Component	Anatomical drug category (ATC)	Anatomical side effect category (MedDRA)	Comments
1	Genitourinary and sex hormones (G), Antineoplastic and immunomodulating agents (L)	Reproductive system and breast, Musculoskeletal and connective tissues	Strongly associated with endocrine therapy drugs (L02) and sex hormones and modulators of the genital system (G03) drugs. Weakly with N06 (psychoanalytic)
2	Cardiovascular (C)	Cardiac, Vascular and Respiratory, thoracic and mediastinal	Also associated with anaesthetics (N01). Strongly associated with arrhythmias
3	Nervous system (N)	Respiratory, thoracic and mediastinal	A weak, less stable signature. Associated with antimycotics (J02) and psychoanalytic (N06)
4	Dermatological (D), Sensory organs (S)	Skin and subcutaneous tissue, Eye and immune system	Strongly related to epidermal and dermal conditions, Ocular infections, irritations and inflammations, including allergic conditions. Also associated with the nasal and transdermal delivery administration
5	Nervous system (N)	Nervous system, Psychiatric disorders	Specific to Nervous system drugs. It is associated with many subcategories of nervous system drugs, except anaesthetics (N01). Also, only weakly associated with psycholeptics (N05). Equal neurologic and psychiatric side effects

Table 1.1: Statistically significant associations between the components of the signatures 1 to 5 and groups of drugs and side effects.

nificantly associated with the sex hormone group of drugs and with the breast disorder group of side effects — these are top-level terms of the ATC and MedDRA hierarchies, respectively (Wilcoxon rank sum test with Benjamini-Hochberg adjusted significance, $p < 4.01 \times 10^{-12}$). When I performed a more in-depth pharmacological analysis by looking at lower levels in the ATC hierarchy (finer granularity), my analysis revealed that

Component	Anatomical drug category (ATC)	Anatomical side effect category (MedDRA)	Comments
6	Respiratory system (R)	Respiratory, thoracic and mediastinal, Infections and infestations	Also associated with drugs used in diabetes (A01), lipid modifying agents (C10) and urological (G04), highlighting some interactions with metabolism/haemostasis Also associated with inhalation and nasal administration
7	Anti-infectives for systemic use (J)	Gastrointestinal	Also linked to drugs for acid-related disorders (A02)
8	Nervous system (N)	Nervous system disorders, Psychiatric disorders	Specific to Nervous system drugs. Specifically, antipsychotics and anxiolytics (N05A/B) More psychiatric side effects. Prominently associated with mood and sleep disorders and disturbance. Associated with oral administration
9	Antineoplastic and immunomodulating agents (L), Anti-infective for systemic use (J)	Metabolism and nutrition, Investigations, Blood and lymphatic system	It is associated with antineoplastic agents (L01), antimycotics and antivirals, both for systemic use (J02/05). Also, with immunosuppressant drugs (L04). Associated with electrolyte and fluid balance conditions and hepatobiliary investigations
10	Antineoplastic and immunomodulating agents (L)	Blood and lymphatic system, Vascular disorders	Strongly associated with antineoplastic agents (L01) and weakly with antithrombotic (B01) Associated with haemorrhagic vascular disorders

Table 1.2: Statistically significant associations between the components of the signatures 6 to 10 and groups of drugs and side effects.

this component corresponds to sex hormones and modulators of the genital system (G03) drugs, and endocrine therapy (L02) drugs (adjusted, $p < 4.06 \times 10^{-6}, p < 1.24 \times 10^{-9}$,

respectively).

Another notable example is component 8 of the signatures. This component is specific to neurological drugs (adjusted, $p < 3.48 \times 10^{-31}$, but $p > 0.05$ for all other drug classes), and to side effects related to the nervous system and psychiatric disorders. In-depth pharmacological analysis reveals that this component is linked to antipsychotics and anxiolytics drugs and with psychiatric side effects (mood and sleep disorders). Conversely, component 5 of the signatures, which is also specific to neurological drugs (adjusted, $p < 1.82 \times 10^{-12}$, $p > 0.05$ for all other drug classes), has more balanced neurological and psychiatric side effect profiles.

In some cases, the signature components can be associated with more than one anatomical class, and the connection between the classes becomes apparent after considering the off-target or off-tissue effect of the drugs. As an example, consider component 2 of the signatures, which is strongly associated with both cardiovascular system drugs (adjusted, $p < 7.13 \times 10^{-10}$) and cardiac and vascular-related side effects (adjusted, $p < 5.24 \times 10^{-4}$, $p < 2.56 \times 10^{-17}$, respectively). There is, however, an unexpected link with nervous system drugs. In-depth pharmacological analysis reveals that component 2 is linked to anaesthetic drugs (N01) — the only neurological drugs associated with this component (adjusted, $p < 1.25 \times 10^{-2}$). Conversely, anaesthetic drugs are not statistically significantly associated with any other components — including components 5 and 8, which are neurological specific. Anaesthetic drugs reportedly affect the regular cardiac electrical activity by interacting with the ion channels — the component 2 of the signatures is indeed strongly associated with arrhythmias (adjusted, $p < 8.88 \times 10^{-10}$).

Furthermore, it is well known that drugs route of administration affects the side effects. I

tested whether components in the signatures can capture this relation. I found that specific components of the drug signatures are significantly associated with several routes. Component 6 of the signatures —associated with the respiratory system — is associated with inhalation and nasal administration. Component 4 — associated with the dermatological system — is also associated with nasal administration and transdermal delivery administration, which is typically known to cause adverse skin reactions. Finally, I found that component 8 — associated with the nervous system — was associated with oral administration (I note, however, that this association could be due to a large number of nervous system drugs in our dataset).

4 Conclusions and Discussion

I presented a novel framework for predicting the frequency of drug side effects. My model learns a low-dimensional representation of drugs and side effects that I called *signatures*. I showed that these signatures encode meaningful biological information about drug activity at the anatomical and molecular level. I envision the use of my system by safety professionals during pre-and post-marketing drug development: in the premarketing phase, to assist in the design of clinical trials by generating a hypothesis on the frequencies of certain side effects; in the post-marketing phase, to complement surveillance reporting systems in the early discovery of severe side effects of very rare occurrence — this requires an analysis of the low scores predicted by my system. Furthermore, my method can be used by health policymakers and regulatory agencies when assessing the safety of candidate drugs.

An innovative technical aspect of my matrix decomposition algorithm is that it can take into account different level of uncertainty associated with the data. The underlying as-

sumption of my model is that the matrix is fully, rather than partially, observed, but that a well-defined set of entries are noisy – in our problem, these are the zeros, corresponding to unobserved drug side effect associations. Earlier matrix decomposition methods, such as singular value decomposition (SVD), or non-negative matrix factorisation (NMF)[29], did not explicitly account for different levels of uncertainty in the data. My multiplicative learning rule is simple, computationally efficient and has theoretical guarantees of convergence. I envisage its use for problems that can be framed as predicting the presence or absence of relationships between pairs of elements where only some entries in the training data are noisy. This is the case for many problems in different areas of biology, chemistry and medicine – for example, for the problems of predicting protein-RNA interaction[5] and disease gene prediction[41] – as well as in social networks analysis, and recommendation systems for e-commerce.

To the best of my knowledge, this is the first method that can predict the frequencies of drug side effects in the population. Other methods had been proposed earlier that were able to predict the probability of a given drug side effect association, but these probabilities are only weakly correlated with the side effect frequencies, and therefore cannot be used effectively for the prediction of frequency classes. I verified this, for example, for the scores obtained by the predictive pharmaco-safety networks (PPN-NET)[6] – their Pearson correlation with the frequency of drug side effects is $\rho = 0.08$ (significance, $p < 1.28 \times 10^{-6}$ — Fig. 1.13 compares the scores obtained by PPN-NET and by my method.

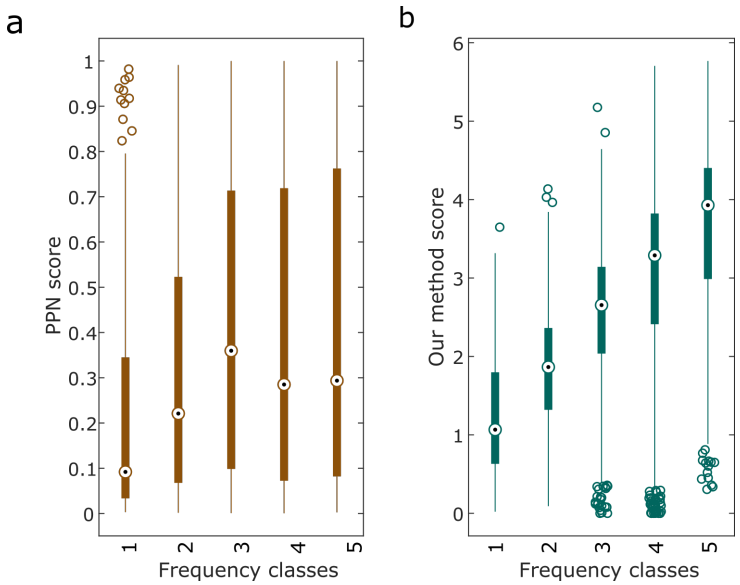

Figure 1.13: Predicted scores by Predictive Pharmacosafety Networks (PPNs) and my method in the held-out test set. (a) Predicted scores by PPNs are weakly correlated to the frequency of the side effect in the population (Pearson correlation $\rho = 0.08, p < 1.28 \times 10^{-06}$); (b) Predicted scores by my method are more strongly correlated to the frequency of the side effect in the population (Pearson correlation $\rho = 0.474, p < 2.39 \times 10^{-209}$).

The seminal work of Campillos et al.[88] had shown that drug side effects are predictive of drug targets. More recently, Wang et al.[9] had shown that drug side effects are predictive of therapeutic indications. Therefore, one interesting question was whether my model's signatures, learned from side effect data, were associated with molecular and clinical drug activity. I found that drugs with similar signatures were more likely to share a protein target and to belong to the same anatomical, therapeutic, pharmacological and chemical category. Intriguingly, the non-negative constraints in my model favour a "parts-based representation"[29] of the signatures: the drug activity becomes explainable in terms of the drug effects on the different "parts" of the human anatomical systems. This representation of drug activity makes sense in the context of network pharmacology[17]: the observed side effect patterns for a given drug can be explained by a combination of perturbations in distinct or-

gan system networks. Figure 1.14 shows signature components with significant activations for groups of drugs and side effects obtained by top-level terms in ATC and MedDRA hierarchies (anatomical level). Specific components of the signatures are strongly associated with specific anatomical classes. The ability of my model to capture the parts-based representation that reflects the human anatomical systems is quite remarkable as signatures are learned from noisy information about a few drug side effects associations.

These reproducible drug and side effect signatures (summarised in Tables 1.1 and 1.2 and Figure 1.14) provide insights into how my model works. The signatures encode biological information about the drug and side effects interplay, and these relationships can be exploited to formulate a biological hypothesis for researchers. The signatures can also be useful in other pharmacological research, such as in the study of frequencies of adverse drug combinations.

There are limitations and biases in public databases of drug side effects. For instance, I observed that the frequencies of side effects are biased towards frequent ones (Fig. 1.1b). Recent reports also indicate that clinical trials are biased towards male gender and certain ethnicity groups: 86% of clinical trials cohorts were Caucasian-dominated in 2014[103]. Numerous previous research also reported divergent drug responses in subjects with a different genetic background[104]. I envision extending my model and the analysis presented here to integrate additional metadata from clinical trials to tailor the prediction for gender- or ethnic-specific intervention groups.

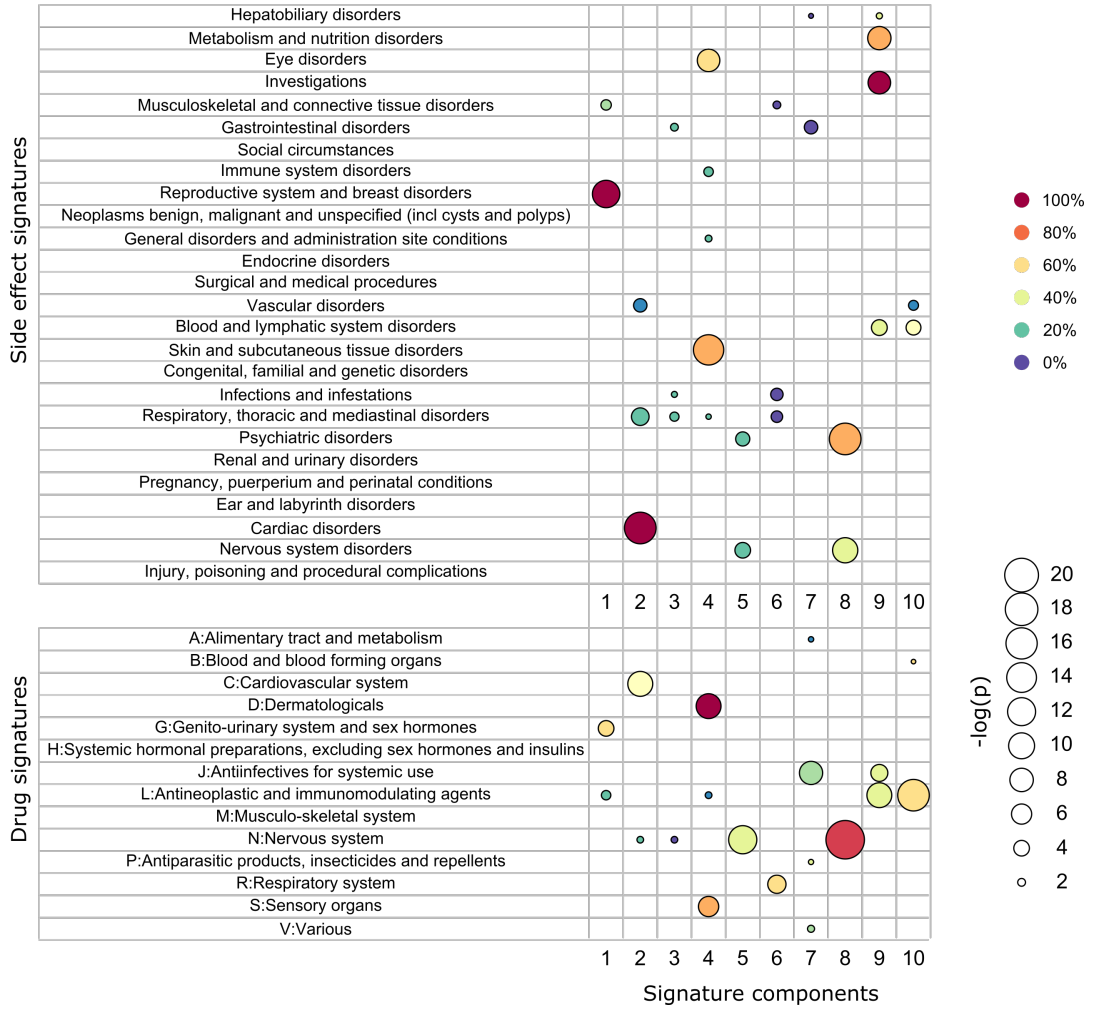

Figure 1.14: Summary of significant activations of drug and side effect signatures per anatomical classes. Drugs were grouped based on their main Anatomical, Therapeutic and Chemical (ATC) classes while side effects were grouped by their System Organ Class (SOC) categories in MedDRA. Only statistically significant associations (One-Tailed Wilcoxon Sum Rank Test with Benjamini-Hochberg adjusted Significance, $p < 0.05$) are shown. The size of the circle represents the significance (p-value), and the colour encodes the effect size of the association — the difference between median in the group compared to the median of all drugs (or side effects).

4.1 Prediction case studies: severe side effects that caused drug withdrawal

A critical question to assess the real applicability of my approach is whether my method can predict the side effects that led to drug withdrawal from the market. Typically, these corre-

spond to severe side effects which are caused by off-target drug effects, which by their nature are unforeseen from the chemistry and pharmacology. These correspond to the Rumsfeld's "unknown unknowns"[‖], that is, there are severe drug side effects that are unknown to the investigator but are also unknown in the chemical or pharmacological literature.

I showcase case studies for eight withdrawn drugs from different pharmaceuticals (see Table 1.3 and Figs. 1.15, 1.16, 1.17, 1.18, 1.19, 1.20, 1.21). To obtain the side effect that caused drug withdrawal, I accessed the online version of DrugBank v5.1.4 and obtained the side effects by manual inspection. Note that in most cases, the cause of withdrawal is a general term, e.g. gastrointestinal disorders, which maps into an entire top MedDRA side effect category. Therefore, due to the lack of a systematic database on the specific side effect(s) that led to drug withdrawal, I analysed the top MedDRA categories in each case. The question is whether my model's prediction, that relies on data from clinical trials, can predict the severe side effects that led to drug withdrawal. These predictions can be helpful as initial evidence to study off-target mechanism of these drugs.

In general, I found that my model was able to predict severe side effects belonging to the MedDRA category associated to the cause of withdrawal. Let me analyse few examples in detail. Fig. 1.15 shows the predicted scores for the drug Alosetron, originally indicated for diarrhoea-predominant irritable bowel syndrome in women. Alesotron was withdrawn from the market due to severe gastrointestinal disorders such as ischemic colitis, constipation and severely obstructed or ruptured vowels[106]. Interestingly, many gastrointestinal dis-

[‖]Rumsfeld is an American politician who once stated: "Reports that say that something hasn't happened are always interesting to me, because as we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say, we know there are some things we do not know. But there are also unknown unknowns—the ones we don't know we don't know. And if one looks throughout the history of our country and other free countries, it is the latter category that tends to be the difficult ones.". The concept of "known unknowns" has been also used in analytical chemistry[105].

orders shows up in my model predictions either as frequent or infrequent. In another case, I found that my model was able to retrieved the exact side effect term that was reported as the cause of drug withdrawal. This is the case of Sitaxentan (see Fig. 1.16), originally indicated for pulmonary arterial hypertension. Sitaxentan was withdrawn due to hepatotoxicity, which was in fact predicted as a `rare` side effect by my model. It has been reported that four deaths and one case of liver transplantation has been observed amongst 2,000 patients treated worldwide[107]; this observation counts precisely as a rare side effect for the drug (4/2,000).

Another example is Pergolide (see Fig. 1.20), originally indicated for Parkinson disease, which was withdrawn from the market due to heart valve damage. In this case, we can see that, for instance, cardiac failure is predicted as frequent. Interestingly, cardiac failure is typically a complication of heart valve damage. This example illustrates how my model might not necessarily predict the causative side effect but rather other related side effects that are a consequence (or complications) of the cause. Even in this scenario, multiple evidence might help to undercover the underlying cause.

Drug name	Indication	Severe side effect	Year of withdrawal	Company
Alosetron	severe diarrhoea-predominant irritable bowel syndrome (IBS) in women	gastrointestinal side effects	2000	Prometheus Lab Inc.
Sitaxentan	pulmonary arterial hypertension	hepatotoxicity	2010	Pfizer
Rofecoxib	osteoarthritis rheumatoid arthritis	heart attack and stroke	2004	Merck & Co.
Colestilan	hyperphosphataemia	gastrointestinal disorders (haemorrhage)	2013	Mitsubishi Tanabe Pharma
Valdecoxib	osteoarthritis and dysmenorrhoea	skin disorders	2005	G. D. Seatle & Co.
Pergolide	Parkinson disease	heart valve damage	2007	Boehringer Ingelheim
Tegaserod	irritable bowel syndrome with constipation (ISB-C)	cardiac disorders	2007	Sloan Pharmaceuticals

Table 1.3: Case studies of drugs that have been withdrawn from the market due to severe side effects.

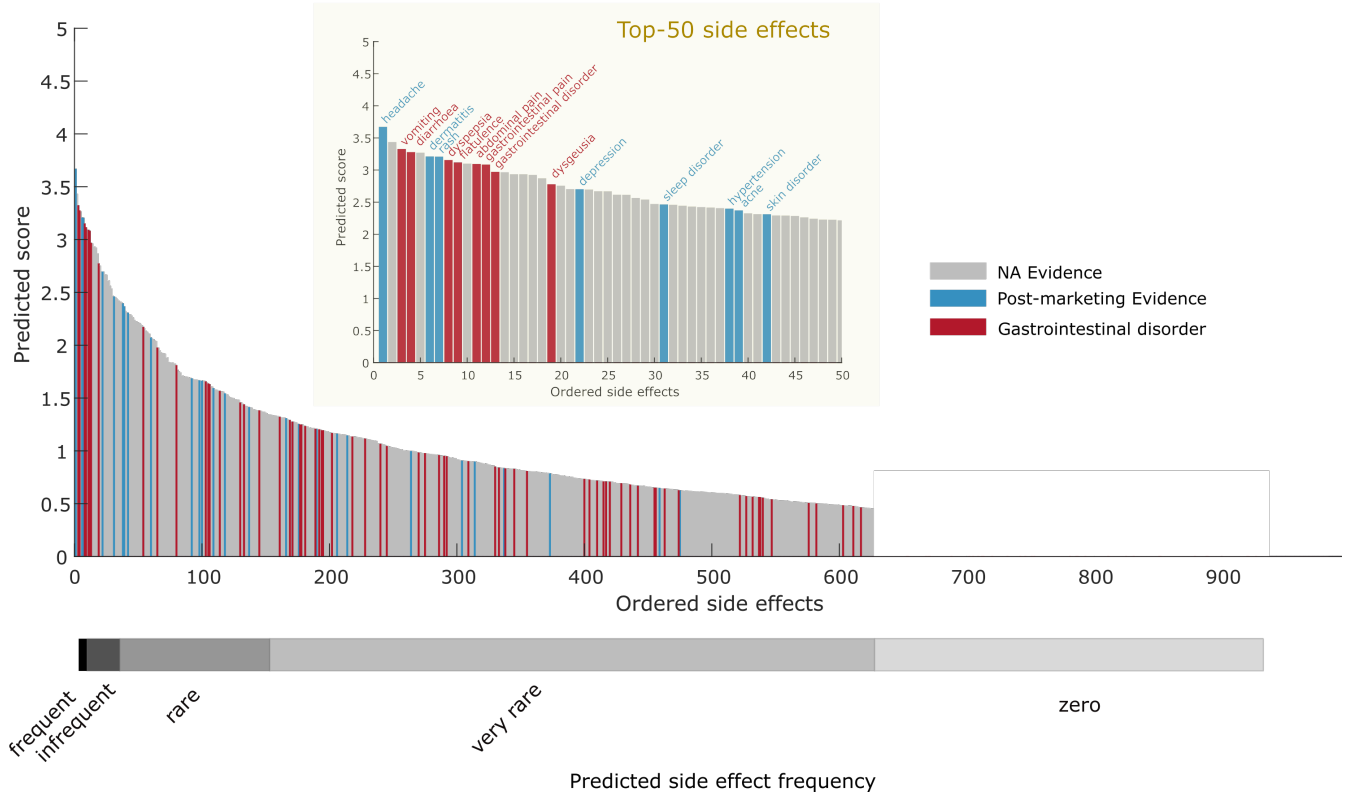

Figure 1.15: Barplot of the predicted side effect scores for the withdrawn drug Alosetron. Alosetron was withdrawn due to severe gastrointestinal adverse reactions. The y-axis shows the predicted score by my matrix decomposition model while the x-axis shows all the 994 side effects in our dataset. The horizontal bar shows the assigned class according to my MLE model. *Inset*. Top-50 side effects. Colours are shown for all the side effects belonging to the top MedDRA category that correspond to the cause of withdrawal (in red), and other post-marketing evidence (in blue).

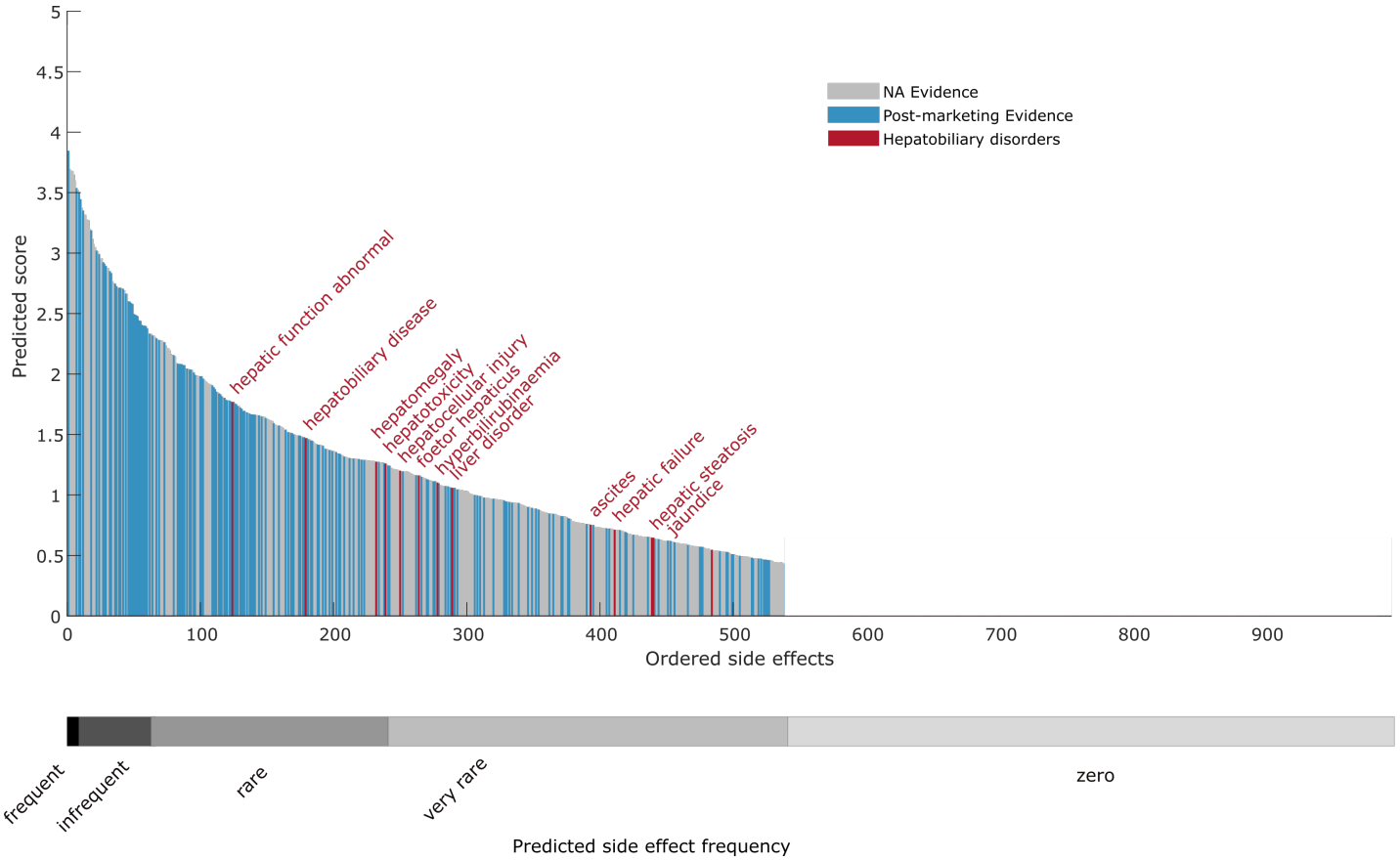

Figure 1.16: Barplot of the predicted side effect scores for the withdrawn drug Sitaxentan. The y-axis shows the predicted score by my matrix decomposition model while the x-axis shows all the 994 side effects in our dataset. The horizontal bar shows the assigned class according to my MLE model. *Inset.* Top-50 side effects. Colours are shown for all the side effects belonging to the top MedDRA category that correspond to the cause of withdrawal (in red), and other post-marketing evidence (in blue).

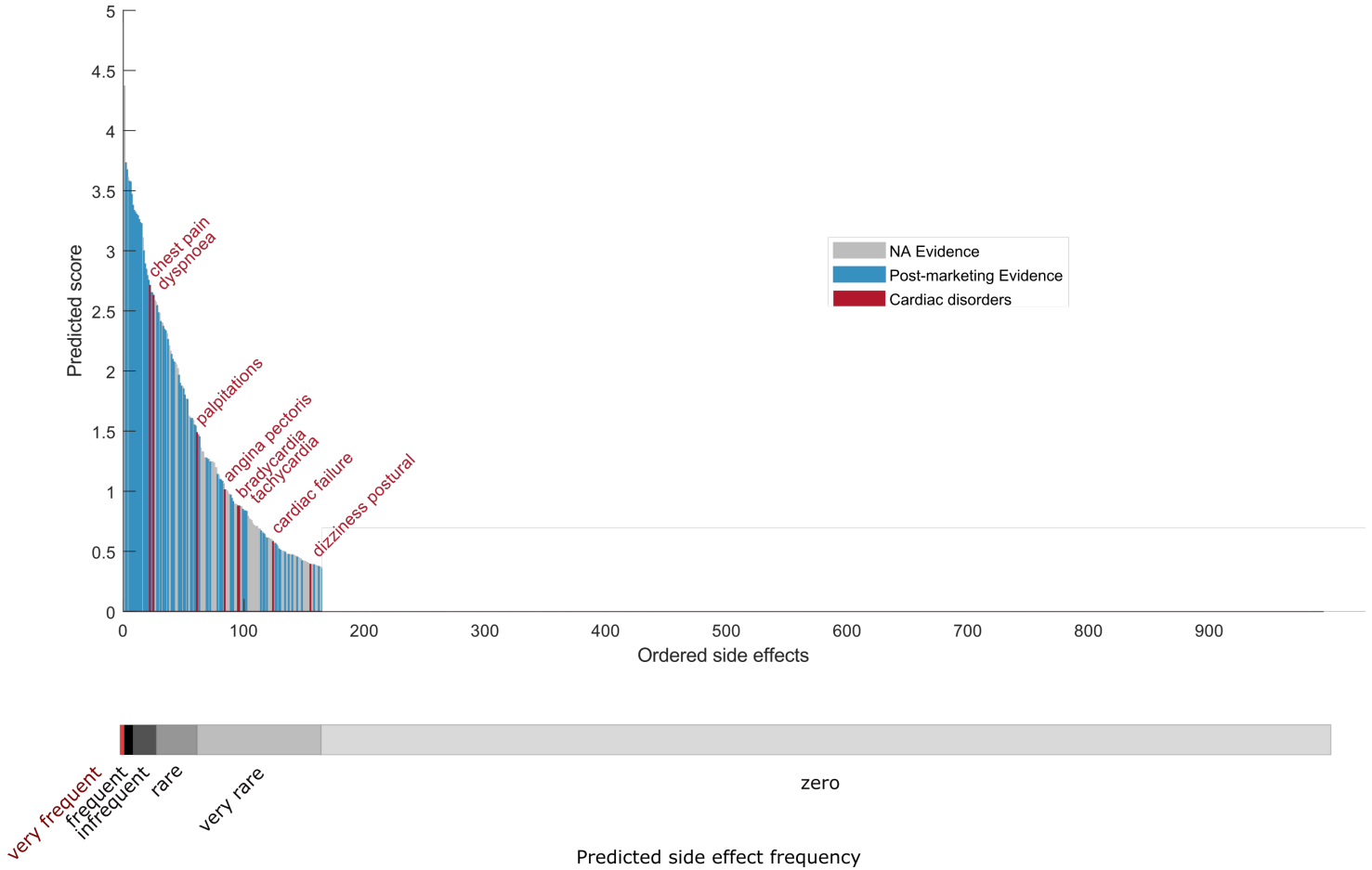

Figure 1.17: Barplot of the predicted side effect scores for the withdrawn drug Rofecoxib. Rofecoxib was withdrawn due to severe cardiac-related adverse reactions. The y-axis shows the predicted score by my matrix decomposition model while the x-axis shows all the 994 side effects in our dataset. The horizontal bar shows the assigned class according to my MLE model. *Inset*. Top-50 side effects. Colours are shown for all the side effects belonging to the top MedDRA category that correspond to the cause of withdrawal (in red), and other post-marketing evidence (in blue).

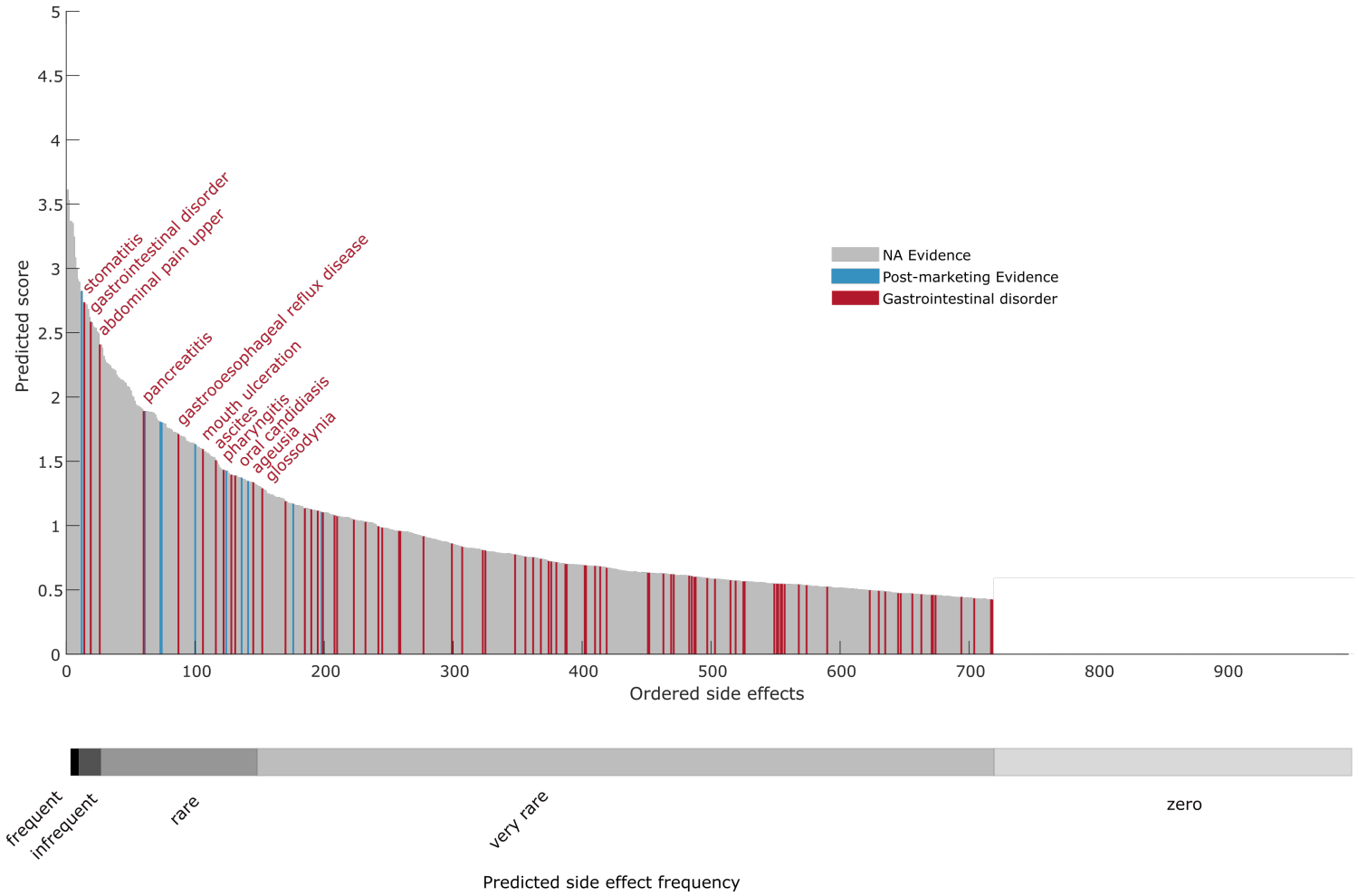

Figure 1.18: Barplot of the predicted side effect scores for the withdrawn drug Colestilan. Colestilan was withdrawn due to severe gastrointestinal adverse reactions. The y-axis shows the predicted score by my matrix decomposition model while the x-axis shows all the 994 side effects in our dataset. The horizontal bar shows the assigned class according to my MLE model. *Inset*. Top-50 side effects. Colours are shown for all the side effects belonging to the top MedDRA category that correspond to the cause of withdrawal (in red), and other post-marketing evidence (in blue).

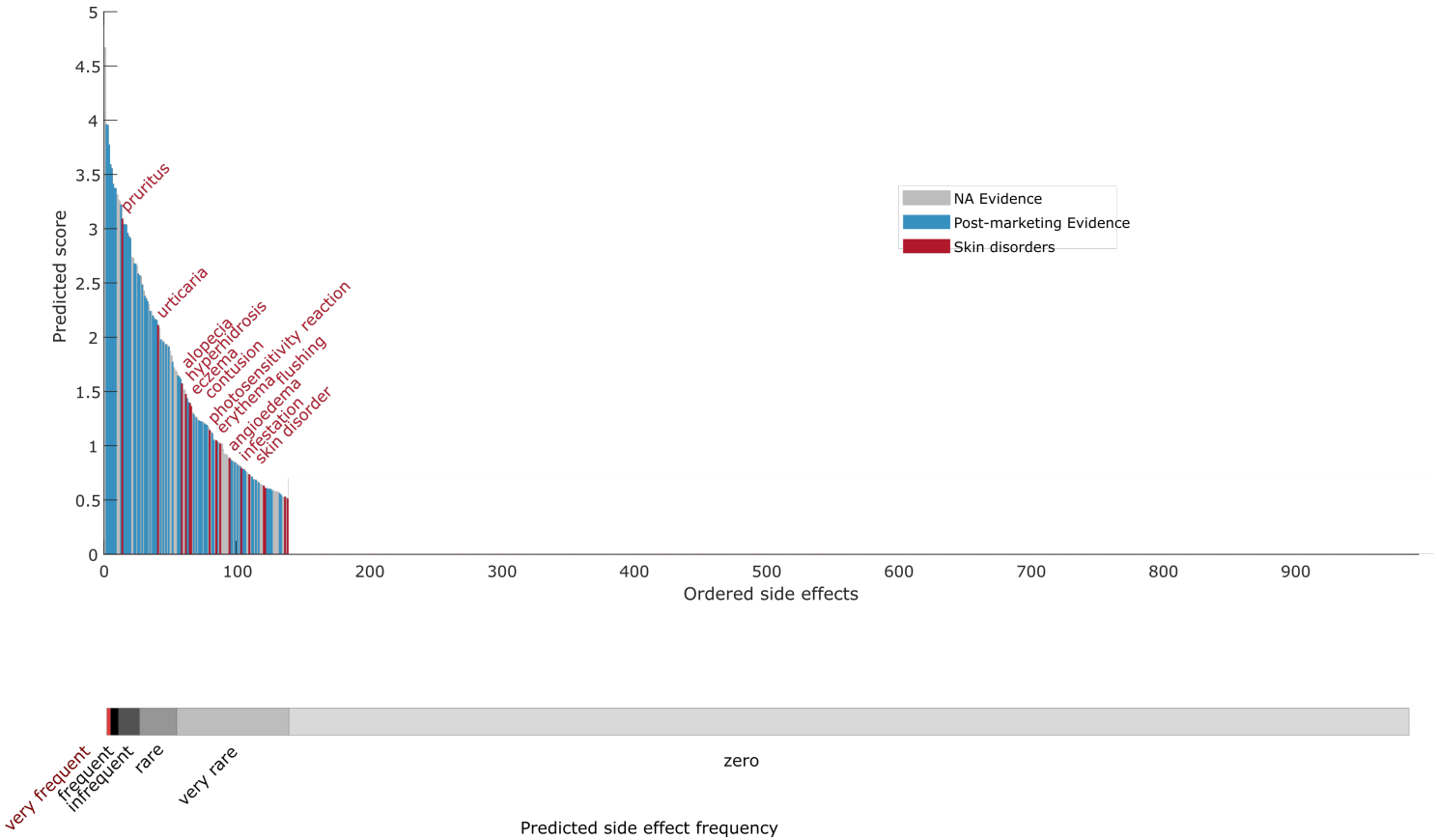

Figure 1.19: Barplot of the predicted side effect scores for the withdrawn drug Valdecoxib. Valdecoxib was withdrawn due to severe skin adverse reactions. The y-axis shows the predicted score by my matrix decomposition model while the x-axis shows all the 994 side effects in our dataset. The horizontal bar shows the assigned class according to my MLE model. *Inset.* Top-50 side effects. Colours are shown for all the side effects belonging to the top MedDRA category that correspond to the cause of withdrawal (in red), and other post-marketing evidence (in blue).

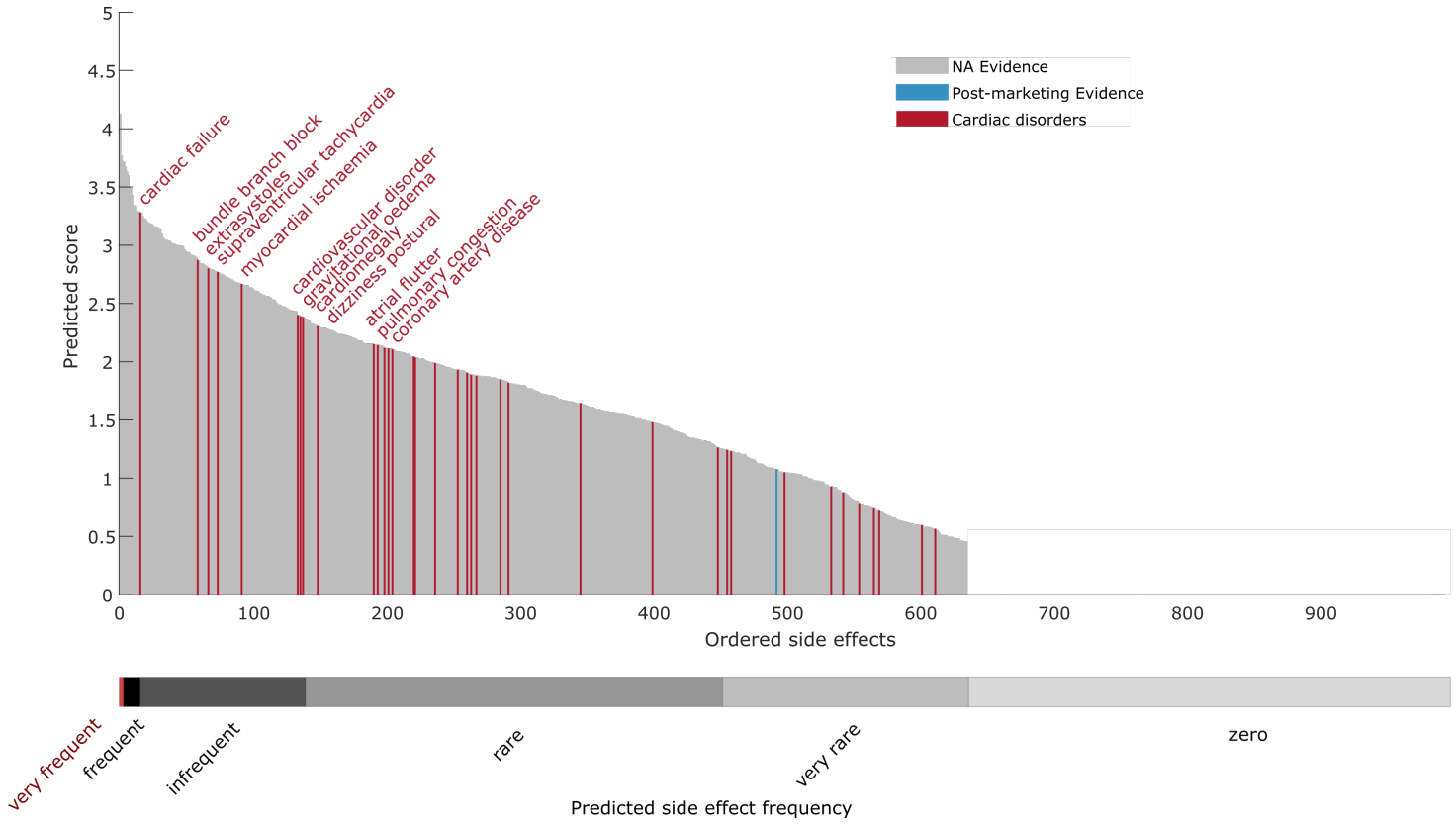

Figure 1.20: Barplot of the predicted side effect scores for the withdrawn drug Pergolide. Pergolide was withdrawn due to cardiovascular adverse reactions. The y-axis shows the predicted score by my matrix decomposition model while the x-axis shows all the 994 side effects in our dataset. The horizontal bar shows the assigned class according to my MLE model. *Inset.* Top-50 side effects. Colours are shown for all the side effects belonging to the top MedDRA category that correspond to the cause of withdrawal (in red), and other post-marketing evidence (in blue).

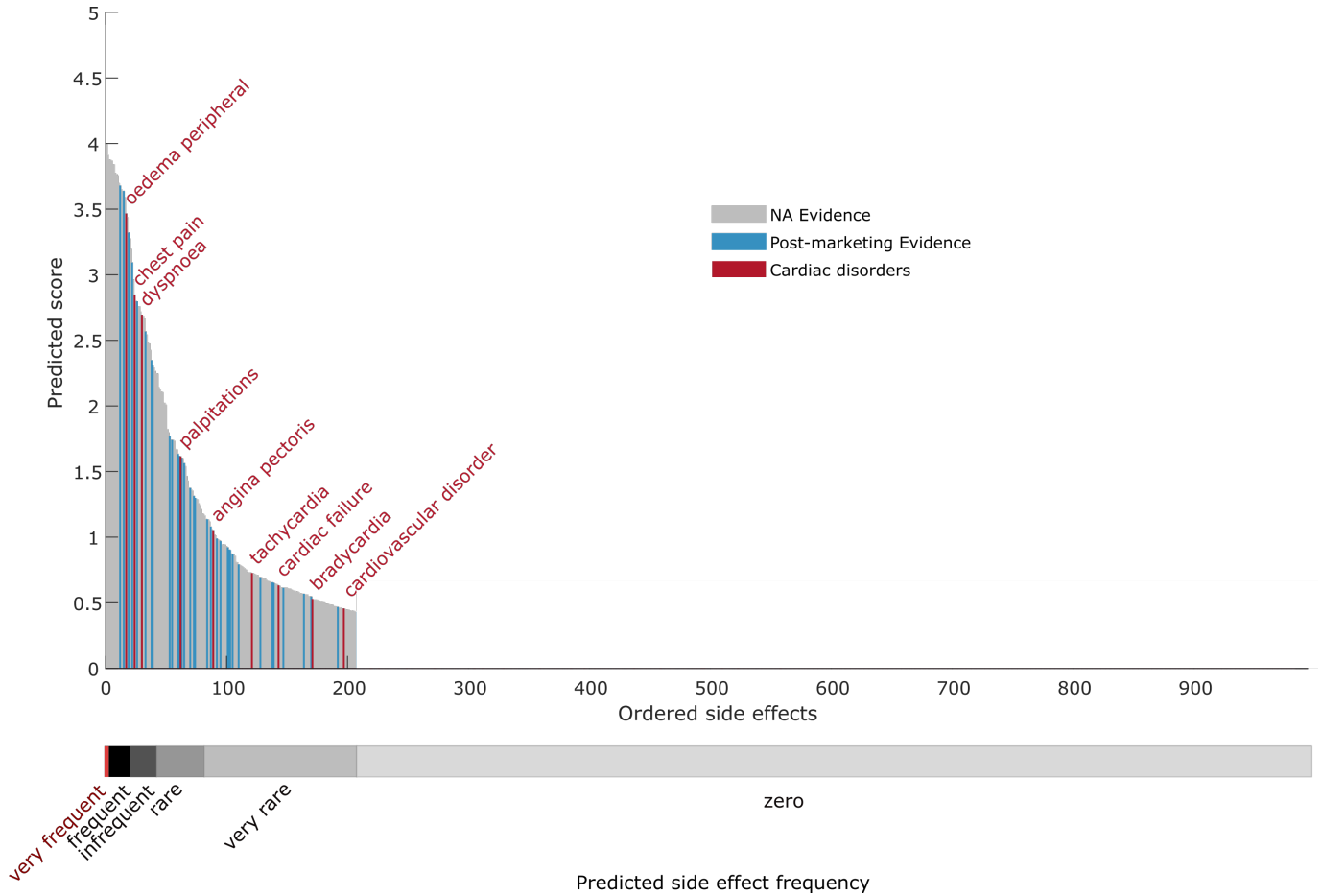

Figure 1.21: Barplot of the predicted side effect scores for the withdrawn drug Tegaserod. Tegaserod was withdrawn due to cardiovascular adverse reactions. The y-axis shows the predicted score by my matrix decomposition model while the x-axis shows all the 994 side effects in our dataset. The horizontal bar shows the assigned class according to my MLE model. *Inset.* Top-50 side effects. Colours are shown for all the side effects belonging to the top MedDRA category that correspond to the cause of withdrawal (in red), and other post-marketing evidence (in blue).

The remedy is worse than the disease.

Francis Bacon (1561-1626)

2

Drug side effect prediction

Much of the current research on drug side effect prediction focuses on predicting the presence or absence of side effects (hereafter, side effect identification or simply drug side effect prediction), ignoring, owing to lack of systematic data and well-defined framework, the frequencies of the side effects. Although I dedicated Chapter 1

to describe a general framework for predicting the frequencies of drug side effects, which generalises over the (binary) side effect identification, it is important to also address the problem of side effect prediction. The reason is twofold. First, drug side effect frequencies can only be predicted for drugs for which at least few frequency associations are available (this is only for about 50% of the drugs in SIDER 4.1[65]). Second, side effect identification methods can make use of additional evidence collected in post-market stages from observational databases — such as spontaneous reports, drug-specific patient registries, administrative claims databases, and electronic health records —, which are continuously monitored for increased of side effect rates due to a given drug.

A wide range of computational approaches have been proposed for predicting drug side effects (for reviews see[63,64]). The common assumption underlying these methods is that there is biological or pharmacological relational information between drugs, e.g., chemical structure, and between side effects, e.g., phenotype similarity, that can be exploited for the prediction task. Current methods typically rely on well-defined heuristics and/or hand-crafted features. For instance, Cami et al.[6] extracts several feature covariates from the bipartite network built by connecting drugs to side effects, including other chemical and taxonomic covariates obtained for drugs and side effects, and then trains a Bernoulli expectation model based on multivariate logistic regression. Bean et al.[31] built a knowledge graph by connecting drugs, side effects, protein targets, and indications and then applied enrichment analysis to predict missing links in the network. Other network-based approaches include random walks and label propagation on side information networks[32,33].

More recently, the drug side effect prediction problem has been framed as a matrix completion task. For instance, Li et al.[35] proposed a low-rank model called inductive matrix

completion, that integrates side information using kernel matrices of drugs and side effects. Similarly, Zhang et al.[34] proposed a low-rank model that incorporates smoothness constraints on graphs built from drug side information.

In this chapter, I propose two methods to tackle the side effect prediction problem. The first approach, which I discuss briefly in section 2.1, is a low-rank model with $\mathcal{L}_2$ regularisation. The second approach, which is the main focus of this chapter, is a self-representation learning model that is able to overcome the limitations of heuristic-based approaches and extend the expressiveness of low-rank factorisation models for matrix completion. My model builds upon the recent development of high-rank matrix completion based on self-expressive models (SEM)[8], as well as the recent trend of deep learning on graphs[25,26,24]. I propose a geometric SEM model that integrates relational inductive bias about drugs (and side effects) in the form of drug (and side effect) similarity graphs. Extensive experiments on a standard benchmark dataset show that my method outperform existing state-of-the-art approaches in drug side effect prediction, and that the inclusion of relational inductive bias significantly improves the performance while enhancing model interpretability.

1 Related work

In this section, I introduce several state-of-the-art methods for drug side effect prediction. One limitation that I encounter on the literature of side effect prediction is the lack of systematic comparison between state-of-the-art methods on the same benchmark dataset. To ameliorate this situation, I will compare several state-of-the-art methods that I present in this section to my proposed models (the comparison is presented in section 3).

1.1 Predictive Pharmacosafety Networks (PPNs)

PPNs[6] is based on the idea that the connectivity patterns in the bipartite network that connects drugs to side effects are important for the prediction of missing links in the network. PPNs model the binary response variable $X_{uj}, u \in \{1, ..., n\}, j \in \{1, ..., m\}$, for n drugs and m side effects, denoting the presence or absence of drug side effect associations. Using Logistic Regression (LR), PPNs modelled this response as a Bernoulli random variable with the following expectation:

$$\mathbb{E}[X_{uj}] = \frac{1}{1 + \exp\left(-\sum_s \beta_s Z_s(u,j)\right)} \tag{2.1}$$

Here, β_s denotes the model parameter and Z_s the model covariate. PPNs considers network covariates, that are extracted from the observed drug-side effect network but not on from other drug or side effect attributes, as well as other covariates that can be extracted from side information about drugs or side effects.

Network Covariates

In the following, let u represent a drug node, and j a side effect node. Then, the degree covariates are defined as follow:

$$\begin{aligned} Z_1(u,j) &= \deg(u) \times \deg(j) \\ Z_2(u,j) &= |\deg(u) - \deg(j)| \\ Z_3(u,j) &= \deg(u) + \deg(j) \\ Z_4(u,j) &= \frac{\deg(u)}{\deg(j)} \end{aligned} \tag{2.2}$$

Here, $\deg(u)$ denotes the degree of node u. The degree product covariate, $Z_1(u,j)$, aims to capture preferential attachment among high-degree drugs and side effects. The degree difference covariate, $Z_2(u,j)$, aims to capture assortativity, i.e. whether high-degree drugs connects to high-degree side effects or to small-degree side effects.

Further, distance covariates accounts for the set of neighbours of drug u, that we denoted as $\mathcal{N}(u)$, and the set of neighbours of side effect j, denoted as $\mathcal{N}(j)$. Let also $J(j,k)$ denotes the Jaccard similarity between the neighbours sets $\mathcal{N}(j)$ and $\mathcal{N}(k)$, which is defined as follow:

$$J(j,k) = \frac{|\mathcal{N}(j) \cap \mathcal{N}(k)|}{|\mathcal{N}(j) \cup \mathcal{N}(k)|} \tag{2.3}$$

The Jaccard-based covariates quantify structural similarity between drug pairs and side

effect pairs:

$$
\begin{aligned}
Z_5(u,j) &= \max_{k \in \mathcal{N}(u)-\{j\}} \{J(j,k)\} \\
Z_6(u,j) &= \max_{k \in \mathcal{N}(j)-\{u\}} \{J(u,k)\}
\end{aligned}
\tag{2.4}
$$

Finally, Jaccard-based predictors based on Kullback-Leibler divergence ($\mathcal{K}_{\mathcal{L}}$) between the overall distribution of similarities between a drug ($\bar{D}_{se}$) and the drugs in its local neighbourhood ($D_{se}(u,j)$) or between side effects ($\bar{D}_{se}$) and side effects in its neighbourhood ($D_{se}(u,j)$) are defined as follow:

$$
\begin{aligned}
Z_7(u,j) &= \mathcal{K}_{\mathcal{L}}(D_{se}(u,j), \bar{D}_{se}) \\
Z_8(u,j) &= \mathcal{K}_{\mathcal{L}}(D_{drug}(u,j), \bar{D}_{drug})
\end{aligned}
\tag{2.5}
$$

Intrinsic (and other) covariates

Given additional information about drugs (or side effects), these can also be included in the prediction. Intrinsic covariates are those that are defined based on the intrinsic properties of drugs, i.e. chemical structure. For instance, let $d_{chem}(u,k)$ represent a chemical distance* between two given drugs u and k, then, the intrinsic covariate is defined as:

$$
Z_9(u,j) = \min_{k \in \mathcal{N}(j)-\{i\}} \{d_{chem}(u,k)\} \tag{2.6}
$$

*There are many distances at molecular level, we used the well-known 2D Tanimoto chemical distance[108].

Other covariates for drugs (or side effects) can also be integrated in a similar fashion, by considering other drug information — for example, drug-drug interactions, drug targets, or drug indications.

1.2 Inductive Matrix Completion (IMC)

IMC[35] is a low-rank model that integrates drug and side effect attributes. For the binary matrix $X \in \mathbb{R}^{n \times m}$ for n drugs and m side effects, let $K_d \in \mathbb{R}^{n \times n}$ represent the kernel similarity matrix for drugs and, similarly, let $K_a \in \mathbb{R}^{m \times m}$ represent the kernel similarity matrix for side effects. When only drug or side effect attributes are available, the kernel matrices are built using similarity measures, e.g. Tanimoto chemical similarity from chemical attributes. IMC aims to approximate X by:

$$X \simeq K_d W H K_a \tag{2.7}$$

By minimising the following loss function:

$$\min_{W,H} \mathcal{L}(W,H) = \frac{1}{2}\|\Omega \circ (X - K_d W H K_a)\|_F^2 + \frac{\lambda}{2}(\|W\|_F^2 + \|H\|_F^2) \tag{2.8}$$

Here, Ω represents a projection over the observed entries in X, $\circ$ is the element-wise product between matrices, $W \in \mathbb{R}^{n \times k}$ is the drug latent representation matrix and $H \in \mathbb{R}^{k \times m}$ is the side effect latent representation matrix. The second term in Eq. 2.8 is a $\mathcal{L}_2$ regularisation term applied to the latent matrices W and H, with a positive penalty $\lambda > 0$.

Gradient descent-based methods are required to approximate the solution. Thus, the

derivatives of (2.8) are:

$$\frac{\partial \mathcal{L}(W,H)}{\partial W} = -K_d(X - \Omega \circ (K_d W H K_a))K_a H^T + \lambda W \tag{2.9}$$

$$\frac{\partial \mathcal{L}(W,H)}{\partial H} = -W^T K_d(X - \Omega \circ (K_d W H K_a))K_a + \lambda H \tag{2.10}$$

1.3 Feature-derived graph regularised matrix factorisation (FGRMF)

The FGRMF[34] integrates several low-rank models, one per side information graph, by using a logistic regression model. Each low-rank model is optimised independently and then combined. For the binary matrix $X \in \mathbb{R}^{n \times m}$ for n drugs and m side effects, FGRMF first minimises the following loss:

$$\min_{W,H} \mathcal{L}(W,H) = \underbrace{\frac{1}{2}\|X - WH\|_F^2}_{\text{low-rank model}} + \underbrace{\frac{\lambda}{2}(\|W\|_F^2 + \|H\|_F^2)}_{\text{regularisation}} + \underbrace{\frac{\alpha}{2}\|W\|_{\mathcal{D},\mathcal{G}}^2}_{\text{smoothness}} \tag{2.11}$$

where $\|.\|_F$ denotes the Frobenius norm, and $\|C\|_{\mathcal{D},\mathcal{G}}$ the Dirichlet norm on the graph $\mathcal{G} = (\{1,\ldots,n\}, \mathcal{E}, G)$, i.e. the weighted undirected graph with edge weights $g_{ij} > 0$ if $(i,j) \in \mathcal{E}$ and zero otherwise, representing the similarities between drugs.

The second step of FGRMF is to combined the solution from multiple graphs $\{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_p\}$, that gives different approximations to X, $\{\hat{X}_1, \hat{X}_2, ..., \hat{X}_p\}$ as follow:

$$h_\theta = \frac{1}{1 + \exp\left(-(\theta_0 + \sum_{i=1}^{p} \theta_i \hat{X}_i\right)} \tag{2.12}$$

where the weights θ_j are learned for each independent model. h_θ is used as a final predic-

tion model.

1.4 Label propagation using consistency method (LP)

In the seminal work of Atias et al.[32], they proposed a Logistic Function combination of two models to predict drug side effects: a network-based label propagation model (consistency method[109]) and a Canonical Correlation Analysis (CCA)-based method. Of these, only the label propagation can be adapted to our setting, as we focus on predicting side effects for drugs for which few side effects are already available. CCA-based methods, such as Sparse CCA[110], have been proposed for drug side effect prediction but for a different setting: to predict side effects for compounds by taking as input, an encoding of the chemical structure of a compound[111] — these methods assume not known side effect for the compound in training. The main difference between these two problems is that the first focuses on predicting new side effect for drugs in the post-marketing period, while the second focuses on predicting side effects for a lead compound before marketing. In this chapter, we focus on the first problem.

Label propagation or network propagation is a semi-supervised learning model that heavily relies on a network structure to make predictions. It has two main components: the network and the initial labels. The initial labels are superimposed on the nodes on the network such that the knowledge of the initial labels is "spread" or diffuse to the neighbouring nodes in the network. Label propagation can be formulated either as an iterative sequence of matrix multiplication similar to random walks on networks, or as a regularisation frame-

work. Here we show the latter, which minimises the following objective function[109]:

$$\arg\min_F \underbrace{\frac{1}{2}\sum_{i,j=1}^{m} W_{ij}\left\|\frac{1}{\sqrt{D_{ii}}}F_i - \frac{1}{\sqrt{D_{jj}}}F_j\right\|^2}_{\text{smoothness}} + \underbrace{\frac{\mu}{2}\sum_{i=1}^{n}\|F_i - Y_i\|^2}_{\text{initial labels}} \tag{2.13}$$

where $\mu > 0$ is the regularisation parameter, $F \in \mathbb{R}^{m\times n}$ is the learned matrix of n drugs (columns) and m side effects (rows), $Y \in \mathbb{R}^{m\times n}$ is the matrix of initial labels containing the binary drug side effect associations, and $W \in \mathbb{R}^{m\times m}$ is the matrix of weights of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$ (side effects) and edges $\mathcal{E}$ weighted by W. In addition, $\text{diag}(W) = 0$ is set to avoid self-reinforcement. D is a diagonal matrix with its (i,i)-element equal to the sum of the i-th row of W.

In this case, the network is built from side effect similarities and the initial labels are the known. Label propagation is based on the assumption that drugs tend to cause similar side effect phenotypes, as reveal by the network structure of side effects phenotype similarity. Label propagation have been widely used to amplify genetic signals from phenotypic similarities for predicting disease causing genes[112,41].

The label propagation algorithm consists in the following steps[32,109]:

1. Build he affinity matrix W using the Jaccard similarity between side effects based on the set of drugs two side effects share. Then set $W_{i,i} = 0$.

2. Construct the matrix $S = D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$.

3. Compute the scores $F = (I - \alpha S)^{-1}Y$, which is the globally optimal solution of Eq. 2.13, where $I \in \mathbb{R}^{n\times n}$ is the identity matrix and $\alpha = (1+\mu)^{-1}$ is a parameter that trades between the two constraints in Eq. 2.13.

1.5 An additional baseline: Side effect popularity (TopPop)

My analysis of the distribution of side effects in Chapter 1 revealed that side effects follows a long-tail distribution, where the majority of the side effects are associated with only few drugs. This characteristic of the dataset is not unique to drug side effects but it have been observed, for instance, also in movie datasets (as I showed in Fig. 1.2). Cremonesi et al.[20] used movie popularity as a baseline to assess the performance of movie recommendation system algorithms. I adopted the same baseline as comparison. For our binary data matrix $X \in \mathbb{R}^{n \times m}$ for n drugs and m side effects, TopPop assigns scores to drug side effect pairs (i, j) as follow:

$$\text{TopPop}_{ij} = \sum_{i}^{n} X_{ij} \tag{2.14}$$

2 Proposed models

In this section, I introduced my two proposed models. The first is a low-rank model while the second is a high-rank model.

2.1 Regularised low-rank matrix factorisation (MF)

The MF model[36] is based on the assumption that the binary matrix $X \in \mathbb{R}^{n \times m}$ for n drugs and m side effects has a low-rank $k \ll min\{m, n\}$, such that X can be expressed as the product of two rank-k matrices, as follow:

$$X \approx PQ \tag{2.15}$$

where $P \in \mathbb{R}^{n \times k}$ and $Q \in \mathbb{R}^{k \times m}$. This model amounts to assign a low-dimensional feature vector to each drug and a low-dimensional feature vector to each side effect such that the dot-product between the features vectors is related to the probability that a drug is associated to a side effect. The rank of X is k — the number of features of each drug and side effect. The matrices P and Q are the solution of the following optimisation problem:

$$\min_{P,Q} \mathcal{L}(P,Q) = \underbrace{\frac{1}{2}\|X - PQ\|_F^2}_{\text{low-rank model}} + \underbrace{\frac{\lambda}{2}(\|P\|_F^2 + \|Q\|_F^2)}_{\text{regularisation}} \tag{2.16}$$

where $\|.\|_F$ is the Frobenius, and the term $\lambda(\|P\|_F^2 + \|Q\|_F^2)$ is a $\mathcal{L}_2$ regularisation term with penalty $\lambda > 0$, which is added to the objective function in order to prevent overfitting.

The problem is non-convex in both P and Q. Gradient descent-based optimisation methods are required to approximate the solution. Thus, the derivatives of (2.16) are:

$$\frac{\partial \mathcal{L}(P,Q)}{\partial P} = -(X - PQ)Q^T + \lambda P \tag{2.17}$$

$$\frac{\partial \mathcal{L}(P,Q)}{\partial Q} = -P^T(X - PQ) + \lambda Q \tag{2.18}$$

I optimise Eq. 2.16 using conjugate gradient descent (see details in section 5).

2.2 Geometric sparse matrix completion model (GSMC)

Self-expressive models

The goal of self-expressive models (SEM) is to represent datapoints, i.e., drugs, approximately as a linear combination of a small number of other datapoints. Proposed as a frame-

work for simultaneously clustering and completing high-dimensional data lying over the union of low-dimensional subspaces[113,9], these models can effectively generalise standard low-rank matrix completion models.

Let $X \in \mathbb{R}^{n \times m}$ be the data matrix (each column is a datapoint) and let $C \in \mathbb{R}^{m \times m}$ be the coefficient matrix (each column is a coefficient vector). The goal of self-expressive model is to learn a matrix C such that $X \simeq XC$ where C is sparse according to some sparsity function and $\text{diag}(C) = 0$[113,9]. Observe that the last constraint is needed to prevent the trivial solution of representing each datapoint with itself ($C = I$).

The regularisation framework

Let me denote our drug side effect matrix for n drugs and m side effects with the binary matrix $X \in \mathbb{R}^{n \times m}$ where $X_{ij} = 1$ if drug i is associated with side effect j, or 0 if the association is unreported. GSMC aims at *learning* two sparse zero-diagonal self-representation matrices, one for the drugs $R \in \mathbb{R}^{n \times n}$ and one for the side effects $C \in \mathbb{R}^{m \times m}$. The data matrix X is then approximated by:

$$\hat{X} \simeq pXC + (1 - p)RX \tag{2.19}$$

where $p \in [0, 1]$ is a parameter that controls the balance between the row (drug) and column (side effect) contributions. In the sequel, I shall refer to the first part of the GSMC model XC, as GSMC-c, and to the second part, RX, as GSMC-r.

To this end, two cost functions, $\mathcal{Q}_c(C)$ and $\mathcal{Q}_r(R)$, that takes into account the relational inductive prior between datapoints — the graph network — for drugs and side effects, are

minimised with respect to C and R, respectively:

$$\min_{C\geq 0} \underbrace{\frac{1}{2}\|X - XC\|_F^2}_{\text{self-representation}} + \underbrace{\frac{\beta^c}{2}\|C\|_F^2 + \lambda^c\|C\|_1}_{\text{sparsity}} + \underbrace{\frac{1}{2}\sum_j^P \alpha_j^c \|C\|_{\mathcal{D},\mathcal{G}_j^c}^2}_{\text{smoothness}} + \underbrace{\gamma\,\mathrm{Tr}(C)}_{\text{null diagonal}} \tag{2.20}$$

$$\min_{R\geq 0} \underbrace{\frac{1}{2}\|X - RX\|_F^2}_{\text{self-representation}} + \underbrace{\frac{\beta^r}{2}\|R\|_F^2 + \lambda^r\|R\|_1}_{\text{sparsity}} + \underbrace{\frac{1}{2}\sum_j^Q \alpha_j^r \|R\|_{\mathcal{D},\mathcal{G}_j^r}^2}_{\text{smoothness}} + \underbrace{\gamma\,\mathrm{Tr}(R)}_{\text{null diagonal}} \tag{2.21}$$

where $\|.\|_F$ denoting the Frobenius norm, $\|.\|_{\mathcal{D},\mathcal{G}}$ the Dirichlet norm of the graph $\mathcal{G}$. Here $\mathcal{G}^c = (\{1,\ldots,m\},\mathcal{E}^c,G^c)$ denote the weighted undirected graph with edge weights $g_{ij}^c > 0$ of $(i,j) \in \mathcal{E}^c$ and zero otherwise, representing the similarities between side effects. Let also $\mathcal{G}^r = (\{1,\ldots,n\},\mathcal{E}^r,G^r)$ denote the weighted undirected graph with edge weights $g_{ij}^r > 0$ of $(i,j) \in \mathcal{E}^r$ and zero otherwise, representing the similarities between drugs. In the following, I shall provide the rationale behind (2.20) only, as the same applies to (2.21).

The first term in Equation (2.20) is the *self-representation constraint*, which aims at learning a matrix of coefficients C such that XC is a good reconstruction of the original matrix X. The second term is the *sparsity constraint*, which uses the elastic-net regularisation known to impose sparsity and robustness to noise[114,115]. The fourth term is a penalty for diagonal elements aimed at preventing the trivial solution $C = I$ by imposing $\mathrm{diag}(C) = 0$ (together with $C \geq 0$). Typically, $\gamma^c \gg 0$ is used.

My model is called *geometric* due to the third term in Equation (2.21), the *smoothness*

term[85,116,26], which incorporates structure into the sparse coefficient matrix C. This is achieved by adding smoothness priors from multiple weighted graphs that encode side information about the columns. Let me represent one of these graph by its adjacency matrix $G^c \in \mathbb{R}^{m \times m}$ (each node represents a side effect). Ideally, nearby points in G^c should have similar coefficients in C, which can be obtained by minimising:

$$\sum_{i,j} G^c_{i,j} \|c_i - c_j\|^2 = \mathrm{Tr}(C L_{G^c} C^T) = \|C\|^2_{\mathcal{D},\mathcal{G}^c} \tag{2.22}$$

where c_i and c_j represent column vectors of C, $L_{G^c} = D^c - G^c$ is the graph Laplacian, and $D^c = diag(\sum_i G^c_{i,j})$ is a diagonal matrix. By extending this formulation to multiple graphs $\mathcal{G}^c_j, j \in \{1, 2, .., P\}$ we obtain the third term in Equation (2.20):†

$$\sum_{j}^{P} \alpha^c_j \,\mathrm{Tr}(C L_{G^c_j} C^T) = \sum_{j}^{P} \alpha^c_j \|C\|^2_{\mathcal{D},\mathcal{G}^c_j} \tag{2.23}$$

where the constant values $\alpha^c_j > 0, j \in \{1, ..., P\}$ weigh the relative importance of each graph.

Finally, following[29], I impose *non-negative constraints* on C, as these constraints lead to more interpretable model since they allow only for additive combinations.

The multiplicative learning algorithm

To minimise Equations (2.20) and (2.21) subject to the non-negative constraints $R, C \geq 0$, I developed efficient multiplicative algorithms inspired by the diagonally re-scaled principle

†Note that for Equation (2.21), the graphs $\mathcal{G}^r_j$ have a different number of nodes (each node represents a drug) and the Dirichlet norm is applied to the rows of R, i.e. $\|R\|^2_{\mathcal{D},G^r_j} = \mathrm{Tr}(R^T L_{G^r_j} R)$.

of non-negative matrix factorisation[29,117]. The algorithm consists in iteratively applying the following multiplicative update rules:

$$C_{ij} \leftarrow C_{ij} \frac{(X^\top X + \sum_k^P \alpha_k^c C G_k^c)_{ij}}{(X^T X C + \sum_k^P \alpha_k^c C D_k^c + \beta^c C + \lambda^c + \gamma^c I)_{ij}} \tag{2.24}$$

$$R_{ij} \leftarrow R_{ij} \frac{(X X^\top + \sum_k^Q \alpha_k^r G_k^r R)_{ij}}{(X X^T R + \sum_k^Q \alpha_k^r D_k^r R + \beta^r R + \lambda^r + \gamma^r I)_{ij}} \tag{2.25}$$

In the following, I shall prove that the algorithm in Eq. (2.24) converges to a solution; that the cost function $\mathcal{Q}_c(C)$ is convex, and therefore the solution found is the global optimum; and that the speed of convergence is first-order. Finally I provide a lower bound for the value γ^c. Proofs for Eq. (2.25) are similar and omitted here for brevity.

Lemma 1. *The cost function $\mathcal{Q}_c(C)$ in Equation (2.20) is convex in C.*

Proof Sketch. We need to prove that the Hessian is a positive semi-definite (PSD) matrix. That is, for a non-zero vector $h \in \mathbb{R}^m$ the following condition is met $h^T \nabla^2 \mathcal{Q}_c(C) h \geq 0$. The graph Laplacians are PSD by definition. The remaining terms in the Hessian $(X^\top X + \beta^c)$ are also PSD. Therefore, $\mathcal{Q}_c(C)$ is convex in C. □

Theorem 2 (Convergence). *The cost function $\mathcal{Q}_c(C)$ in Equation (2.20) converges to a global minimum under the multiplicative update rule in (2.24).*

Proof. We need to show that my algorithm satisfies the Karush-Khun-Tucker (KKT) complementary conditions, which are both necessary and sufficient conditions for a global solution point given the convexity of the cost function (lemma 3)[89,90]. KKT require $C_{i,j} \geq 0$ and $(\nabla \mathcal{Q}_c(C))_{ij} C_{ij} = 0$. The first condition holds with non-negative initialisation of C.

For the second condition, the gradient is: $\nabla \mathcal{Q}_c(C) = -X^TX - \sum_j \alpha_j^c CG_j^c + X^TXC + \sum_j \alpha_j^c CD_j^c + \beta^c C + \lambda^c + \gamma^c I$, and according to the second KKT condition, at convergence $C = C^*$ we have $(X^TXC^* + \sum_j \alpha_j^c C^* D_j^c + \beta^c C^* + \lambda^c + \gamma^c I)_{ij} C_{ij}^* - (X^TX + \sum_j \alpha_j^c C^* G_j^c)_{ij} C_{ij}^* = 0$, which is identical to (2.24). That is, the multiplicative rule converges to a global optima. □

Theorem 3 (Rate of convergence). *The multiplicative update rule in (2.24) has a first-order convergence.*

Proof Sketch. Following[89,118], we can represent the updating algorithm as mapping $C^{t+1} = \mathcal{M}(C^t)$ with fixed point $C^* = \mathcal{M}(C^*)$. Then, when C^{t+1} is near C^*, we have $C \simeq \mathcal{M}(C^*) + \nabla\mathcal{M}(C)(C - C^*)$ subject to $C \geq 0$, and thus $\|C^{t+1} - C^*\| \leq \|\nabla\mathcal{M}(C)\| \cdot \|C^t - C^*\|$, with $\|\nabla\mathcal{M}(C)\| \neq 0$ almost surely. That is, the multiplicative update rule is a first-order algorithm. □

Theorem 4 (Lower bounds for the null-diagonal parameter γ^c). *Let $\varepsilon > 0$ be the maximum tolerable value in* $\mathrm{diag}(C)$, *$\sqrt{\sigma}$ the maximum initial value in* $\mathrm{diag}(C)$, *N^c the total number of iterations and $L = \max_i$ diag$(X^\top X)$. Then, $\gamma^c = f(\varepsilon, N^c)$ is bounded by* $(\frac{\sigma^{1/(2N^c)}L}{\varepsilon^{1/N^c}}, \infty)$.

Proof. Assuming that $\gamma^c \gg \max_i \mathrm{diag}(X^TXC + \sum_j \alpha_j^c CD_j^c + \beta^c C + \lambda^c)$ and that $L \gg \max_i diag(\sum_j \alpha_j^c CG_j^c))$, then at the jth iteration, $\varepsilon(j) := \frac{\sqrt{\sigma}L^j}{(\gamma^c)^j}$. At convergence, $j = N^c$, and $\varepsilon = \frac{\sqrt{\sigma}L^{N^c}}{(\gamma^c)^{N^c}}$, from which I can obtain the lower-bound for γ^c. That is, to guarantee at most ε in $\mathrm{diag}(C)$, we need to set a $\gamma^c(\varepsilon, N^c) > \frac{\sigma^{1/(2N^c)}L}{\varepsilon^{1/N^c}}$. The upper bound is obtained when $\varepsilon \to 0$, which causes $\gamma^c(\varepsilon, p) \to \infty$. In practical applications, the upper bound is limited by machine precision. □

The most expensive operation in (2.24) comes from the denominator term $X^T XC$ for which $\mathcal{O}(N^c \times m^3)$ (where N^c is the total number of iterations). The overall complexity can be reduced by pre-computing the constant covariance matrix $X^\top X$ and the linear combination of graphs. A similar reasoning applies to (2.25), giving $\mathcal{O}(N^r \times n^3)$. Algorithm 1 presents a Matlab pseudo-code for solving GSMC-c that follows the NMF implementation guidelines in [91]: (i) $C^{t=0}$ is sample from a uniform distribution in the interval $(0, \sqrt{\sigma}]$; (ii) a small value $\varepsilon \simeq 1 \times 10^{-16}$ is added to the denominator to prevent division by zero. The stopping criteria for the algorithm is (i) when the number of iterations reaches `maxiter` or (ii) when the element-wise change $\delta_C^{(t)}$ between $C^{(t+1)}$ and $C^{(t)}$ is smaller than a predefined tolerance `tolX`, with:

$$\delta_C^{(t)} = \max \left(\frac{|C_{ij}^{(t+1)} - C_{ij}^{(t)}|}{\max_{(i,j)} |C_{ij}^{(t)}| + \varepsilon} \right) \tag{2.26}$$

ALGORITHM 1: GSMC-c

```
Given the parameters α^c ∈ ℝ^a, β^c, λ^c, σ, γ^c > 0 and the graphs G^c of P
 elements in a cell array.
C = rand(m)*sqrt(σ);
I = eye(m);
COV = X'*X;
Dc = zeros(size(C));
Gc = zeros(size(C));
for graph = 1:P do
    Dc = Dc + alpha(graph).*diag(sum(G^c{graph}, 2));
    Gc = Gc + alpha(graph).*G^c{graph};
end
while convergence criterion is not met do
    numer = COV + C*Gc;
    den = COV*C + C*Dc + β^c.*C + λ^c + γ^c. * I +ε;
    C = C .* numer ./ den;
end
```

The algorithm to solve GSMC-r is similar and omitted for brevity. However, note that

algorithm (1) can also be used to solve GSMC-r. This can be understood by considering that the GSMC-r model can be expressed as follow $RX = (X^T R^T)^T = (YA)^T$ where $Y = X^T$ and $A = R^T$ and thus algorithm (1) can be used to solve A in $\hat{Y} \simeq YA$.

3 Experimental Results

3.1 Datasets

Drug side effects were extracted from the SIDER database[87,65]. Our matrix X contains $75,542$ known associations for 702 marketed drugs (rows) and $1,525$ distinct side effect terms (columns) (7.06% density). Each drug and each side effect has at least six known associations. A value $X_{ij} = 1$ if a drug i is known to be associated with side effect j or $X_{ij} = 0$ otherwise.

In order to build graphs representing side information for drugs, I assembled binary matrices describing drug target interactions (702 drugs$\times$401 targets), drug indication associations (702 drugs$\times 5,178$ indications), drug-drug interactions (702 drugs$\times$702 drugs) and Tanimoto chemical similarity (702 drugs$\times$702 drugs) – these datasets were extracted from DrugBank[119] and the Comparative Toxicogenomics database[120]. I then built the graphs using the cosine similarity between the rows of: the drug target matrix (I shall call this graph DT); the drug indication matrix (DInd); the drug-drug interaction matrix (DDI). The chemical graph (Chem) was built using the 2D Tanimoto chemical similarity from the SMILES chemical representation. For each graph, I set the main diagonal of the weighted adjacency matrix to zero. The distribution of similarity scores of each graph is shown in Fig. 2.4. In the experiments, I did not include any graphs representing side information for side effects.

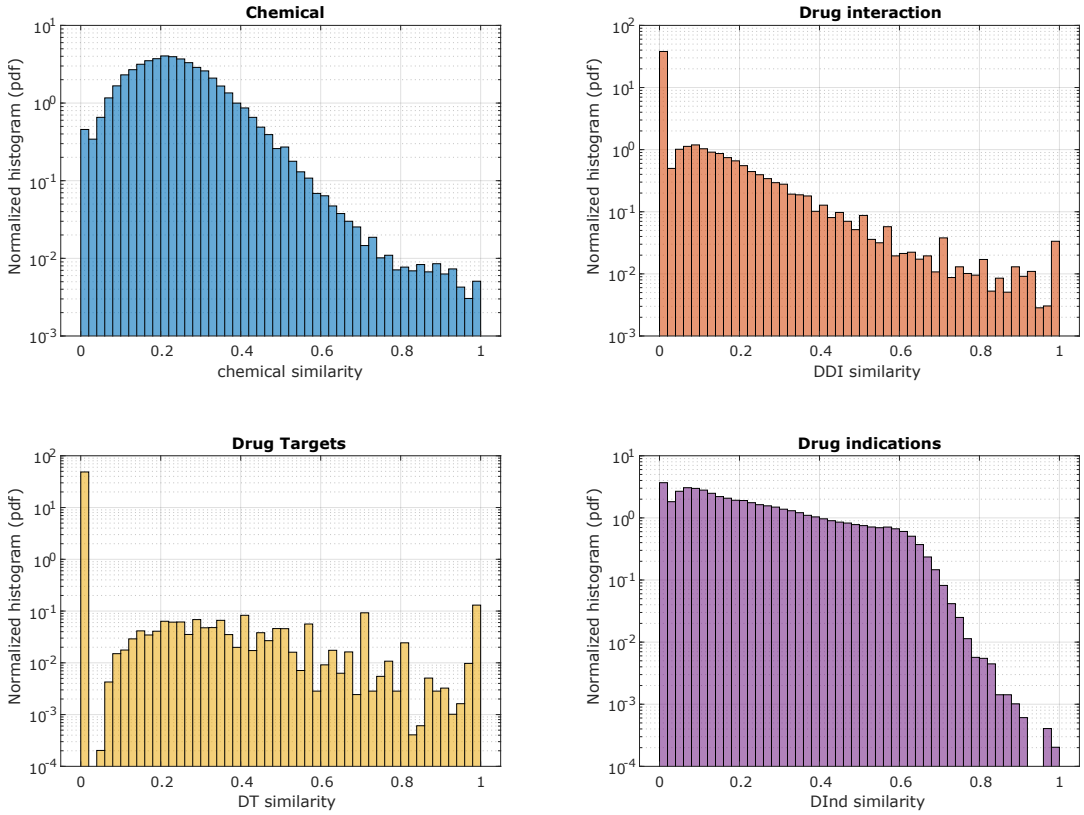

Figure 2.1: Normalised histogram of similarities used as graph side information for drugs: chemical similarity (blue), drug interaction (orange), drug targets (yellow) and drug indications (purple). All the similarities are bounded in the interval $[0, 1]$.

3.2 Experimental setting

Following previous approaches[6,32,36,34,35], I frame the side effect prediction problem as a binary classification problem. I applied ten-fold cross-validation, while optimising the hyperparameters using an inner loop of five-fold cross-validation within each of the ten folds (nested cross-validation for model selection[121]). The performance of the classifier is measured using the area under the receiver operating curve (AUROC) and the area under the precision-recall curve (AUPRC). I report the mean values of the ten folds for each met-

ric ($\overline{\text{AUROC}}$ and $\overline{\text{AUPRC}}$). I compared the performance of my method against Matrix factorisation (MF)[36], Inductive Matrix Completion (IMC)[35], Predictive PharmacoSafety Networks (PPNs)[6], Label propagation (LP)[32], Feature-derived graph regularised matrix factorisation (FGRMF)[34], and side effect popularity (TopPop)[20]. While every algorithm used the drug side effect matrix X, only IMC, PPNs, LP and FGRMF could also make use of the drug side information graphs. Optimal hyperparameters for each model were optimised to maximise the $\overline{\text{AUROC}}$. For GSMC, I optimise both models GSMC-c and GSMC-r independently. Then I set only the hyperparameter p using GSMC-c and GSMC-r with their obtained optimal hyperparameters.

3.3 Performance evaluation

Table 2.1 summarises the performance of the different methods. GSMC greatly outperforms the competitors both in terms of AUROC (by $1.3-19.7\%$) and in terms of AUPRC (by $4 - 30.9\%$). It is interesting to note that side effect popularity (TopPop) is highly predictive of drug side effects – this possibly reflects the fact that clinical reports are biased towards popular side effects such as headache or diarrhoea[65]. The optimal value of p in GSMC was 0.45, indicating that although GSMC-c performs better than GSMC-r individually, the final model weighs the combination in favour of the latter, which includes side information about drugs. Our method also informs about the relative contribution of each side information: I found that molecular networks were weighted higher ($\alpha^r_{\text{DT}} = \alpha^r_{\text{DDI}} = 1$), than networks containing chemical ($\alpha^r_{\text{Chem}} = 0.5$) or phenotype ($\alpha^r_{\text{DInd}} = 0.01$) information. Importantly, I observed that the performance of my model is robust with respect to the setting of the model parameters βs and λs (see the heatmaps in

Figs. 2.2-2.3).

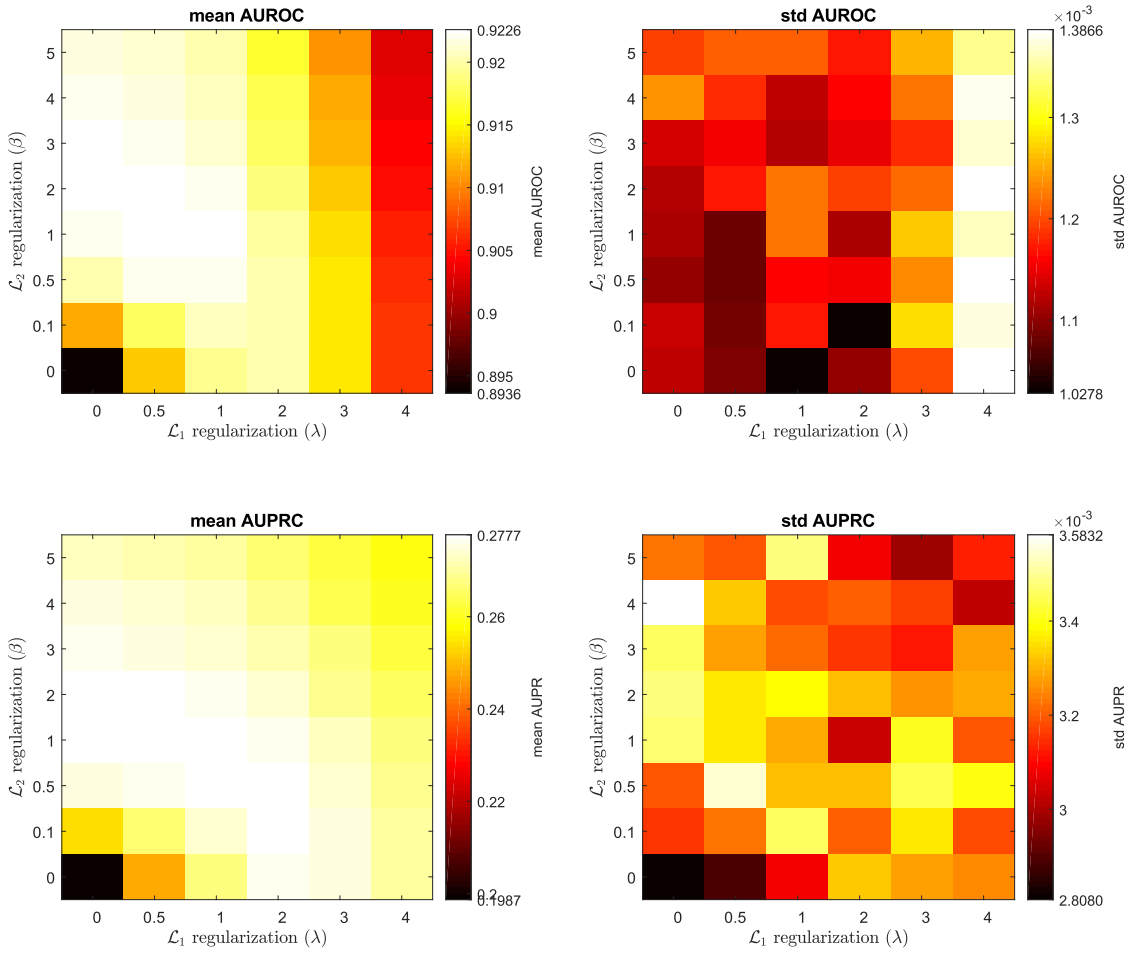

Figure 2.2: Heatmaps of the average performance of GSMC-c during model selection across the five-fold cross-validation in the validation sets. The performance is consistent across folds (small standard deviation) and it is not very sensitive to the setting of the model hyper parameters. Optimal performance can also be achieved by only using $\beta^c > 0$ (with $\lambda^c = 0$).

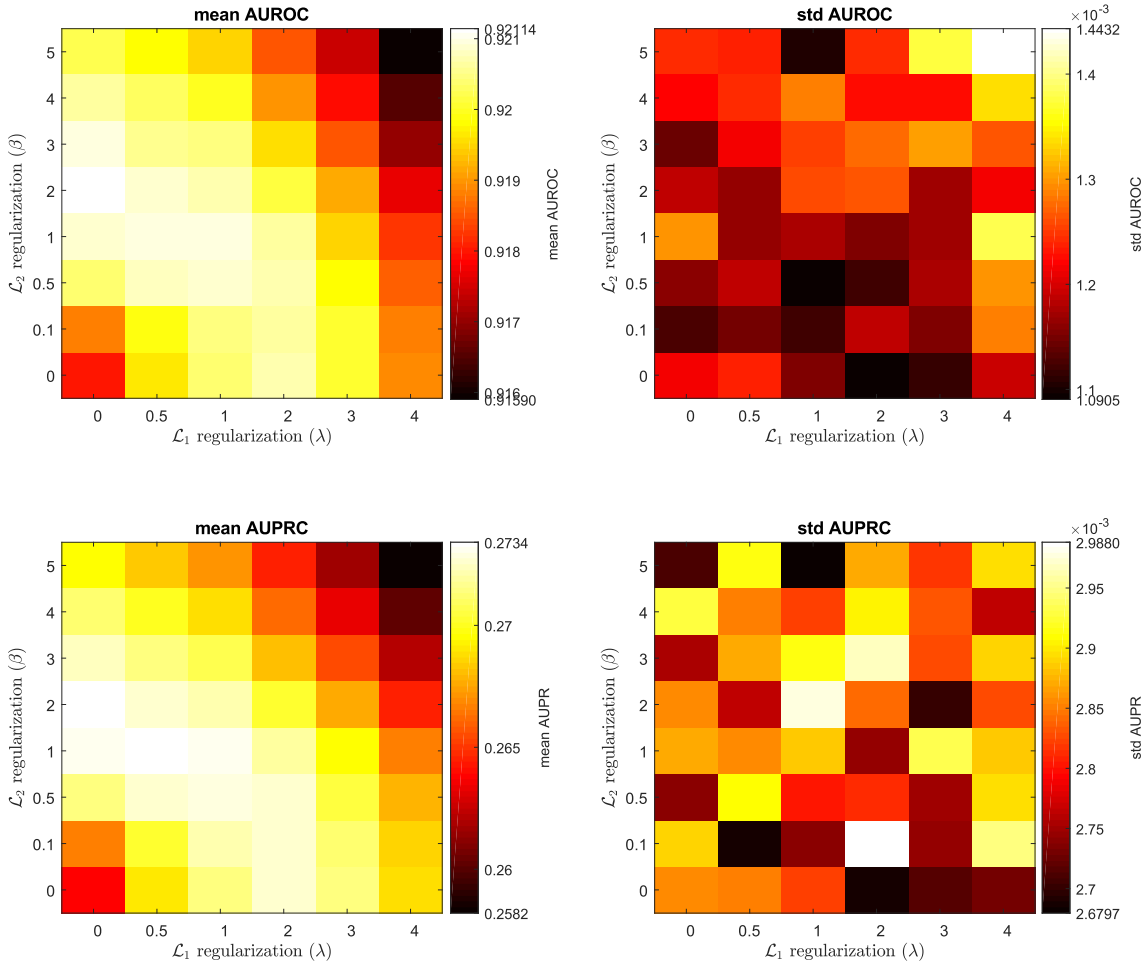

Figure 2.3: Heatmaps of the average performance of GSMC-r (without side graph) during model selection across the five-fold cross-validation in the validation sets. The performance is consistent across folds (small standard deviation) and it is not very sensitive to the setting of the model hyper parameters. Optimal performance can also be achieved by only using $\beta^r > 0$ (with $\lambda^r = 0$).

When comparing my method with competitor approaches, I found that a partial FGRMF[34] model based on the DDI graph only (FGRMF-DDI) performs better than the integrated model FGRMF – the fact that partial models could outperfom the integrated model had already been noted in the original publication. Furthermore, in the original publication[35], the IMC model was optimized using the observed entries only. Although matrix comple-

Table 2.1: Performance comparison for drug side effect prediction. Methods are ordered in ascending order of $\overline{\text{AUROC}}$.

Method	$\overline{\text{AUROC}}\pm$ s.t.d.	$\overline{\text{AUPRC}}\pm$ s.t.d.	Time (s)
IMC[35]	0.747 ± 0.0113	0.016 ± 0.0011	348.95 ± 23.71
TopPop[20]	0.827 ± 0.0031	0.071 ± 0.0028	0.010 ± 0.0014
LP[32]	0.888 ± 0.0021	0.126 ± 0.0033	0.018 ± 0.0032
IMCZeros	0.892 ± 0.0045	0.194 ± 0.010	317.149 ± 16.09
FGRMF[34]	0.911 ± 0.0029	0.237 ± 0.0059	209.27 ± 9.43
PPNs[6]	0.923 ± 0.0020	0.208 ± 0.0056	186 ± 5.91
MF[36]	0.929 ± 0.0019	0.274 ± 0.0071	31.12 ± 4.73
FGRMF-DDI[34]	0.931 ± 0.0020	0.285 ± 0.0075	30.41 ± 1.45
GSMC-r	0.936 ± 0.0014	0.295 ± 0.0073	3.19 ± 0.30
GSMC-c	0.938 ± 0.0023	0.323 ± 0.0024	15.29 ± 1.70
GSMC	0.944 ± 0.0017	0.325 ± 0.0063	17.82 ± 1.95

tion algorithms are predominantly based on this assumption[8,9,10,11,92,12,13,14,15], I found that taking into account the zeros can greatly improve the performance (I refer to this variant as IMCZeros in Table 2.1).

HIGH-RANK STRUCTURE OF THE DRUG SIDE EFFECTS MATRIX I verified that our $702 \times 1,525$ drug side effect matrix X has a high rank – its value is 701[‡] (see the spectra in Fig. 2.4). I observed that the reconstructed matrices also preserve the high-rank structure, but with smooth filtering of the spectra, indicating that smaller singular values are important to model weaker regularities in the data (see Fig. 2.5).

[‡]This was computed using the Matlab built-in function `rank`.

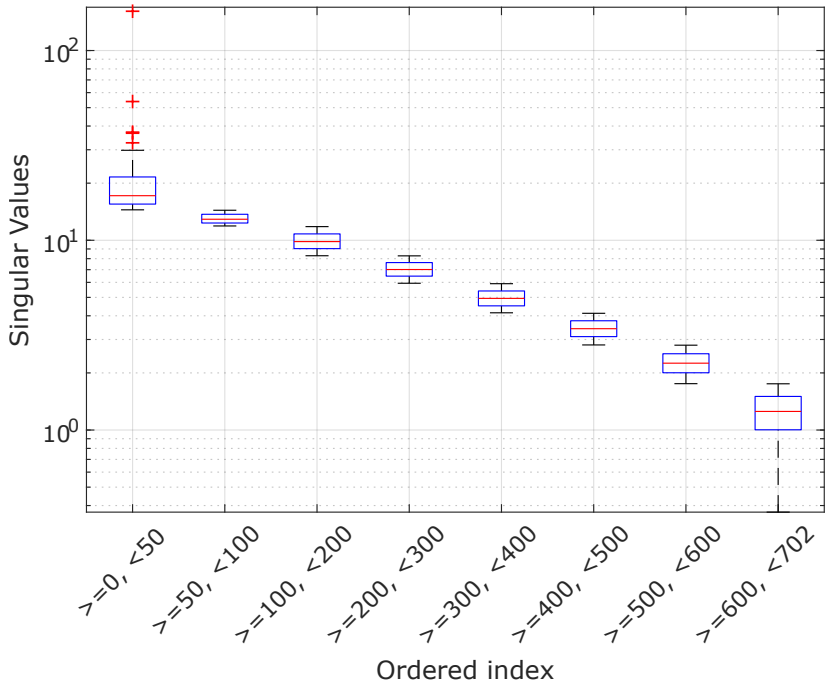

Figure 2.4: High rank structure of drug side effects. Boxplots of singular values of the data matrix X of drug side effects. Singular values were group according to their ordered index. The drug side effects data matrix has a high-rank: `rank(X) = 701`. And even the distribution of singular values 600th to 702th (the group with smaller singular values) rank significantly higher than zero (Wilcoxon Signed Rank Significance, $p < 1.83 \times 10^{-18}$).

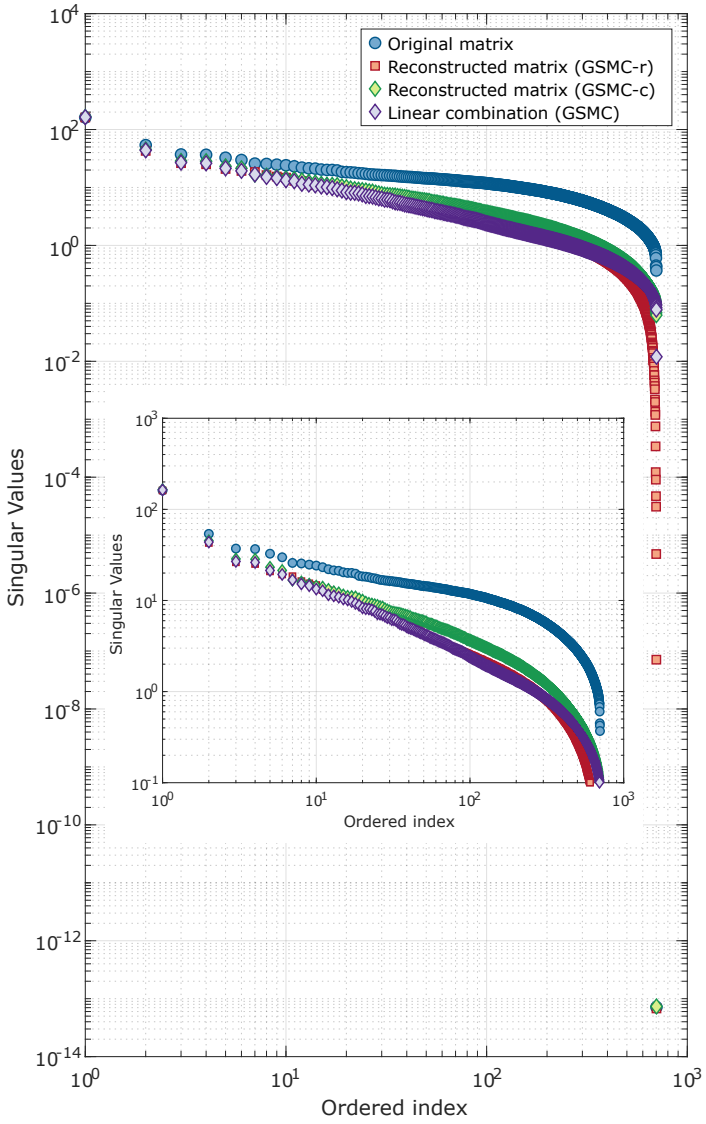

Figure 2.5: Smooth filtering of the spectra of singular values. Singular values of the original matrix X and the reconstructed matrices RX, XC and $\hat{X} \simeq \frac{1}{2}RX + \frac{1}{2}XC$. For this experiment, I did not consider side information graphs. The models GSMC-r and GSMC-c performs a smooth spectral filtering (de-noising). The density of the reconstructed matrix by GSMC-r is 47.74% (R has a density of 19.09%) and 48.83% by GSMC-c (C has a density of 5.58%). The threshold I used to calculate the densities was 0.01 for the reconstructed matrices and 1×10^{-4} for the sparse matrices. *Inset*. Zoom into a region of the spectra.

4 Biological interpretability

The effectiveness of my GSMC model at predicting the presence/absence of drug side effects prompted me to analyse whether the learned self-representation matrices are informative of the biology underlying drug activity. For the following experiments, I trained the models for GSMC using all the available data, fixed parameters ($\beta^r = 4, \lambda^r = 1, \beta^c = 2, \lambda^c = 0.5, \gamma^c = \gamma^r = 10^4$), and without side information graphs to prevent biases.

I first obtained a symmetry version of the learned matrices R and C, defined as $\mathcal{S}_R := R + R^T$ and $\mathcal{S}_C := C + C^T$, respectively. Drug and side effect similarities were then defined as the cosine similarity between rows of $\mathcal{S}_R$ and $\mathcal{S}_C$, respectively[§].

4.1 Drug self-representation predicts clinical activity and drug targets

I assessed model interpretability by exploring the extent to which drug self-representations (matrix R) were related to well-known drug clinical activity. Drug clinical activity was defined using the Anatomical, Therapeutic and Chemical (ATC) taxonomy, a hierarchical organisation of terms describing clinical activity where lower levels of the hierarchy contain more specific descriptors. I tested whether the similarity between two drugs was higher when they shared clinical activity. The evaluation was framed as a binary classification problem where the aim was to predict whether two drugs share an ATC category at different level of the taxonomy.

Figure 2.6a shows that my similarity is predictive of shared drug clinical activity. The predictions improve as we consider terms located lower in the ATC hierarchy (finer gran-

[§]I found that using the cosine similarity between the rows of $\mathcal{S}_R$, instead of $\mathcal{S}_R$ directly, slightly improves the prediction performance. This is probably due to the fact that the cosine similarity is less noisy as it takes into account the similarity between all the neighbours of each drug.

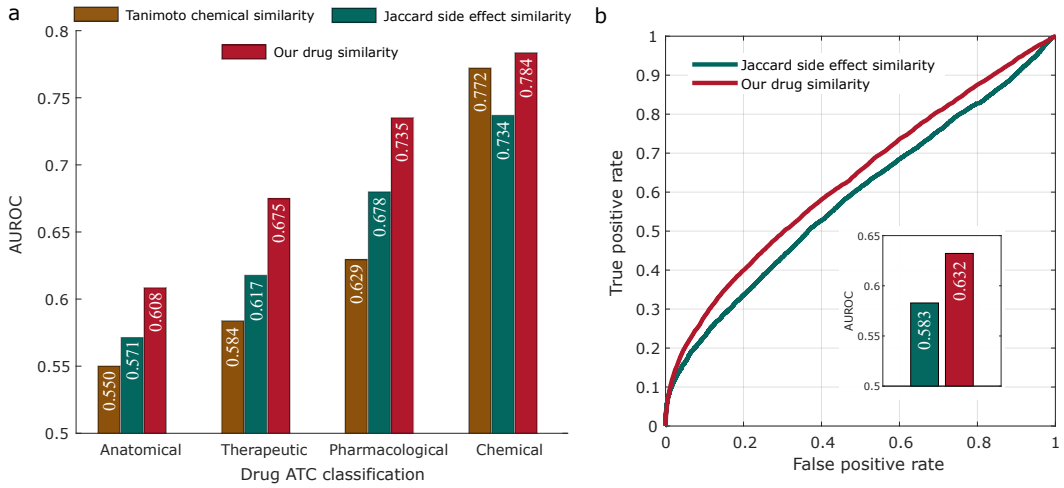

Figure 2.6: Our drug similarity captures drug clinical and molecular activity (a) AUROC representing the performance of my drug similarity, side effect similarity (Jaccard) and Tanimoto chemical similarity at predicting whether a pair of drugs share Anatomical, Therapeutic and Chemical (ATC) category at each level of the ATC taxonomy. *(b)* ROC curve representing the performance of my drug similarity at predicting whether pairs of drugs share a target. *Inset* AUROC barplot.

ularity) – this correctly reflects the fact that drug clinical responses become more similar as we move to lower (or more specific) levels of the ATC hierarchy. The figure also shows a comparison of the performance obtained for this problem with other methods used elsewhere[122,102,97]: 2D Tanimoto chemical similarity and Jaccard side effect similarity. The fact that my similarity performs better than the Tanimoto chemical similarity in the chemical ATC subclass is quite remarkable, as in my model drugs are characterised only by noisy information about a few side effects, rather than exact knowledge of chemical structures.

Fig. 2.7 presents the embedding of drugs in 3D based on $\mathcal{S}_R$ that is obtained applying t-SNE[123] together with the heatmap of the mean inter- and intra-class similarity $\mathcal{S}_R$ for each ATC anatomical classes. The figure shows that anatomically related drugs tend to be significantly similar (in their self-representation) that anatomically unrelated drugs.

Encouraged by these results, I decided to test whether my drug similarity could even be

used for the prediction of shared drug targets. Having framed this as a binary classification problem, I found that my drug similarities are predictive of shared protein targets between drugs (see Figure 2.6b). Note that, drug side effect similarity had previously been found to be predictive of drug protein targets at molecular level[88,97], but the fact that my similarity, that is built using the same data, works better, means that my model is able to exploit the information more effectively (4% AUROC improvement).

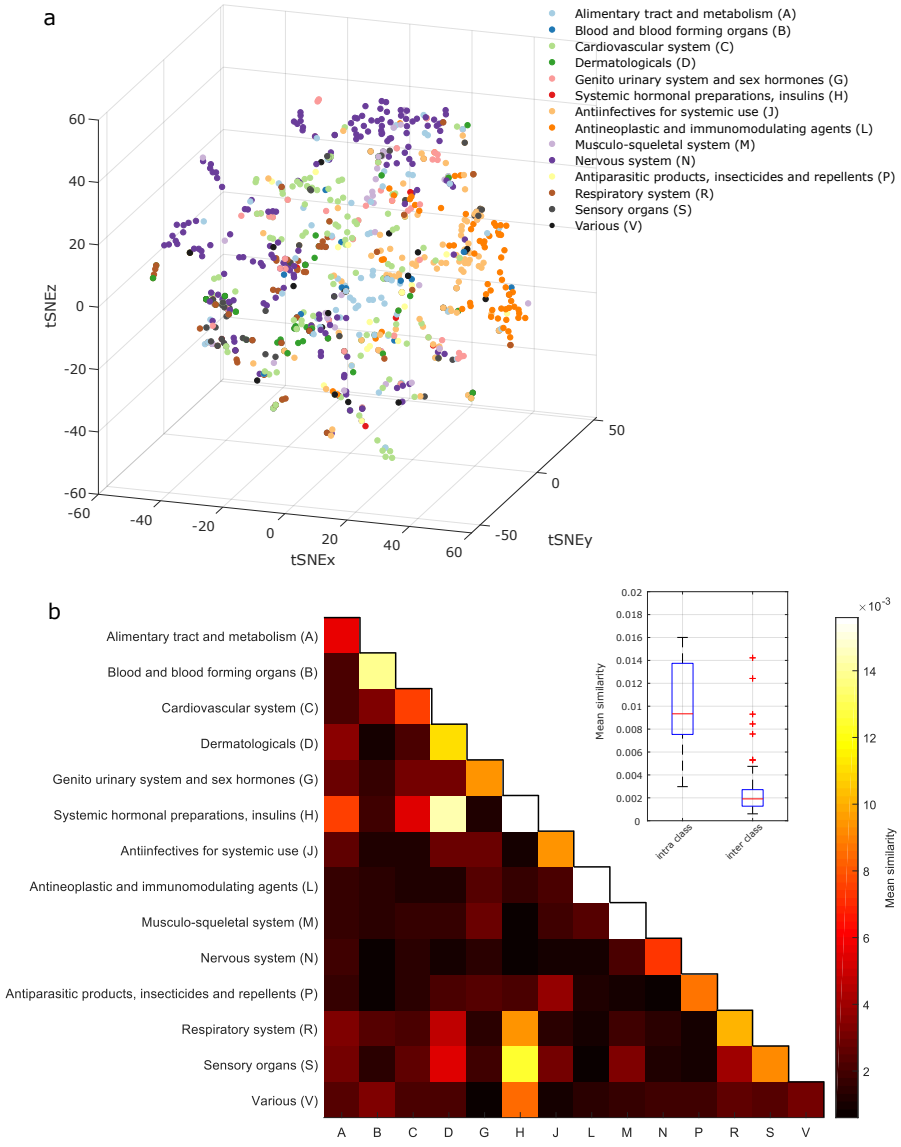

Figure 2.7: Drug self-representation similarity captures drug clinical activity (a) Embedding of drugs in 3D space using t-SNE. Each point represents a drug. Colours are assigned based on their anatomical category. Distance between points is related to the cosine distance of the drug clustering similarity $\mathcal{S}_R = R + R^T$. *(b)* Heatmap of mean drug similarities $\mathcal{S}_R$ per anatomical class. Each (x, y) tile represents, for each main Anatomical, Therapeutic and Chemical (ATC) drug category, the mean similarity of drug pairs where one drug belong to category x and the other to category y. The value ranges from 3×10^{-4} (Muscular skeletal system - Systemic Hormonal and Preparations) to 0.0078 (Muscular skeletal system-Muscular skeletal system). The colours range between the minimum mean similarity and 0.0156, with all values above 0.0156 (In the diagonal: 0.0921 (H), 0.0160 (M)) set to 0.0156. *Inset*: the average intra-class similarity is significantly higher than the average inter-class similarity (t-test Significance, $p < 7.12 \times 10^{-13}$).

I also analysed the link between side effect similarities and the anatomy/physiology of the side effect phenotypes. Side effects were grouped based on their anatomical class according to MedDRA[124]. I found that similarities for two side effects tend to be higher when they are phenotypically related. Figure 2.8 shows that, in most cases, the side effect similarity within system organ classes (top level of the MedDRA hierarchy) is higher than the similarity between classes. Moreover, side effect similarity is predictive of shared MedDRA category at each of the different levels and predictions improve as we move to more specific terms in the MedDRA hierarchy.

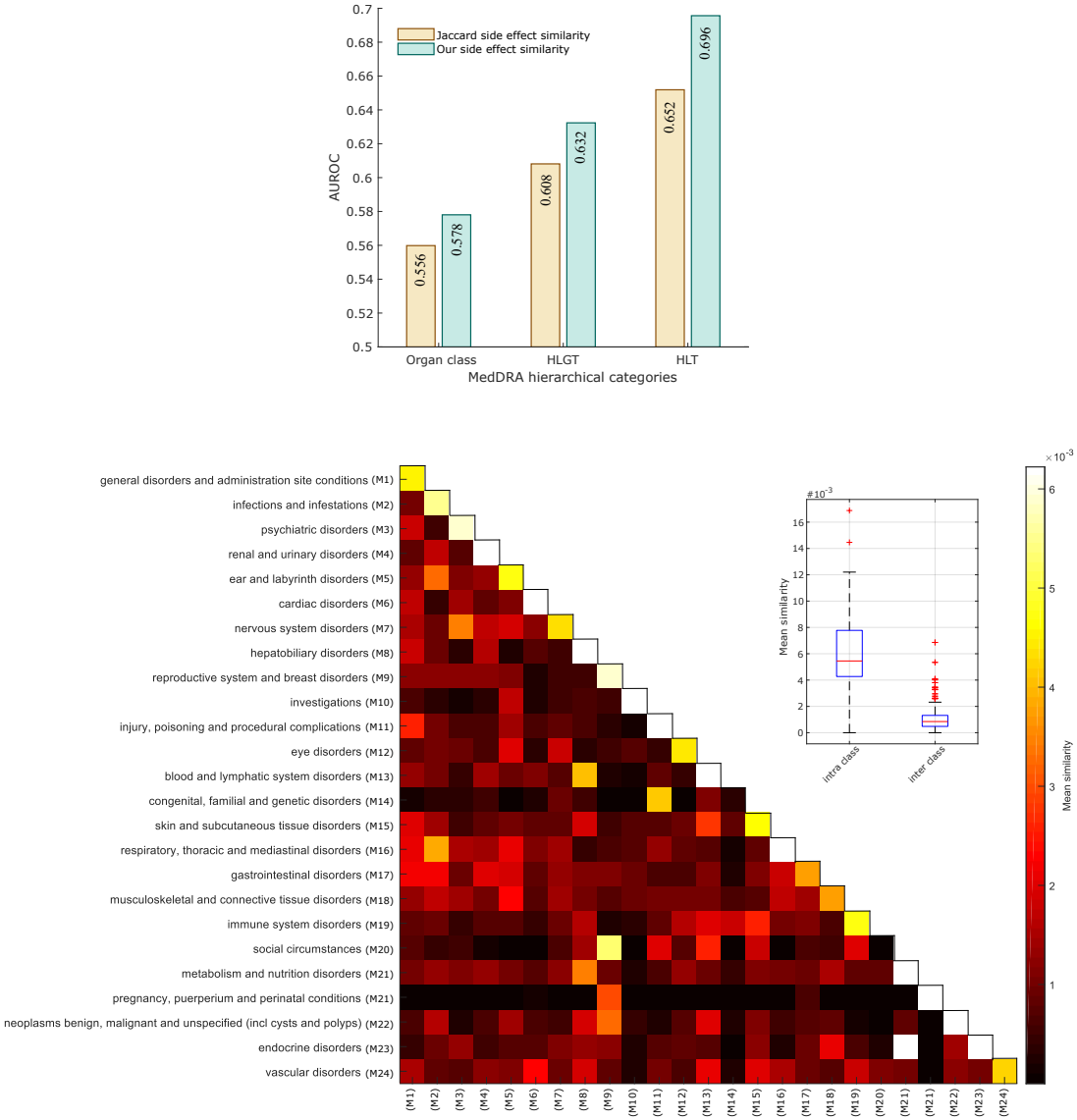

Figure 2.8: Side effect sparse matrix of coefficients similarity captures human phenotype similarity (Top) Ability of my side effect similarity ($\mathcal{S}_C = C + C^T$) and the Jaccard side effect similarity to predict whether two side effects belong to the MedDRA class at different levels of the hierarchy. *(Bottom)* Heatmap of mean side effect similarities $\mathcal{S}_C$ per organ class. Each (x, y) tile represents, for each main MedDRA organ class, the mean similarity of side effect pairs where one side effect belong to category x and the other to category y. The value ranges from 1.29×10^{-24} (M21 - M14) to 0.017 (M8-M8). The colours range between the minimum mean similarity and 0.0062, with all values above 0.0062 (In the diagonal: 0.0075 (M4), 0.0098 (M6), 0.0169 (M8), 0.010 (M10), 0.0067 (M11), 0.014 (M13),0.0062 (M16),0.0065(M21), 0.00863 (M23),0.012 (M24); off-diagonal:0.00686 (M24-M21)) set to 0.0062. *Inset*: the average intra-class similarity is significantly higher than the average inter-class similarity (t-test Significance, $p < 7.14 \times 10^{-81}$).

For completeness, in this section, I introduce details about the implementation, optimisation, and hyperparameters tuning for each method. In some cases, I also show the performance of the methods using individual graphs. All the models were run in a small cluster with 32 physical cores and 100GB of RAM. Each processor is a Intel(R) Xeon(R) CPU E5-2683 v4 @ 2.10GHz. I implemented most of the algorithms in Matlab R2018a 64-bit. The setting of the model parameters was executed in parallel using the Parallel Computing Toolbox version 6.12.

Low-rank matrix factorisation (MF) For the optimisation, I used conjugate gradient descent (CGD) and approximated line searches based on polynomial interpolation with Wolfe-Powel conditions to find the local minima in (2.16). I used the same Matlab implementation of the Rassmusen minimiser to solve the problem[¶]. To run my algorithm, I initialise P and Q as normally distributed random variables with small variance $\sigma^2 = 0.01$. I tuned both model parameters: the number of latent factors k and the regularisation penalty λ in the grid: $k \in \{10, 30, 50, 70, 90, 100\}$ and $\lambda \in \{0.1, 1, 5, 10, 15, 20\}$.

Predictive Pharmacosafety Networks (PPNs) For the optimisation, I used the Matlab built-in function `mnrfit` (link function `logit`) to learn the coefficients estimates (β_s) of the multinomial logistic regression of the nominal responses in X on the predictors in Z. PPNs does not require parameter tuning.

[¶] http://learning.eng.cam.ac.uk/carl/code/minimize/

Inductive Matrix Completion (IMC) For the optimisation, I used conjugate gradient descent (CGD) and approximated line searches based on polynomial interpolation with Wolfe-Powel conditions to find the local minimum. I initialised W and H as normally distributed random variables with small variance $\sigma^2 = 0.01$. I tuned both model parameters: the number of latent factors k and the regularisation penalty λ in the grid: $k \in \{10, 30, 40, 50, 70, 90, 100\}$ and $\lambda \in \{0.1, 1, 5, 10, 15, 20\}$.

Feature-derived graph regularised matrix factorisation (FGRMF) For the optimisation, I used conjugate gradient descent (CGD) and approximated line searches based on polynomial interpolation with Wolfe-Powel conditions to find the local minimum in equation (2.11). To optimise equation (2.12), I use the Matlab built-in function `mnrfit` (link function `logit`) to learn the coefficients estimates (θ_j). I initialise W and H as normally distributed random variables with small variance $\sigma^2 = 0.01$. I tuned the three model parameters in the grid $\lambda \in \{1, 5, 10, 15, 20\}$, $k \in \{20, 30, 40, 50, 60, 70, 80, 90, 100\}$, and $\alpha \in \{0.1, 1, 2, 3, 4, 5\}$.

Label propagation using consistency method (LP) For the optimisation, I used the close form of F to obtain the scores for drug side effects. I tuned α from 0.001 to 0.9 in steps of 0.01 (90 values).

Geometric sparse matrix completion (GSMC) I followed the recommended guidelines used to implement non-negative matrix factorization (NMF) in [91]. R and C were initialised with weights from a uniform distribution between $(0, \sqrt{\sigma}]$ for a $\sigma = 0.01$. The maximum number of iterations was set to 100 and `tolX` = 0.01. With this `tolX`, con-

vergence occurs in roughly 50 iterations. To set γ^c, γ^r, using our theoretical bounds, I observed that, for GSMC-c $L^c = \max_i \text{diag}(X^\top X) = 602$, and for GSMC-r, $L^r = \max_i \text{diag}(XX^T) = 644$. Thus, a $\gamma^c = \gamma^r = 10^4$ is enough to obtain an $\varepsilon \approx 9.54 \times 10^{-63}$. Therefore, γ^c and γ^r were set to 10^4 for all the experiments. For the partial GSMC-c model I tuned the parameters in the following grid $\beta^c \in \{0, 0.1, 0.5, 1, 2, 3, 4, 5, 10, 20\}$, $\lambda^c \in \{0, 0.5, 1, 2, 3, 4, 5\}$. For the partial GSMC-r model that integrates side information, I reduced the grid due to the larger number of possible combinations: $\beta^r \in \{1, 2, 3, 4, 5, 10\}$, $\lambda^r \in \{0.1, 0.5, 1\}$ and $\alpha^r \in \{0.01, 0.1, 0.5, 1\}$. Finally, to train the GSMC model (the linear combination of GSMC-c and GSMC-r), I tuned only $p \in \{0, 0.01, 0.02, ..., 1\}$ while setting the following hyperparameters for the partial models: GSMC-c ($\beta^c = 1, \lambda^c = 0.5$) and GSMC-r ($\beta^r = 2, \lambda^r = 0.5, \alpha^r_{chem} = 0.5, \alpha^r_{DDI} = 1, \alpha^r_{DT} = 1, \alpha^r_{DInd} = 0.01$). These settings were based on the optimal values during model selection of each partial model.

6 Conclusion and Discussion

In this chapter, I have focused on the problem of drug side effect identification. This problem is typically framed as a binary classification problem where the goal is to correctly identify the presence or absence of drug side effect associations. I have introduced two matrix completion models to solve the problem.

The first model is a simple low-rank model that learns a low-dimensional embedding for each drug and each side effect. The embedding, i.e. the coordinates of each drug and each side effect in a space (datapoints), are *learned* through an optimisation process. These drug and side effect datapoints are not randomnly placed in a space but rather their relative positions to each other determines whether a drug might cause a side effect. In effect, the objec-

tive function tries to ensure that the scalar product between the vector representing a drug, and the vector representing a side effect, reflects whether the drug is associated to the side effect. That is, a drug-side effect pair known to be associated will tend to be placed closeby in this space, whereas a pair not known to be associated will tend to be placed far apart from each other. Notice that this principle will not always obey due to the dimensionality reduction, i.e. many far-apart datapoints will be placed closeby in the low-dimensional space; this is indeed the goal of the modelling.

The second model that I presented in this chapter is a high-rank model. In a low-rank model, we explicitly hard-constrained the rank of the resulting matrix to be low (at most k); this is why each drug and each side effect are in the same low-dimensional space. In contrast, in a self-expressive model, which is a high-rank model, not such hard-constrain exist. Notice that while in a low-rank model both drugs and side effects are represented with latent components that are not inherently interpretable, in a self-expressive model, each drug or each side effect will be represented in terms of all the other drugs or all the other side effects, respectively. It is a re-construction of a datapoint from its neighbours in a drug-space or a side effect space. This is why I enforce this representation to be sparse, so that each drug or each side effect is only represented by few of its neighbours, leading to more interpretable representations. For instance, Fig. 2.9 shows an example using the GSMC-c model (that learns self-representations for side effects) and Lindane, a drug that has been withdrawn from the market due to concerns about neurotoxicity. Lindane is amongst the drugs with the smallest number of side effects in our dataset (1.5^{th} percentile) – only 10 side effects are present. Figure 2.9a shows the histogram of the values found in the row corresponding to Lindane in XC. My model predicts that Lindane is likely to cause

hypotension (the score is in the 98.8th percentile) and indeed this side effect has been repeatedly reported[125,126]. Figure 2.9b provides the rationale behind this prediction. The score for Lindane-hypothension is the sum of the (non-negative) coefficients in the column of C corresponding to hypotension for the 10 known side effects of Lindane. Notice how `seizures`, a condition normally associated to hypothension, explains 37.92% of the score strength. It is important to mention that since my algorithm reaches a global optima, this score is the optimal one. As illustrated by this example, an analysis of the self-representation coefficients learned by my model can potentially provide biological clues to generate medical and pharmacological hypothesis when assessing the safety of a drug.

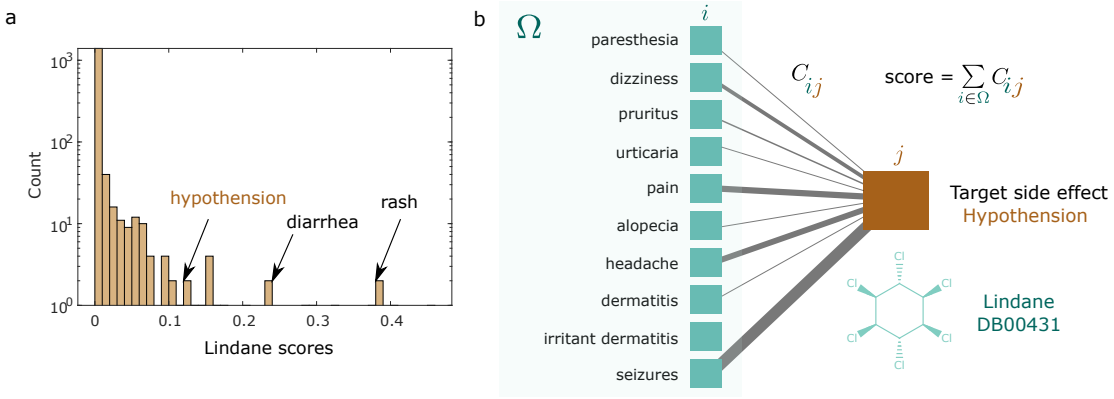

Figure 2.9: Example of explainable predictions for the withdrawn drug Lindane (a). Histogram of predicted scores for Lindane using GSMC-c; *(b)* Network diagram depicting how the model generates the predictions for a given target side effect under study. In the figure, Ω represents the set of known side effects indexed by i, and j is the target side effect. The thickness of the connections are proportional to the learned coefficients.

Inherently interpretable models are critical for applications involving high-stakes decisions in health care[27]. In this context, it is important to mention that I refer to *interpretability* in terms of the learned representations in my machine learning models. Can the learned representations can be understood in the light of some human rationale?. As discussed before, when negative weights are allowed in a matrix multiplication, it is unfeasible to obtain

interpretability, as the representations tend to be *holistic* rather than parts-based; the principle in which human cognition is based[127].

To my knowledge, my work is the first that relies on the high-rank models to predict drug side effects and also disease genes. I envision the application of my models to other problems in computational biology and pharmacology with similar high-rank structure, including in the study of social networks. Interestingly, I observed that many datasets used in different application domains have a high-rank structure that could be exploited to improve modelling performance and generalisation capabilities. I will show the use of self-expressive models for predicting disease genes in Chapter 2 and user preferences in recommender systems in Chapter 3; in both cases, I compare the performance of high-rank vs low-rank models.

Open the book of life and you will see a text of about 3 billion letters, filling about 10,000 copies of the new York Times Sunday edition. Each line looks something like this: `TCTAGAAACA ATTGCCATTG TTTCTTCTCA TTTTCTTTTC ACGGGCAGCC`

Albert-László Barabási (Linked, 2014)

3

Disease gene prediction

The identification of genes that are associated with common and rare heritable human diseases* is of central importance for disease prevention, diagnosis and therapy, and thus has attracted increasing attention over the last decades[128,129,130]. Traditional methods

*Common diseases such as diabetes are complex diseases, i.e. there are many factors involved (both genetic and non-genetic) in their pathogenesis. Whereas rare, heritable diseases are often monogenic, i.e. caused by a single genetic defect.

for identifying disease genes have focused on genetic linkage analysis, which uses statistical tools to discover chromosome regions that are likely to harbour heritable trait genes[131]. However, such type of analysis has been inefficient for complex traits that are caused by mutations in multiple genes, e.g. diabetes[132]. Alternative approaches have recently shifted the focus to DNA sequencing studies of large case-control populations[133,134]. Disease associated alleles can then be discovered by measuring their difference in frequencies between cases and controls. However, such methodologies often result in hundreds of candidate genes and identifying the particular gene affected by a specific genomic variant remains a challenging task[135].

Recent network medicine based approaches exploit the fact that genes involved in the same disease tend to interact with each other[136] and that mutations in interacting proteins tend to cause similar disease phenotypes[19]. These interdependencies between genes and (patho)phenotypes are essential to several state-of-the-art methods that operates under the guilt-by-association principle[137,138]. The common idea of these approaches is that genes are prioritised by their functional "proximity" to those genes already known associated to the disease. Functional proximity is typically quantify in a human Protein-Protein Interaction (PPI) network, where nodes are proteins and links represent relevant functional relationships between the proteins, such as physical interaction[19]. The methods differ in how the proximity is quantified over the network; by direct neighbours[129], random walks[139], graph kernels and Markov random fields[140], or propagation flow and clustering techniques[141]. Amongst these, network propagation has emerged as a powerful paradigm to amplify a biological signal based on the assumption that genes underlying similar phenotypes tend to interact with one another[112]. For instance, PRINCE[40] uses a label propagation method to

"diffuse" the knowledge on gene labels — known disease gene associations— over a PPI network. ProDiGe[39], which implements a positive-unlabelled learning algorithm, is particularly powerful because it incorporates information about known disease genes from multiple sources and shares such information across all diseases to help suggest new candidate genes for query diseases. There are four methods in the ProDiGe family, and they differ in their respective disease sharing kernels. Cardigan[41], proposed more recently, uses the consistency method[109] and enriches the initial gene labels by defining a prior probability distribution over all known genes associated to diseases using phenotype similarities[142].

In network medicine, genetic diseases are seen as localised perturbations within a neighbourhood of the PPI network[143,19] — the disease module[19]. DIAMonD[38] has been developed to detect disease modules on the PPI network. A disease module is obtained by iteratively adding new genes that have the greatest connections with the existing disease genes into the module. The order in which the genes are added to the disease module can be used as a ranking that prioritises disease genes.

More recently, the disease gene prediction problem has been framed as a matrix completion task. This means that the ground truth knowledge about n genes and m diseases is modelled by a (binary) matrix $X \in \mathbb{R}^{n \times m}$ which is the product of an $n \times k$ matrix W whose rows are the gene feature vectors and a $k \times m$ matrix H whose columns are the disease feature vectors. The rank of X is k — the number of features assigned to each gene or disease. For instance, Natarajan et al.[144] proposed an inductive matrix completion model which integrates gene and disease features in a regularisation framework. Yang et al.[145] obtained low-dimensional embedding of nodes in a heterogeneous network that includes disease-symptom associations, disease-gene associations, gene-function associations, i.e.

gene-ontology terms, and PPI network. The low-dimensional embedding was then used to compute similarity scores for the prediction. The use of matrix completion models have also been useful in related prediction tasks such as miRNA-disease association prediction[146,147,148,149].

In this chapter, I propose a High-Rank Matrix Completion (HRMC) model with graph regularisation for prioritising genes for common and rare genetic diseases. The model that I present in this chapter is similar to the one previously presented for predicting drug side effects in Chapter 2. My model for disease gene prediction is more interpretable as it learns sparse self-representations of nodes: each disease is represented by a linear combination of few other diseases. My self-representation model takes into consideration the relational graph structure of diseases and genes built from their relatedness in terms of human phenotypes[142] and physical interactors in the PPI network. The use of complementary information is crucial to predict genes for diseases with not known molecular basis[144,41,1]. Through extensive experiments on the Online Mendelian Inheritance in Man (OMIM) database, I show that my method outperforms state-of-the-art approaches in gene prioritisation. I also validate the predictions with a prospective analysis of case studies. That is, having trained my models with data available until 2017, I checked whether some top predictions made by my model has been associated with diseases in 2018. The prospective analysis is important to illustrate a realistic use of my approach by preserving the chronological order in which information becomes available.

1 The HRMC model

1.1 The HRMC objective function

Let us denote our gene-disease association matrix for n genes and m diseases with the binary matrix $X \in \mathbb{R}^{n \times m}$ where $x_{ij} = 1$ if gene i is known to be associated to disease j, or $x_{ij} = 0$ otherwise. The goal of my HRMC model is to *learn* two sparse matrices of coefficients, one for the row elements ($R \in \mathbb{R}^{n \times n}$) and one for the column elements ($C \in \mathbb{R}^{m \times m}$), such that the data matrix X is approximated by:

$$\hat{X} \simeq pXC + (1-p)RX \tag{3.1}$$

where $p \in [0, 1]$ is a hyperparameter that controls the balance between the row (genes) and column (diseases) contributions. In the sequel, I shall refer to the first part of the HRMC model XC, as HRMC-c, and to the second part, RX, as HRMC-r.

To this end, two cost functions, $\mathcal{Q}_c(C)$ and $\mathcal{Q}_r(R)$, that take into account the relational inductive prior between datapoints — the graph network — for diseases and genes, are minimised with respect to C and R, respectively:

$$\min_{C \geq 0} \mathcal{Q}_c(C) = \underbrace{\frac{1}{2}\|X - XC\|_F^2}_{\text{self-representation}} + \underbrace{\frac{\beta^c}{2}\|C\|_F^2 + \lambda^c \|C\|_1}_{\text{sparsity}} + \underbrace{\frac{\alpha^c}{2}\|C - G^c\|_F^2}_{\text{graph regularisation}} + \underbrace{\gamma \operatorname{Tr}(C)}_{\text{null diagonal}} \tag{3.2}$$

$$\min_{R \geq 0} \mathcal{Q}_r(R) = \underbrace{\frac{1}{2}\|X - RX\|_F^2}_{\text{self-representation}} + \underbrace{\frac{\beta^r}{2}\|R\|_F^2 + \lambda^r \|R\|_1}_{\text{sparsity}} + \underbrace{\frac{\alpha^r}{2}\|C - G^r\|_F^2}_{\text{graph regularisation}} + \underbrace{\gamma \operatorname{Tr}(R)}_{\text{null diagonal}} \tag{3.3}$$

where $\|.\|_F$ denoting the Frobenius norm. and $\mathcal{G}^c = (\{1,\ldots,m\}, \mathcal{E}^c, G^c)$ denote the weighted undirected graph with edge weights $g^c_{ij} > 0$ of $(i,j) \in \mathcal{E}^c$ and zero otherwise, representing the similarities between diseases. $\mathcal{G}^r = (\{1,\ldots,n\}, \mathcal{E}^r, G^r)$ denote the binary undirected graph with edge weights $g^r_{ij} > 0$ of $(i,j) \in \mathcal{E}^r$ and zero otherwise, representing the protein interaction network. C and R in Equation (3.1) are learned by minimising Equations (3.2) and (3.3), respectively. In the following, I provide the rationale behind (3.2) only, as the same applies to (3.3).

The first term in Equation (3.2) is the *self-representation constraint*, which aims at learning a matrix of coefficients C such that XC is a good reconstruction of the original matrix X. The second term is the *sparsity constraint*, which uses the elastic-net regularisation known to impose sparsity and grouping-effect[114,115]. The third term is the *graph regularisation* constraint, which incorporates the structure revealed by the pairwise disease similarities into the sparse coefficient matrix C. The fourth term is the *null-diagonal constraint*, which has the important role of preventing the trivial solution $C = I$ by imposing $\text{diag}(C) = 0$. This is achieved through a regularised trace operator $\gamma^c \text{Tr}(C)$. Typically, $\gamma^c \gg 0$ is used. Finally, following[29], I impose *non-negative constraints* on C, as these constraints lead to more interpretable model since they allow only for additive combinations.

It is worth noticing that the main difference between my model presented here and my geometric model shown in chapter 3 lies in the way the graph structure is incorporated into the model. In the GSMC model of chapter 3, I used the smoothness constraint that enforces each self-representation vector (columns of C) to reflect the similarities between pairs

of nodes in the graph, which is obtained by minimising:

$$\sum_{i,j} G^c_{ij} \|c_i - c_j\|^2 \tag{3.4}$$

However, in the HRMC model, each element in the self-representation vector (an entry c_{ij}) is enforced to reflect the similarities between pairs of nodes in the graph, which is obtained by minimising:

$$\sum_{i,j} \|c_{ij} - G^c_{ij}\|^2 \tag{3.5}$$

Clearly, the constraint in Eq. 3.5 is different to the constraint in Eq. 3.4. To understand this, consider that Eq. 3.4 penalises the self-representation vectors of two diseases i and j by their phenotype similarity in G^c_{ij}; phenotypically similar diseases will have similar self-representations. Conversely, Eq. 3.5 encourage a single entry c_{ij} to reflect the similarity between diseases i and j. A single entry in C indicates the subspace proximity between disease i and disease j as revealed by the self-expressive model. The reason to adopt the constraint in Eq. 3.5 rather than using the geometric constraint in Eq. 3.4 is due to time complexity. The algorithm derived for the graph regularisation (non-geometric) can be run in parallel for each disease in the matrix ($\mathcal{O}(m)$ vs $\mathcal{O}(m^3)$), which benefits the leave-one-out procedure.

1.2 The multiplicative learning algorithm

To minimise Equations (3.2) and (3.3) subject to the non-negative constraints $R, C \geq 0$, I developed two efficient multiplicative algorithms inspired by the diagonally rescaled princi-

ple of non-negative matrix factorisation[29,117]. The algorithm consists in iteratively applying the following multiplicative update rules:

$$C_{ij} \leftarrow C_{ij} \frac{(X^T X + \alpha^c G^c)_{ij}}{(X^T X C + \alpha^c C + \beta^c C + \lambda^c + \gamma^c I)_{ij}} \tag{3.6}$$

$$R_{ij} \leftarrow R_{ij} \frac{(X X^T + \alpha^r G^r)_{ij}}{(X X^T R + \alpha^r R + \beta^r R + \lambda^r + \gamma^r I)_{ij}} \tag{3.7}$$

It is straightforward to prove that my objective functions are convex and that my algorithms satisfy the KKT complementary conditions of global minimum convergence. Since the proofs are very similar to those shown for our geometric model in Chapter 2, I omit the proofs here.

2 Overview of my approach

At the first stage of the HRMC approach (Fig. 3.1), I integrated data from multiple sources, including data on gene-disease associations, disease similarities and protein-protein interaction network. Next, I constructed three matrices containing these associations (Fig. 3.1B). My model generates two score matrices (RX and XC) that are then linearly combined as a final prediction score. I performed a systematic evaluation of the model's performance using leave-one-out cross-validations. I report the results in two test cases[1]: (i) *molecularly characterised diseases*: if a disease for which a gene is found in the test set has known genes in the training set; and (ii) *molecularly uncharacterised diseases*: if a disease for which a gene is found in the test set does not have any known gene in the training set. I also validate some of my top predictions with more recent gene-disease associations in the 2018 snapshot.

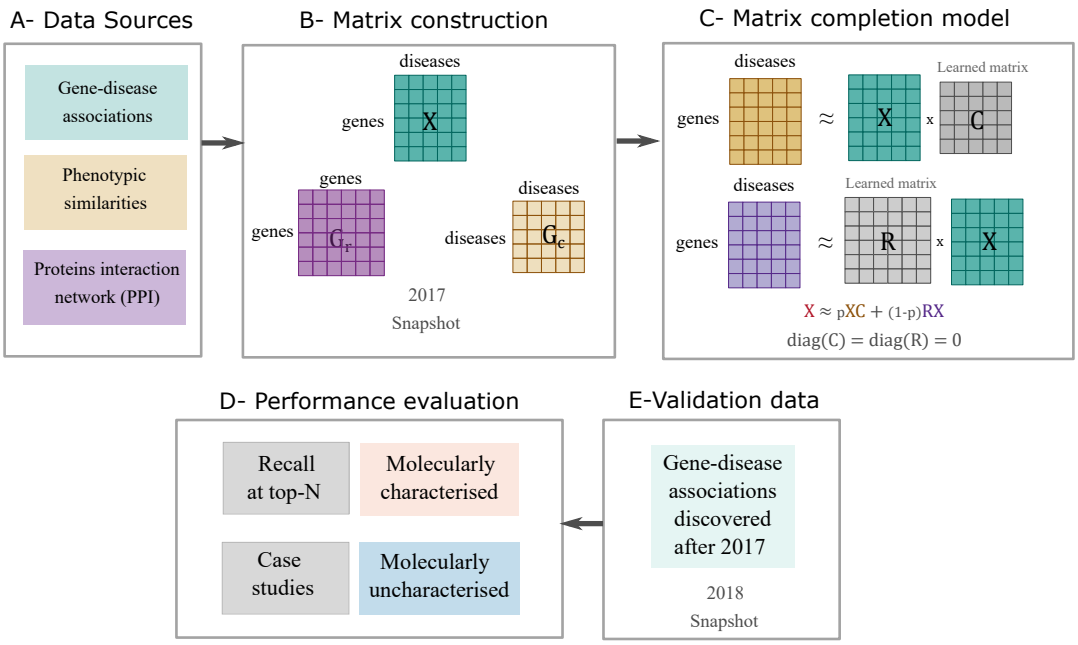

Figure 3.1: Overview of the HRMC approach. (A) First, data were integrated from multiple sources, including disease causing genes (gene-disease associations contained in two chronologically separated snapshots in OMIM: one from 2017 and another from 2018), disease similarities (built from information available up to 2017), and biological data (protein-protein interaction network from 2010). (B) Next, matrices of all associations contained in the 2017 database snapshot was constructed. (C) The matrices were used to train the row and column high-rank matrix completion models. Each separate model generates a score matrix for all disease gene associations. These are then linearly combined. (D) Next, leave-one-out cross validation was used to assess the recall of the method at different top-Ns, for both cases, *molecularly characterised* and *molecularly uncharacterised* diseases. (E) Finally, I further validate the predictions by case-studies of newly reported gene-disease associations in 2018.

3 Data description

The gene-disease associations were obtained from the Online Mendelian Inheritance in Man (OMIM) at two different points in time: 2017 and 2018. The 2017 snapshot contains in total 4,027 associations covering a total of $n = 9,670$ genes and $m = 5,768$ genetic diseases. In this dataset, only 3,455 (60%) diseases have known associations. Of these, 3,252 diseases have a single known gene, whereas the remaining have not known gene in 2017. A binary protein-protein interaction network was obtained from the Human Protein Refer-

ence Database (HPRD)[150]. Our PPI covers 9,670 genes with 37,041 known experimental interactions. Phenotype disease similarity was obtained from Caniza et al.[142]. The disease similarity proposed by Caniza et al.[142] relies on Medical Subject Headings (MeSH) terms associated with publications. Since MeSH terms are also dynamic — their content and number depend directly on the publications available in a particular point in time — I also built our semantic similarity based on MeSH data available up to 2017.

3.1 HRMC learns an aggregated guilt-by-association

I can explain my HRMC model in terms of the guilt-by-association (GBA) principle. The GBA principle is based on the assumption that genes which are associated or interacting are more likely to share function[137]; thereby, more likely to be involved in the same disease[19]. This principle is illustrated in Figure 3.2a, and I referred to it as gene-based GBA. In this case, a new disease-gene association is *established* by relating a known gene to a target gene through physical interaction. My HRMC-r model can be interpreted as an extension of this principle (see Figure 3.2b), where a new disease-gene association is *predicted* by relating the known genes to a target gene through a learning process that also involves the knowledge of protein interaction networks[†].

[†]To understand this in detail, consider that HRMC-r generates scores through the product RX. For a given $(RX)_{ij}$, the prediction is obtained by the dot product of r_i (the ith row of R) and x_j (the jth column of X), as follow: $(RX)_{ij} = r_i x_j$. Recall that X is binary, therefore $(RX)_{ij} = \sum_{(k \in \Omega)} (r_i)_k, \Omega \in \{k | (x_j)_k > 0\}$.

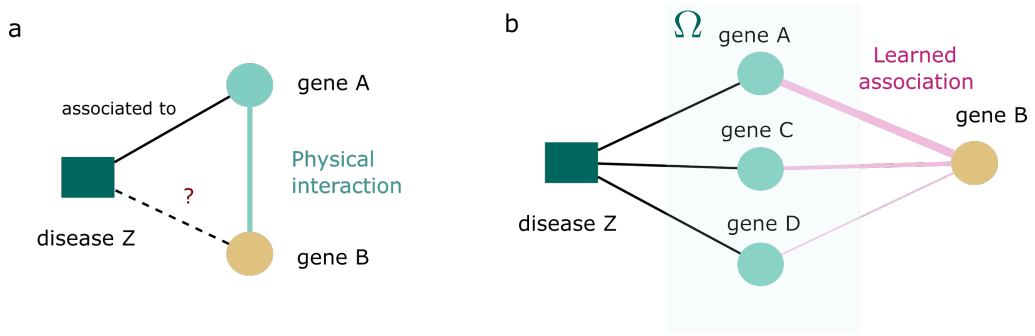

Figure 3.2: HRMC-r learns an aggregated gene-based guilt-by-association (GBA). **(a)** The GBA is established through physical protein-protein interaction; **(b)** Network diagram depicting how HRMC-r generates the prediction for a target disease-gene pair (Z, B). HRMC-r aggregates all the known disease associated genes (set Ω) and *learns* the subspace "proximity" between these genes and the target gene B. The predicted score is the sum of these learned associations.

An analogous idea can be applied to diseases and I shall refer to it as disease-based GBA. Figure 3.3a illustrates that a new disease-gene association is *established* by relating a known disease to a target disease through phenotype similarities. The idea that phenotypically similar disease tend to share disease causing genes is also based on the principles of network medicine. My HRMC-c model can be interpreted as an extension of this principle (see Figure 3.3b), where a new disease-gene association is *predicted* by relating the known diseases to a target disease through a learning process that also involves the knowledge of phenotype similarities between diseases[‡].

[‡]To understand this in detail, consider that HRMC-c generates scores through the product XC. For a given $(XC)_{ij}$, the prediction is obtained by the dot product of x_i (the ith row of X) and c_j (the jth column of C), as follow: $(XC)_{ij} = x_i c_j$. Recall that X is binary, therefore $(XC)_{ij} = \sum_{(k \in \Omega)} (c_j)_k, \Omega \in \{k | (x_i)_k > 0\}$.

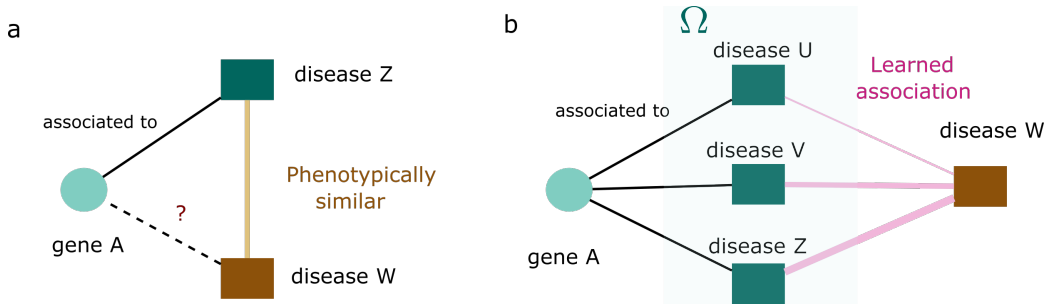

Figure 3.3: HRMC-c learns an aggregated disease-based guilt-by-association (GBA). **(a)** The GBA is established through phenotype similarity between diseases; **(b)** Network diagram depicting how HRMC-c generates the prediction for a target gene-disease pair (A, W). HRMC-c aggregates all the known diseases associated to gene A (set Ω) and *learns* the subspace "proximity" between these diseases and the target disease W. The predicted score is the sum of these learned associations.

An interesting observation, based on this interpretation of HRMC, is the implicit model assumption that each partial model carries. The HRMC-c model can provide scores only for genes that are known to be associated with a genetic disease. As a consequence, the HRMC-c imposes a strong prior on the genes that are already known to be associated with genetic diseases. A limitation of this prior is that not novel gene can be predicted with this model. Conversely, HRMC-c can predict genes for a molecularly uncharacterised disease, as long as the target disease can be related to the other diseases through phenotype similarities. Similarly, the HRMC-r model can provide scores only for the diseases that are already known to be associated with a gene, i.e. molecularly characterised. Conversely, HRMC-r model can predict novel genes associated with diseases, as long as the target gene can be related to the known genes by physical interactions. Altogether, the HRMC integrates the strength of each partial model for the prediction of potential novel genes for characterised diseases and new genes for molecularly uncharacterised diseases.

So far, I have explained the inner assumptions of the HRMC model, but I have not discussed how my model generates the learned associations between genes or diseases. Intu-

itively, we can understand the learned associations by virtue of the two terms in the numerator of the multiplicative learning algorithm (see Eqs. 3.6 and 3.7). Without loss of generality, I will focus on the HRMC-c model. The first term in the numerator of Eq. 3.6 is the data covariance, which means that a pair of diseases that are known to share disease genes would likely remain associated in C, and the strength of their association is related to the number of genes they share. Since the current gene-disease association matrix is extremely sparse, with most diseases associated to a single gene, C also needs to learn from the graph encoding phenotype similarities between diseases. In other words, the update rule incorporates additional information about diseases via disease phenotype similarities (the second term in the numerator of Eq. 3.6) and uses this side information to help discover hidden associations among diseases, which in turn can help identify new disease genes.

4 Experimental settings

4.1 Evaluation procedure

Following previous approaches[141,151,40,39,38] I used leave-one-out cross validations. In this setting, a single gene-disease association is removed and the model is trained with all the remaining associations in the matrix X (built using the 2017 snapshot). We have two test cases. The first test case, which is the typical testing scenario in the literature, assumes that diseases have known genes even after synthetic removal of a single association — referred to as *molecularly characterised* disease. There are 203 diseases that are *molecularly characterised* corresponding to 775 associations (19.25% of the associations in X). The second test case corresponds to a more difficult scenario, where the synthetic removal of the single association results in a disease without known genes — referred to as *molecularly uncharacterised*

acterised. There are 3,252 diseases in this test case scenario with a total of 3,252 associations (80.75% of the associations in X). In total, the model was trained 4,027 times for each individual test set. Each time, my model outputs a score matrix $\hat{X}$ that I used to ranked all the genes corresponding to the query disease. Following previous authors [40,39,152], the final performance of the algorithm is computed as the recall rate in the Top-N predictions for values of $N \in \{1, 10, 100, 200\}$. The recall rate is defined as follows:

$$r(N) = \frac{\text{number of gene-disease associations retrieved in top-}N\text{ predictions}}{\text{M}}.$$

where M is the total number of associations to be retrieved. $M = 775$ for the molecularly characterised test case, while $M = 3,252$ for the molecularly uncharacterised test case.

4.2 Hyperparameters tuning

The two models HRMC-c and HRMC-r were trained independently. I performed a grid search for $\alpha^c, \alpha^r \in \{0.1, 0.5, 1, 2, 3\}$, and $\beta^c, \beta^r \in \{0.5, 1, 2, 3, 4, 5, 10\}$. The parameter $p \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1\}$ was then trained using each model optimal hyperparameters. In all the experiments, $\gamma^c = \gamma^r = 10000$ and $\texttt{tolX} = 0.001$. Given a set of hyperparameters, my MATLAB implementation can finish in less than 2 hours, on a device with 32 GB RAM and 3.60GHz processor. The optimal hyperparameters found during cross validation are as follows: $\alpha^c = 0.5, \beta^c = 1, \lambda^c = 0.5, \alpha^r = 0.5, \beta^r = 0.5, \lambda^c = 0.5$ and $p = 0.70$.

5 HRMC yields accurate gene prioritisation

I compared the performance of HRMC against PRINCE[40], ProDiGe1, ProDiGe4[39] and DIAMonD[38], by means of leave-one-out cross-validation. I also include a baseline based on non-negative matrix factorisation (NMF)[29,127]. The reason to include NMF is twofold: (i) NMF is a well-known matrix decomposition technique based on non-negative constraint; and (ii) NMF is a low-rank approximation, allowing to display more clearly the advantages of using a high-rank model versus a low-rank model. Lastly, I also include Random prioritisation of genes as a baseline. Figure 3.4a shows the recall rate for distinct values of Top-N predictions for *molecularly characterised* diseases in the 2017 snapshot. My approach outperforms the baselines by $6.45 - 14.45\%$ in the top-1, $0.30 - 27.77\%$ in the top-10, $6.72 - 53.94\%$ in the top-100 and by $11.58 - 62.70\%$ in the top-200 predictions. Even in the top-1 predictions, HRMC outperforms by 6.45% to the best performing method. Intriguingly, a simple non-negative matrix factorisation of the matrix X, that does not consider complementary information of genes or diseases, performs slightly better (in the top-1) than methods based on semi-supervised learning such as PRINCE or Prodige.

Figure 3.4b shows the recall rate for distinct values of Top-N predictions for *molecularly uncharacterised* diseases. This corresponds to synthetic removal of diseases with a single gene in the 2017 snapshot (section 4.1). For this test case, only PRINCE and Prodige4 can yield predictions. In this case, my approach shows superior performance than these methods, outperforming the baselines by $6.03 - 9.06\%$ in the top-1, $19.89 - 31.02\%$ in the top-10, $28.61 - 46.99\%$ in the top-100 and by $29.08 - 48.48\%$ in the top-200 predictions. This test case scenario shows more clearly the advantages of my method.

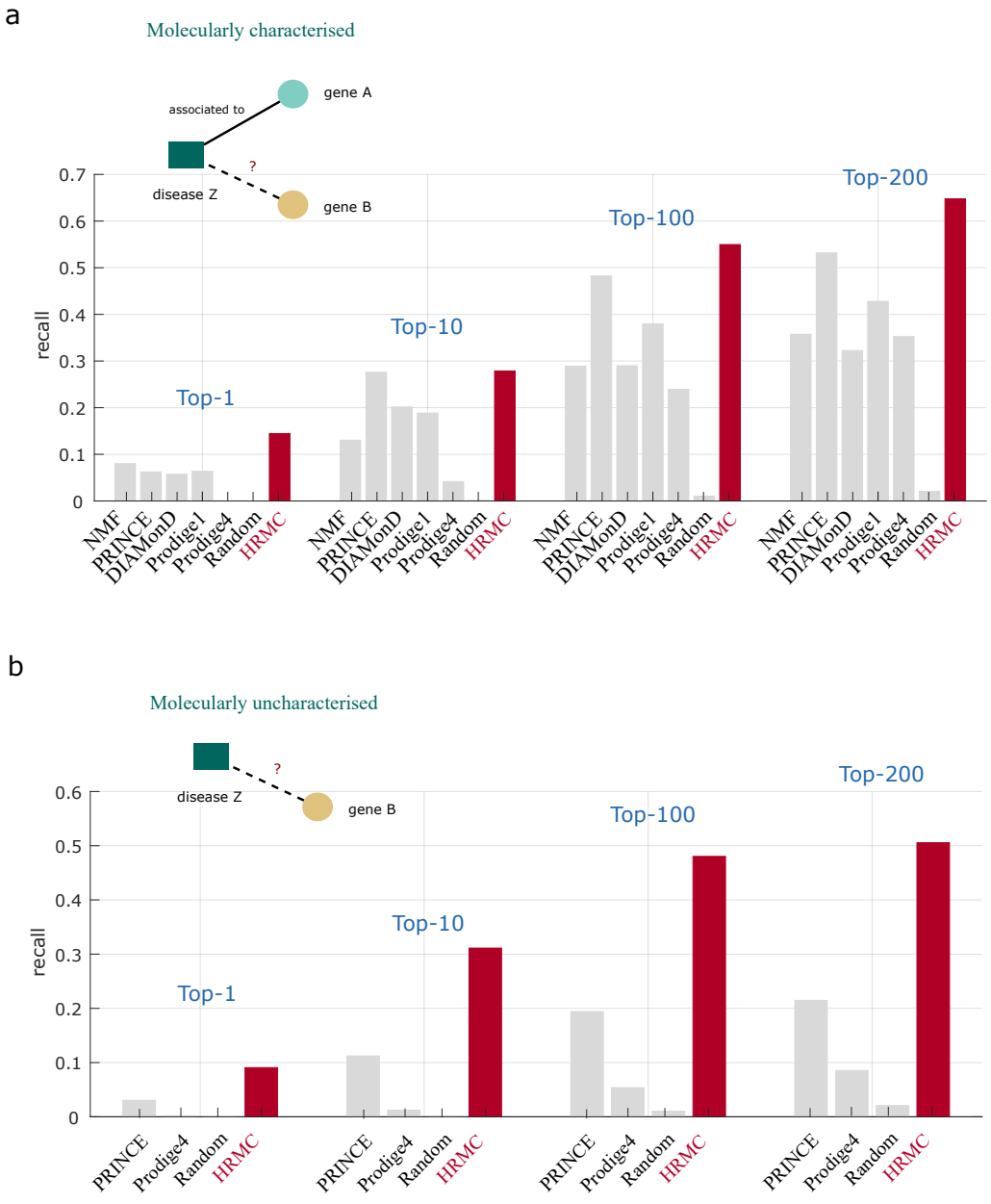

Figure 3.4: Gene prioritisation predictions. (a) Predictions from the 2017 snapshot for *molecularly characterzed* diseases. Bar height corresponds to recall at the Top-N ranked predictions, for $N \in \{1, 10, 100, 200\}$. I compared HRMC to the state-of-the-art methods PRINCE, DIAMonD, Prodige1, Prodige4 and the two baselines NMF and Random, by means of leave-one-out cross validation. (b) Predictions from the 2017 snapshot for *molecularly uncharacterised* diseases. I compared HRMC to the methods capable of such predictions, by means of leave-one-out cross validation.

I investigated whether some top predictions made by my model using the 2017 snapshot could be found in a more recent 2018 snapshot of the database. I trained my model using all the data in the 2017 snapshot with the optimal hyperparameters set by cross-validation (see section 4.2). Table 3.1 shows some top predictions made by my model for both molecularly characterised and uncharacterised diseases in our dataset. Additional evidence is provided for these pairs. For instance, schizencephaly is a congenital brain malformation characterised by infolding of cortical grey matter along a hemispheric cleft near the primary cerebral fissures. Previous molecular genetics analysis of the disorder indicates that schizencephaly is associated with mutations in the EMX2, SHH and SIX3 genes. More recently, Sato et. al.,[153] reported that the disorder is also associated with a novel mutation in COL4A1. My system has indeed reported this association in the top-1 prediction using data available up to 2017.

Top-*N*	Genetic disease	Known genes in our set	Predicted gene	Evidence
1	Schizencephaly	EMX2, SIX3, SHH	COL4A1	Sato et.al., 2019[153]
8	Budd-Chiari syndome	F5	JAK2	Mukund et. al., 2018[154]
3	Choroidal Dystrophy, central areolar	None	GUCY2D	Chen et al. 2017[155]
24	Focal cortical dysplasia, type II	None	TSC1	Lim et al., 2017[156]
51	Focal cortical dysplasia, type II	None	TSC2	Lim et al., 2017[156]

Table 3.1: Analysis of some top predictions made by HRMC using the 2017 snapshot. These top predictions were found in the 2018 snapshot of OMIM as confirmed genes associated to each corresponding genetic disorder.

7 Conclusions and Discussion

The discovery of disease genes has become a research topic gaining increasing attention over the last decades. Computational methods, based on advanced statistical and machine learning techniques together with large-scale, complicated human health data, have proven to outperform traditional approaches for tackling this problem. In this chapter, I have proposed a high-rank matrix completion algorithm for gene prioritisation. The central idea behind my approach is that high-quality predictions can be produced via an aggregated guilt-by-association principle where the associations are learned by integrating supplementary biological information about genes and diseases. The learned matrices are capable of extracting hidden relations among genes and diseases residing in the data and thus can help produce accurate and interpretable predictions.

I have formulated gene prioritisation as a high-rank matrix completion task. My model is motivated by the recent development of self-expressive models[8,113,9]. Elhamifar[8] has proposed self-expressive models for simultaneously clustering and completion of incomplete high-dimensional data. In principle, a self-expressive model presumes that one datapoint can be efficiently reconstructed by few other datapoints belonging to a common subspace[113], or equivalently, that each column of the data matrix X can be represented as a combination of a few other columns[9]. Such models, therefore, can generalise low-rank approximation models. Although my model is inspired by self-expressive models, it differs from them as I assume that our data matrix is *fully* – rather than partially – observed while its entries are noisy.

My approach has a number of advantages over many existing gene prioritisation algo-

rithms. Network-based methods heavily depend on the topology of the protein-protein interaction networks, so information about gene-disease associations, which is typically carried by gene labels and random walkers, cannot be effectively transferred between two genes (or sets of genes) that are not linked. My method, however, is able to produce scores for genes that share no similarity with other genes, that is, that is not connected with other nodes in the gene interaction network. Furthermore, the majority of the approaches are not applicable to predict genes for *molecularly uncharacterised* diseases since no initial seed is available. I show that my approach, however, is able to produce accurate predictions in this more challenging task.

An important aspect of my model is that it favours model interpretability. The prediction can be explained as an extension of the guilt-by-association principle where the associations between diseases or genes are learned by the *sparse self-representation* matrices C and R, respectively. Furthermore, due to the fact that my objective function is guaranteed to converge to a globally optimum solution, these sparse coefficients are also *reproducible* under arbitrary random initialisation of the weights: a desirable property for reproducible biological interpretations. The ability of my model to produce interpretable and reproducible predictions is critical for healthcare[27], since predictions made by the model can be understood and validated by molecular biologists. Furthermore, my model does not trade between accuracy and interpretability to produce the predictions. Motivated by my findings, I provided a complete list of predictions made by my model in Tables 1-4 — it includes novel predictions for over 2,313 *molecularly uncharacterised* genetic disorders. These can be freely accessed at `http://www.paccanarolab.org/hrmc-gene/`.

If we knew how to do it, we'd have already done it

Reed Hastings, chief executive of Netflix, 2006

4

Top-N recommender systems

The advent of e-commerce and the increasing amount of users and products poses a difficult challenge to recommendation systems. In particular, in this chapter, I focus on the problem of top-*N* recommender systems, that aims at accurately predicting user preference in a small set of *N* items[157,20]. This setting is common in online entertainment platforms

such as Netflix as well as in commercial platforms such as Amazon. The problem of top-N recommender system is often formulated in terms of a high-dimensional user-item feedback matrix $X \in \mathbb{R}^{n \times m}$ for n users and m items where the entry x_{ij} indicates if the user i has a preference towards an item j. The goal of a top-N recommender system is to predict – in a short list of N items – the missing preference(s) for a given user. Given that X tends to be highly sparse, it is also common to incorporate domain-specific knowledge about users and items — side information — to improve the prediction[158,159,26,160]. For instance, in movies recommendation systems, one can exploit the features extracted from movie plots or user characteristics[161].

Existing algorithms for top-N recommendations can be divided into two main categories[162,20,163,42,164,165]. The first group of algorithms, known as neighbourhood models, assume that datapoints (rows or columns of X) are *locally* related according to a specific definition of proximity, e.g. Pearson correlation. However, the performance of these methods often depends on how proximity is defined — usually by well-defined heuristics. The second group of algorithms, known as latent factor models, assume that data lie in a single low-dimensional subspace. These typically assign a low-dimensional feature vector to each user and a low-dimensional feature vector to each item such that the dot-product between these vectors characterise the user-item preference. PureSVD[20], for instance, learns the latent factors via truncated Singular Value Decomposition (SVD).

A novel class of item-based algorithms based on Sparse Linear Method (SLIM)[43] have been recently proposed for top-N recommendations[158,166,167,168,169,170]. SLIM aims to learn a sparse zero-diagonal matrix of coefficients $W \in \mathbb{R}^{m \times m}$ for items such that $X \approx XW$, where the null diagonal aims at ruling out the trivial solution $W = I$. Several SLIM-based algo-

rithms have already been proposed. For instance, models that integrate side information[158], contextual-aware information[167] or those that account for high-order relationships between items[170]. SLIM solves the optimisation problem by a two-metric projection method[171]. This algorithm has two main limitations: (i) it requires to set a learning rate and; (ii) it uses a projection function to guarantee the non-negative constraint. Furthermore, from the theory of constrained optimisation standpoint, there is no theoretical guarantees of convergence of the algorithm.

I have drawn a connection between SLIM-based models and Self-Expressive Model (SEM)[8]. In fact, SLIM is a SEM that learns items self-representations such that each item (a column of X) is represented as a linear combination of few other items. SEM has recently been proposed as a framework for simultaneously clustering and completing high-dimensional data lying in the union of low-dimensional subspaces. Thus, often leading to a high/full rank matrix. SEM also covers low-rank models as a special case. The difference between SLIM and previous SEM models[82] is that SLIM assumes that X is *fully* — rather than *partially* – observed while its entries are noisy.

Most of the algorithmic developments in this chapter has been used in Chapter 1, for the drug side effect prediction problem, and in Chapter 2, for the disease gene prediction problem. In this chapter, I propose a Non-negative Self-Expressive Model (NSEM) starting from the SLIM formulation. I showcase novel mutiplicative algorithms with globally optimal guarantees of convergence, and show results outperforming previous state-of-the-art formulations. I further show that the items self-representations are interpretable in terms of popularity and novelty metrics. Theoretical results are supported by empirical evaluations in Movielens, Netflix and three Amazon datasets.

In this section, I introduce Sparse Linear Method (SLIM) from the perspective of Self-Expressive Models (SEM). The goal of SLIM is to represent each item vector approximately as a weighted linear combination of a small number of other item vectors. Each item vector $x_i \in \mathbb{R}^m$ is represented using other item vectors $x_1, x_2, \ldots, x_m \in \mathbb{R}^m$ and a sparse vector of weights or coefficients $w \in \mathbb{R}^m$ such that $x_i \approx \sum_{j \neq i} x_j w_j$. In matrix form: let $X \in \mathbb{R}^{n \times m}$ be the user-item matrix (each column is an item vector), let $W \in \mathbb{R}^{m \times m}$ be the sparse coefficient matrix (each column is a coefficient vector). $x_{ij} = 1$ if the user i is associated* to item j, or zero otherwise. SLIM assigns scores to missing user-item associations by the following linear model:

$$\hat{X} \approx XW \tag{4.1}$$

where W is the solution of the following optimisation problem:

$$\begin{aligned} \min_{W} \quad & \underbrace{\frac{1}{2}\|X - XW\|_F^2}_{\text{self-representation}} + \underbrace{\frac{\beta}{2}\|W\|_F^2 + \lambda\|W\|_1}_{\text{sparsity}} \\ \text{subject to} \quad & W \geq 0, \ \text{diag}(W) = 0. \end{aligned} \tag{4.2}$$

where $\|.\|_F$ denoting the Frobenius norm, $\beta, \lambda > 0$ are regularisation parameters. I further denote the cost function in Eq. 4.2 by $\mathcal{Q}_{\text{SLIM}}(W)$.

The first term in Eq. 4.2 is the *self-representation constraint*, which aims at learning a matrix of coefficients W such that XW is a good reconstruction of the original matrix X. The second term is the *sparsity constraint*, which uses the elastic-net regularisation known to

*In this study, we are limited to an implicit feedback representation of the user-item interactions.

impose sparsity and robustness to noise[114,115]. The norms are defined as follows:

$$\| W \|_1 = \sum_{i=1}^{m} \sum_{j=1}^{m} |w_{ij}|, \quad \| W \|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{m} |w_{ij}|^2$$

Additional constraints are to prevent the trivial solution $W = I$ by imposing $\text{diag}(W) = 0$ (together with $W \geq 0$).

SLIM solves the constrained optimisation problem in Eq. 4.2, for each column of W independently, using a two-metric projection method[172]. That is, at each iteration, a search direction is computed that is a combination of a Newton method and a scaled steepest-descent step. Under this procedure, the update rule takes the form $W^{t+1} = \mathcal{P}_W(W^t - \eta^t(\nabla^2 J(W^t))^{-1}\nabla J(W^t))$, where η^t is the stepsize at t iteration and $\mathcal{P}_W(.)$ is the projection function required to guarantee non-negativity $W \geq 0$ and null-diagonal $\text{diag}(W) = 0$. SLIM estimates the stepsize at each iteration using the Armijo/backtraking approximation rule.

One disadvantage of SLIM is that it does not allow to integrate any relational prior about users or items — side information. cSLIM[158], is a variation that has been proposed to address this problem. The motivation is that complementary information about items can help to establish relationships between items that cannot be inferred from past users behaviour. Typically, this is crucial in a cold-start scenario[161]. For instance, let $Z \in \mathbb{R}^{p \times m}$ represent a movie side information matrix containing words in a corpus of movie plots descriptions (rows) for each movie (columns). $z_{qj} = 1$ if the word q appears in the movie plot

j, or is $z_{qj} = 0$ otherwise. cSLIM aims to solve the following optimisation problem:

$$\begin{aligned} &\min_{W} \mathcal{Q}_{\text{SLIM}}(W) + \frac{\alpha}{2} \parallel Z - ZW \parallel_F^2 \\ &\text{subject to } \; W \geq 0, \; \text{diag}(W) = 0. \end{aligned} \tag{4.3}$$

where the regularisation parameter $\alpha > 0$ is tuned to adjust for the confidence on the side information. The user-item score matrix is then computed as in Eq. 4.1. cSLIM can be optimised using the SLIM two-metric projection method by replacing X in Eq. 4.2 by $Y \in \mathbb{R}^{(n+p)\times m}$ defined as $Y = [X, \sqrt{\alpha}Z]^T$.

1.1 Non-negative Self-Expressive Model (NSEM)

My first goal is to obtain a *smooth* objective function, both continuous and differentiable in the feasible region $W \geq 0$. To this end, I propose a relaxation of the null-diagonal constraint using a regularised trace operator $\gamma \, \text{Tr}(W) \equiv \gamma \sum_i W_{ii}$ such that $\text{diag}(W) \to 0$ when $\gamma \gg 0$. I shall refer to my model as NSEM, which aims to minimise the following cost function:

$$\min_{W \geq 0} \; \underbrace{\frac{1}{2}\|X - XW\|_F^2}_{\text{self-representation}} + \underbrace{\frac{\beta}{2}\|W\|_F^2 + \lambda\|W\|_1}_{\text{sparsity}} + \underbrace{\gamma \, \text{Tr}(W)}_{\text{null diagonal}} \tag{4.4}$$

I denote the cost function in Eq. 4.4 by $\mathcal{Q}_{\text{NSEM}}(W)$. It is straightforward to show that this function is smooth. I prove the smoothness of my function in the following lemma.

Lemma 2 (Smoothness). *The cost function $\mathcal{Q}_{NSEM}(W)$ in Eq. (4.4) is smooth in the feasible region $W \geq 0$.*

Proof. I need to prove that all the derivatives $\Delta^p \mathcal{Q}_{\text{NSEM}}(W)$, for any $p > 0$, exist. For $W \geq 0$, we get

$$\begin{aligned}
\Delta^1 \mathcal{Q}_{\text{NSEM}}(W) &= -X^\top (X - XW) + \beta W + \lambda + \gamma I \\
\Delta^2 \mathcal{Q}_{\text{NSEM}}(W) &= X^\top X + \beta \\
\Delta^{p>2} \mathcal{Q}_{\text{NSEM}}(W) &= 0
\end{aligned} \tag{4.5}$$

□

In the following lemma, I prove that my objective function is strictly convex.

Lemma 3 (Convexity). *For a real matrix Y, the objective function in Eq. 4.4 is strongly convex in the feasible region $W \geq 0$.*

Proof. I need to show that the Hessian of $\Delta^2 \mathcal{Q}_{\text{NSEM}}(W) \succ 0$ is a positive definite matrix[90]. The Hessian is positive definite iif for a non-zero column vector $h \in R^m$ of real numbers the scalar $h^T \nabla^2 \mathcal{Q}_{\text{NSEM}}(W) h > 0$, which can be written as follow:

$$\begin{aligned}
h^T \nabla^2 \mathcal{Q}_{\text{NSEM}}(W) h &> 0 \\
h^T (X^\top X + \beta) h &> 0 \\
h^T X^\top X h + \beta h^T h &> 0 \\
\| Xh \|^2 + \beta \| h \|^2 &> 0 \quad \forall h, \beta > 0
\end{aligned} \tag{4.6}$$

That is, my objective function in Eq. 4.4 is strictly convex on $W \geq 0$. □

I can now prove the following theorem.

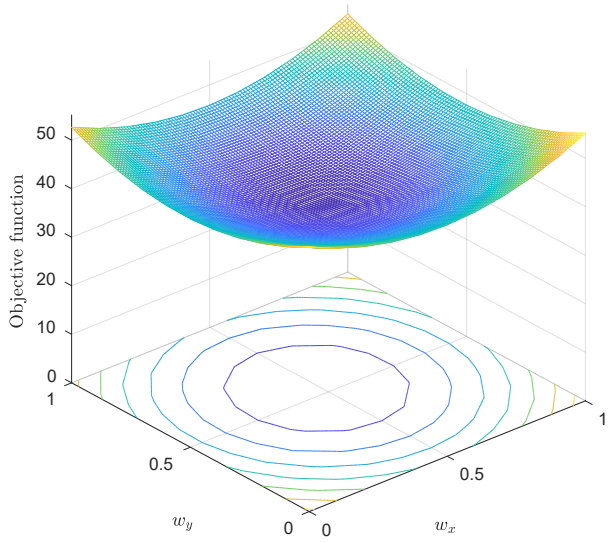

Figure 4.1: Simulated NSEM objective function $\mathcal{Q}_{\text{NSEM}}(w_x, w_y)$. Example for a binary random matrix $Y_{100\times 2}$ and parameters $(\beta, \lambda, \gamma) = (0.1, 0.1, 10^4)$. The convex function is plotted as a function of the off-diagonal elements of W. The contour is also shown.

Theorem 5 (Uniqueness). *The objective function in Eq. 4.4 has an unique solution on the convex set* $W \geq 0$.

Proof. From lemmas (2) and (3), since my objective function is strictly convex on $W \geq 0$ and $W \geq 0$ is a convex set, then the optimal solution (assuming it exists) must be unique. □

I illustrate a toy problem for a 2×2 matrix $W = \begin{bmatrix} w_x & 0 \\ 0 & w_y \end{bmatrix}$ in Figure 4.1. In this simulated example, I evaluated Eq. 4.4 on a grid $w_x, w_y \in [0, 1]$. The objective function is convex in the feasible region $w_x, w_y \geq 0$.

1.2 NSEM multiplicative algorithm

To solve the NSEM objective function in Eq. 4.4, I propose the following multiplicative learning algorithm:

$$w_{ij} \leftarrow w_{ij} \frac{(X^\top X)_{ij}}{(X^\top X W + \beta W + \lambda + \gamma I)_{ij}} \tag{4.7}$$

The advantage of my algorithm is that it is guaranteed to converge to a global minimun under the non-negative constraint $W \geq 0$. I shall prove this in the following theorem.

Theorem 6 (Global minimum solution). *The objective function in Eq. 4.4 converges to a global minimum under the update rule in Eq. 4.7.*

Proof. From the theory of constrained optimisation[89], we need to show that my algorithm satisfies the Karush-Khun-Tucker (KKT) complementary conditions, which are both necessary and sufficient conditions for a global solution point given the convexity of the cost function (lemma 3)[89,90]:

$$w_{ij} \geq 0, \quad (\nabla \mathcal{Q}(W))_{ij} w_{ij} = 0$$

The first KKT condition holds if $w_{ij}^{t=0} \geq 0$. For the second KKT condition, we have

$$\begin{aligned} \nabla \mathcal{Q}(W))_{ij} w_{ij} &= 0 \\ (X^\top X W + \beta W + \lambda + \gamma I)_{ij} w_{ij} - (X^\top X)_{ij} w_{ij} &= 0 \end{aligned} \tag{4.8}$$

at local minimum $W = W^*$. From equation (4.7), the multiplicative update rule at convergence is:

$$w_{ij}^* = w_{ij}^* \frac{(X^\top X)_{ij}}{(X^\top X W^* + \beta W^* + \lambda + \gamma I)_{ij}}$$

which is identical to equation (4.8). That is, the multiplicative rule converges to a global minimum. □

My algorithm in Eq. 4.7 has several advantages over the two-metric projection method used by SLIM. First, it does not require to set a learning rate nor use projection functions to guarantee non-negativity. Second, it is guaranteed to converge to a globally optimal solution point under arbitrary initialisation of $W \geq 0$. Although SLIM's two-metric projection might provide an approximation to the solution of the objective in Eq. 4.2, it is unfeasible to guarantee its optimal convergence in the feasible region $W \geq 0$. Furthermore, similar to standard optimisation methods such as conjugate gradient descent or gradient ascend, I can also show that my algorithm is a first-order algorithm.

Theorem 7 (Rate of convergence). *The multiplicative update rule in Eq. 4.7 has a first-order convergence.*

Proof. Following the procedure used in [89,118], we can assume that when W is close to optimal, it has a linear convergence rate and we can represent the updating algorithm as mapping $W^{t+1} = \mathcal{M}(W^t)$ with fixed point $W^* = \mathcal{M}(W^*)$. Then, when W^{t+1} is near W^*, we have $W \simeq \mathcal{M}(W^*) + \nabla\mathcal{M}(W)(W - W^*)$ subject to $W \geq 0$, and thus: $\| W^{t+1} - W^* \| \leq \| \nabla\mathcal{M}(W) \| \cdot \| W^t - W^* \|$ with $\| \nabla\mathcal{M}(W) \| \neq 0$ almost surely. That is, the multiplicative update rule is a first-order algorithm. □

At a first glance, the relaxation of the null diagonal constraint in Eq. 4.4 might seem as a disadvantage as we include an additional hyperparameter that requires proper tuning. However, I will now show that this is not the case, as there is an approximate lower bound

for the parameter γ that facilitates its setting. A lower bound can be obtained from my multiplicative learning algorithm in terms of a maximum tolerable error in the main diagonal of W, as follow:

Theorem 8 (Theoretical bounds for γ). *The null diagonal penalty γ is bounded in terms of a maximum small value $\varepsilon > 0$ in diag(W) by $(\frac{\sigma^{\frac{1}{2N}}\mathcal{C}_{max}}{\varepsilon^{\frac{1}{N}}}, \infty)$, where $\sqrt{\sigma}$ is the maximum initial value of W, N is the maximum number of iterations and $\mathcal{C}$ is the maximum value in the main diagonal of the data covariance, $\mathcal{C}_{max} = \max_i$ diag$(X^\top X)$.*

Proof. From the multiplicative learning rule in Eq. 4.7, we shall notice that for a $\gamma \gg 0$ we can approximate the maximum value of the diagonal of W at each iteration j as follows:

$$\frac{\sqrt{\sigma}\mathcal{C}^j_{max}}{\gamma^j}$$

Let's now assume that at the p_{th} iteration, we tolerate a maximum small value $\varepsilon \to 0$ in the diagonal elements of W, i.e.

$$\varepsilon = \frac{\sqrt{\sigma}\mathcal{C}^p_{max}}{\gamma^p} \tag{4.9}$$

Solving (4.9) for γ provides a lower bound in terms of the tolerable value ε,

$$\gamma(\varepsilon, p) > \frac{\sigma^{\frac{1}{2p}}\mathcal{C}_{max}}{\varepsilon^{\frac{1}{p}}} \tag{4.10}$$

The upper bound can be obtained when $\varepsilon \to 0$, which causes $\gamma(\varepsilon, p) \to \infty$. □

1.3 Extending NSEM to collective SLIM

My approach to solve SLIM can be easily extended to cSLIM. I begin by extending the objective function to my *smooth* formulation:

$$\min_{W \geq 0} \mathcal{Q}_{\text{NSEM}}(W) + \frac{\alpha}{2} \parallel Z - ZW \parallel_F^2 \tag{4.11}$$

where $Z \in \mathbb{R}^{p \times m}$ represent a movie side information matrix containing words in a corpus of movie plots descriptions (rows) for each movie (columns). $z_{qj} = 1$ if the word q appears in the movie plot j, or is $z_{qj} = 0$ otherwise.

I shall refer to the model in Eq. 4.11 as collective NSEM (cNSEM), for which a similar multiplicative learning algorithm can be obtained:

$$w_{ij} \leftarrow w_{ij} \frac{(X^\top X + \alpha Z^\top Z)_{ij}}{(X^\top XW + \alpha Z^\top ZW + \beta W + \lambda + \gamma I)_{ij}} \tag{4.12}$$

Ning and Karypis[43] hypothesised that W is learning *proximity* between items in a similar fashion than other neighbourhood models such as itemkNN. My algorithm, that learns a globally optimal W, provides more accurate interpretation of the learned coefficients in W. At each iteration, W updates proportionally to the items covariance matrix $X^\top X$. The immediate consequence is that only co-purchased items $(X^\top X)_{ij} > 0$ can belong to the same subspace as $w_{ij} = 0$ if $(X^\top X)_{ij} = 0$; regardless of the regularisation parameters. The *proximity* between the items then depends on the product between the previous w_{ij}^{t-1} and a regularised version of the item covariance matrix. Note also that the sparse self-representation matrix W is not naturally symmetric (columns are learned independently in Eq. 4.7), which

means that my algorithm recovers the subspace-sparse representation of each item in each column of W. We can now understand how the side information can help to capture hidden relationships between items which cannot be inferred from past user behaviors. Looking at the numerator of Eq. 4.12, zero entries in the covariance matrix $(X^\top X)_{ij} = 0$ are filled in by the weighted item side information covariance $\alpha Z^\top Z$. In order to be useful for the prediction, the item feature vector Z must encode relevant information about the problem, e.g. movie plot descriptions.

1.4 Algorithm complexity and stopping criteria

My algorithm is similar to the multiplicative learning rules from non-negative matrix factorisation (NMF)[29]. Therefore, for the implementation shown in Algorithm 2, I followed the recommended guidalines for NMF in[91]. W was initialise with uniformly distributed random weights in the interval $(0, \sqrt{\sigma})$. A small value $\varepsilon \simeq 1 \times 10^{-16}$ was also added to the denominator to prevent division by zero. My multiplicative algorithm preserves all the desirable properties of the original SLIM algorithm[43]. First, my multiplicative algorithm can also be optimised for each column of W independently. In fact, given that the NSEM update rule in Eq. 4.7 depends on the data covariance $X^\top X$, we can decouple the optimisation for each column jth as follows

$$w_j \leftarrow w_j \frac{(X^\top X)_j}{X^\top X w_j + \beta w_j + \lambda + \gamma \mathbb{1}_j} \tag{4.13}$$

where w_j is the jth column of W, the division is element-wise and the indicator function $\mathbb{1}_j = 1$ for the jth element of w_j (corresponding to the main diagonal of W), or zero otherwise. This allows for a fast parallel computation of the algorithm for each item indepen-

dently; or when the recommendations wants to be produced for only a subset of the items. Second, as it happens for the original SLIM algorithm, my algorithm can also exploit the sparsity in the data and in W to offer fast recommendations. The complexity to deliver recommendations for an user u is $\mathcal{O}(nnz_u + nnz_w + m \log m)$, where nnz_u is the number of non-zero elements in y_u and nnz_w is the average number of non-zero elements in the rows of W, and $m \log m$ is the cost of sorting the items.

ALGORITHM 2: NSEM learning algorithm

Given $\beta, \lambda, \sigma, \gamma, \varepsilon > 0$

```
W = rand(m)*sqrt(σ); % initialisation
I = eye(m); % identity matrix
C = X'*X; % numerator
for iter = 1:maxiter do
    denominator = C*W + β*W + λ + γ*I + ε;
    W = W .* (C ./ denominator);
end
```

One advantage of my multiplicative algorithm is that it can further exploit the sparsity in the covariance matrix $X^\top X$ to reduce the computational complexity. From Eq. 4.7 is clear that given the element-wise multiplication between W and $X^\top X$ at each iteration, W can have non-zero weights only in the entries where the covariance has non-zero values. Therefore, the entries in W corresponding to the zero elements in $X^\top X$ can be set to zero and thus reducing the number of operations at each iteration. In fact, the most expensive operation in my algorithm comes from the denominator term $X^\top XW$, which can be reduced from $\mathcal{O}(m^3)$ to $\mathcal{O}(m \times nnz_x \times nnz_w)$ by setting $w_{ij} = 0$ for $(X^\top X)_{ij} = 0$ at $t = 0$ — nnz_x is the average number of non-zero elements per row in the covariance and nnz_w is the average number of non-zero elements per column in W.

The stopping criteria for the algorithm is (i) when the element-wise change $\delta_W^{(t)}$ between

$W^{(t+1)}$ and $W^{(t)}$ is smaller than a predefined tolerance `tolX`; or (ii) when the number of iterations reaches `maxiter`. $\delta_W^{(t)}$ is computed as follow;

$$\delta^{(t)} = \max \left(\frac{|W_{ij}^{(t+1)} - W_{ij}^{(t)}|}{\max_{(i,j)\in\Omega} |W_{ij}^{(t)}| + \varepsilon} \right) \tag{4.14}$$

2 Empirical results

2.1 Datasets

I used five public datasets, namely Movielens[173], Netflix[174], and three Amazon datasets[175] on different categories: office products, instant video, and sports and outdoors products (see Table 4.1). For Netflix, I extracted a representative subset while ensuring that each user and each item had known associations, using a procedure similar to the one used in [43,158]. In the Netflix subset, each user rated 10-116 movies and each movie was rated by 5-914 users. For the Amazon datasets, I used their 5-core subsets where each user and item has at least 5 known associations. Furthermore, following the procedure in [43], I transformed explicit feedback datasets into implicit feedback datasets. That is, whenever rating value was provided, e.g. 1-5 stars rating system, I simply use a value of 1 for every rating value or 0 otherwise. To compare cSLIM and cNSEM, I used the same side information as in the original cSLIM publication[158] which consisted on movie plots for Movielens. I extracted movie plots from the Internet Movie Database (IMDb)[176], as summarized in Table 4.1. I then pre-processed the plots to remove stop words and then converted each word to its stem[†]. I also considered only words that appeared in at least five plots.

[†]i used the Python Natural Language Toolkit (NLTK) 3.2.5 https://www.nltk.org/.

Table 4.1: Public datasets

dataset	#users	#items	#nnzs	density
Movielens	943	1,682	100,000	6.30%
Netflix	9,948	3,995	463,484	1.66%
Office	4,905	2,420	53,258	0.45%
Video	5,130	1,685	37,126	0.38%
Sports	35,598	18,357	296,337	0.045%
side info	#movpp	#words	#nnzs	density
IMDb ML	1568	1027	15,131	0.88%

The columns #users, #items and #nnzs indicate the number of users, items and non-zero entries, respectively. The density is the percentage of #nnzs from the total number of entries, i.e. density = #nnzs / (#users×#items). The column #movpp indicates the number of movies that could be mapped to IMDb using the movie title provided, #words is the unique number of words in the corpurs of movies.

2.2 Evaluation procedure

To evaluate the performance of the methods, I followed the standard procedure used to compare top-N recommendation systems[43,158]. For each dataset, I applied five times Leave-One-Out cross validation (LOOCV). In each run, each of the datasets is split into a training set $\mathcal{T}^{train}$ and a testing set $\mathcal{T}^{test}$ by randomly selecting one of the non-zero entries of each user and placing it into the $\mathcal{T}^{test}$. The cardinality of the test set is the total number of users, $|\mathcal{T}^{test}| = \#\text{users}$. $\mathcal{T}^{train}$ is then used to train a model. Then for each user, a size-N ranked list of recommended items is generated by the model. The evaluation is performed by comparing the recommendation list of each user and the item of that user in $\mathcal{T}^{test}$. The recommendation performance is then computed using the Hit Rate (HR) and the Average Reciprocal Hit Rate (ARHR). HR is defined as the fraction of users for which the items in $\mathcal{T}^{test}$ were

retrieved in the top-N recommendations as follows:

$$\mathrm{HR}(N) = \frac{\#\text{hits@top-}N}{\#\text{users}} \tag{4.15}$$

ARHR gives importance to the *position* (r_j) at which the recommended item appeared, and is defined as;

$$\mathrm{ARHR}(N) = \frac{1}{\#\text{users}} \sum_{j=1}^{\#\text{hits@top-}N} \frac{1}{r_j}. \tag{4.16}$$

For both measures, I reported the mean HR ($\overline{\mathrm{HR}}$) and the mean ARHR ($\overline{\mathrm{ARHR}}$) of the five repetitions.

2.3 Hyperparameters tuning

I ran SLIM using the code provided by Ning and Karypis[43172]. For both methods, I optimised $\beta, \lambda \in \{0, 0.1, 0.5, 1, 2, 3, 4, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 100\}$ and for cSLIM and cNSEM I used $\alpha \in \{0.1, 0.3, 0.5, 0.7, 1\}$. In the NSEM and cNSEM algorithms, I set $\gamma = 10^4, \varepsilon = 10^{-16}, \sigma = 0.01$ for all the datasets. The convergence tolerance was set to $\texttt{tolX} = 10^{-2}$. The convergence occurred in less than 50 iterations in all the datasets.

2.4 Performance Comparison

Table 4.2 summarises the performance of the methods for the different datasets in the top ten recommendations ($N = 10$). We observed that NSEM is consistently better than SLIM in all the datasets regarding both $\overline{\mathrm{HR}}$ and $\overline{\mathrm{ARHR}}$. In terms of $\overline{\mathrm{HR}}$, NSEM is better than

Table 4.2: Perfomance in the top-N ($N = 10$) recommendations

Dataset	params		$\overline{\text{HR}}$		$\overline{\text{ARHR}}$	
	NSEM	SLIM	NSEM	SLIM	NSEM	SLIM
Movielens	$3, 0$	$25, 0$	0.348	0.32	0.160	0.149
Netflix	$1, 0$	$15, 4$	0.171	0.156	0.080	0.069
Office	$2, 0.5$	$15, 0$	0.116	0.108	0.056	0.049
Video	$2, 0$	$15, 0$	0.346	0.328	0.188	0.171
Sports	$0, 2$	$25, 0$	0.082	0.080	0.040	0.036

Columns corresponding to params represent the model parameters (β, λ) that were set for the top ten ($N = 10$) recommendations. HR indicates the mean Hit Rate, ARHR the mean Average Reciprocal Hit Rate.

SLIM by 6.1% in Movielens, 9.62% in Netflix, 7.41% in Office, 5.49% in Video and 2.5% in Sports. Similar percentage improvements are observed in terms of $\overline{\text{ARHR}}$: by 7.38% in Movielens, 15.94% in Netflix, 14.29% in Office, 9.94% in Video and 11.11% in Sports. This implies that NSEM ranks relevant items significantly higher than SLIM in all the datasets. In models that integrate side information (cNSEM vs cSLIM), we obtained similar performance gains. In terms of $\overline{\text{HR}}$, cNSEM is better than cSLIM by 12.14% (0.351 for cNSEM and 0.313 for cSLIM). In terms of $\overline{\text{ARHR}}$, cNSEM is better than cSLIM by 19.85% (0.163 for cNSEM and 0.136 for cSLIM).

2.5 Recommendation at different Top-N

To verify whether the improvements were consistent for different top-N recommendations, I tested the performance of the algorithms for different values of $N \in \{5, 10, 15, 20, 25\}$. Table 4.3 shows the performance of the methods. We observe that at different top-N recommendations, NSEM performs significantly better than SLIM in earlier retrieval of relevant items: by 5.26-11.21% better across datasets in the top-5 recommendations, by 2.5-9.62%

across datasets in the top-10 recommendations, and by 0-8.81% across datasets in the top-15 recommendations. The performance improvement was consistently better across top-Ns in the Netflix dataset: by 7.17-11.21% better, while slightly decreasing in the Sports dataset as N increased.

3 Sensitivity analysis of hyperparameters

A critical question is whether the model's performance is robust with respect to the specific choice of the model parameter. This was only partially addressed in [43] and for one small dataset only. I have extensively studied this for both algorithms NSEM and SLIM for an initial grid of $\beta, \lambda \in \{0, 0.1, 0.5, 1, 2, 3, 4, 5\}$. Figure 4.2a compares the performance of NSEM and SLIM for the top-10 recommendations in the different datasets. At a first glance, one can observe that the performance of NSEM is very robust to the specific setting of the parameters β and α. Optimal performance with NSEM is obtained for small values of β and α and equally accurate models can be obtained for a wider range of parameters. This is not the case for SLIM, whose optimal performance seems not near to be found — possibly in $\beta > 5$ and $\lambda \geq 2$. By increasing the grid search for SLIM (Figure 4.2b), we can find that its optimal performance across datasets lie in a specific window of $10 \leq \beta \leq 40$ and $0 \leq \lambda < 5$. SLIM thus requires a much finer tuning of model parameters to achieve optimal recommendation performance.

3.1 Parameter-free model

Furthermore, while both $\beta > 0$ and $\lambda > 0$ are critical for obtaining a good recommendation performance in SLIM, in NSEM, even if one (or both) parameters are zero, the

Table 4.3: Perfomance at different top-N recommendations

	N									
Dataset	5		10		15		20		25	
	NSEM	SLIM	NSEM	SLIM	NSEM	SLIM	NSEM	SLIM	NSEM	SLIM
Movielens	0.245	0.227	0.348	0.328	0.415	0.399	0.462	0.447	0.502	0.491
Netflix	0.119	0.107	0.171	0.156	0.210	0.193	0.243	0.225	0.269	0.251
Office	0.084	0.076	0.116	0.108	0.138	0.131	0.155	0.152	0.169	0.169
Video	0.278	0.256	0.346	0.328	0.384	0.373	0.412	0.406	0.433	0.433
Sports	0.060	0.057	0.082	0.080	0.096	0.096	0.106	0.110	0.115	0.122

Columns corresponds to the $\overline{\text{HR}}$ of the methods at different values of top-N recommendations.

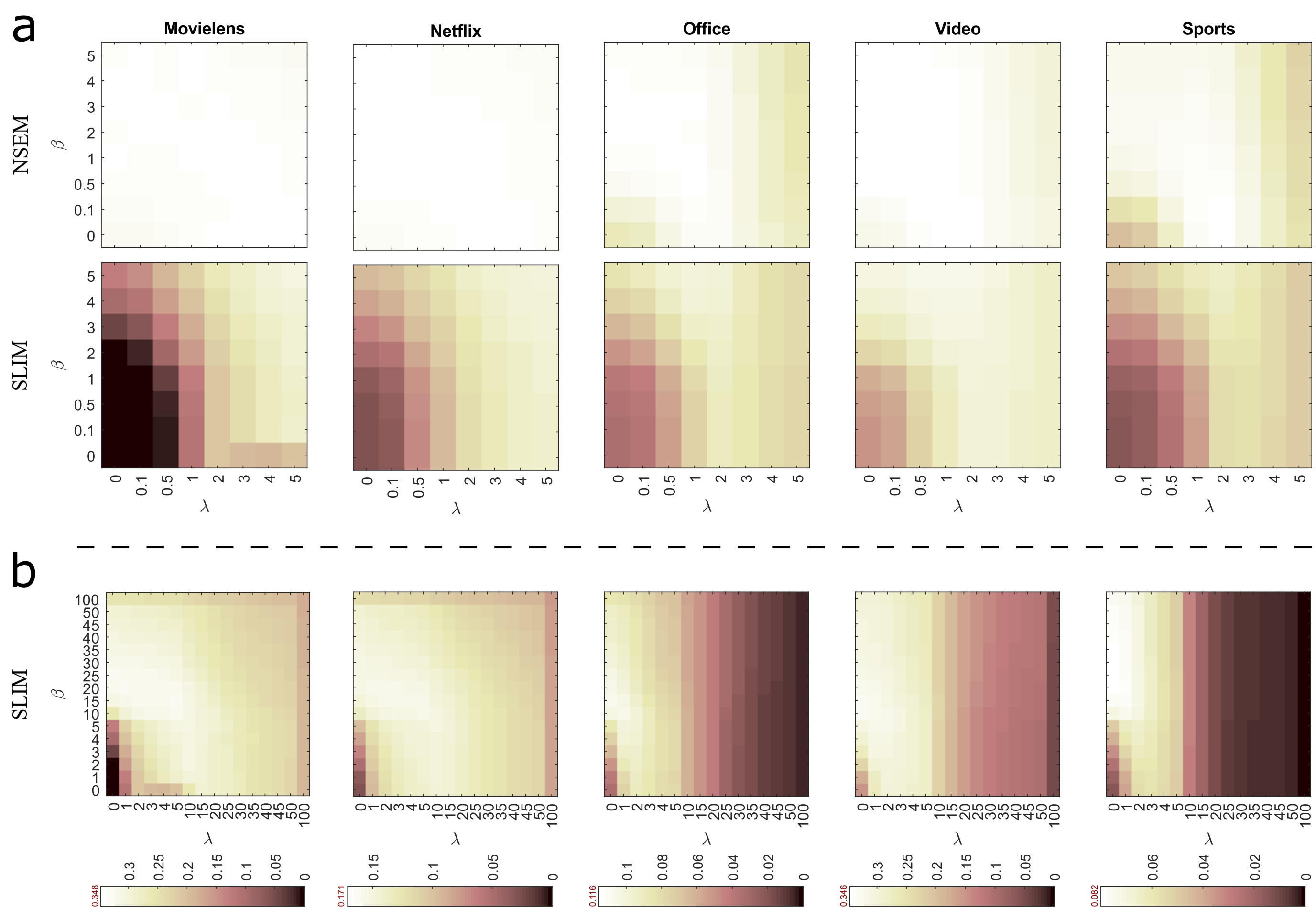

Figure 4.2: Heatmaps of the performance sensitivity to model parameters in terms of $\overline{\text{HR}}$ in the top 10 recommendations.

performance of the model seems unaffected. In detail, when $\beta = \lambda = 0$, NSEM prevents over-fitting without great loss in performance (Fig. 4.3) which ranges between 75.33% and 96.11% of the optimal performance in Movielens, Netflix, Office and Video, while for the sparser Sports dataset, NSEM achieves 57% of its optimal performance. Conversely, SLIM performs poorly without regularisation parameters, with performance ranging between 0.07% and 46.99% of the optimal performance in all the datasets. When only β is set ($\lambda = 0$), the performance of NSEM ranges between 87.66% and 100% of its optimal performance in all the datasets (at the different top-Ns), while the performance of SLIM only ranges between 40.4% and 97.5% of its optimal performance. When only λ is set ($\beta = 0$) the performance NSEM ranges between 95.78% and 100% of its optimal performance in all the datasets (for the different top-Ns), while the performance of SLIM only ranges between 65.5% and 96.09% of its optimal performance. This suggests that while SLIM depends heavily on both regularisation parameters to achieve optimal performance, this is not the case for NSEM, for which only λ could be used to enforce sparsity in the solution.

3.2 On the importance of the parameter γ

At a first glance, it may seem that the regularisation terms (in particular the $\mathcal{L}_2$ norm) could be enough to satisfy the null-diagonal constraint $\text{diag}(W) = 0 \equiv \text{Tr}(W) = 0$. This is due to the fact that when $\beta \gg 0$, all the values in W shrink down to zero. However, large β does not imply good recommendation performance. I illustrate this point on the Movielens dataset (Fig. 4.5a). The figure shows that although β reduces the trace, it also exerts a detrimental effect on the recommendation performance. There is in fact, a strong correlation between the recommendation performance and the $\text{Tr}(W)$ (Pearson correlation, $\rho > 0.95$,

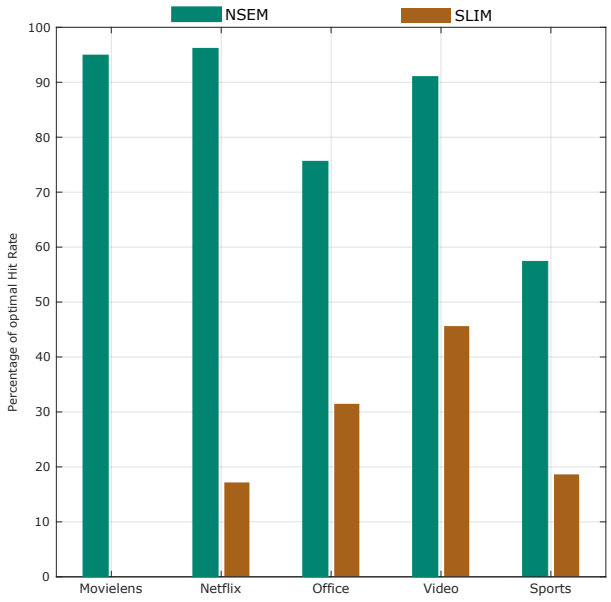

Figure 4.3: Percentage of the optimal $\overline{\text{HR}}$ at top 10 recommendations, achieved as parameter-free model ($\beta = \lambda = 0$). Percentages are relative to the optimal performance of each model.

Significance $p < 8.753 \times 10^{-9}$). This implies that the term $\gamma \, \text{Tr}(W)$ has an important role to guarantee the null diagonal constraint without trading recommendation performance. Figure 4.5b shows how the recommendation performance improves when γ increases and plateaus for a large enough γ. This is related to the fact that large γ guarantees a divergence from the trivial solution $W = I$ (Fig. 4.5c). In fact, the HR is moderately correlated to the distance from the trivial solution (Pearson correlation, $\rho > 0.598$, Significance $p < 0.015$). Importantly, these observations are supported by my theoretical guarantees given in Eqs. 4.9 and 4.10, that indicates a lower bound for γ but not an upper bound.

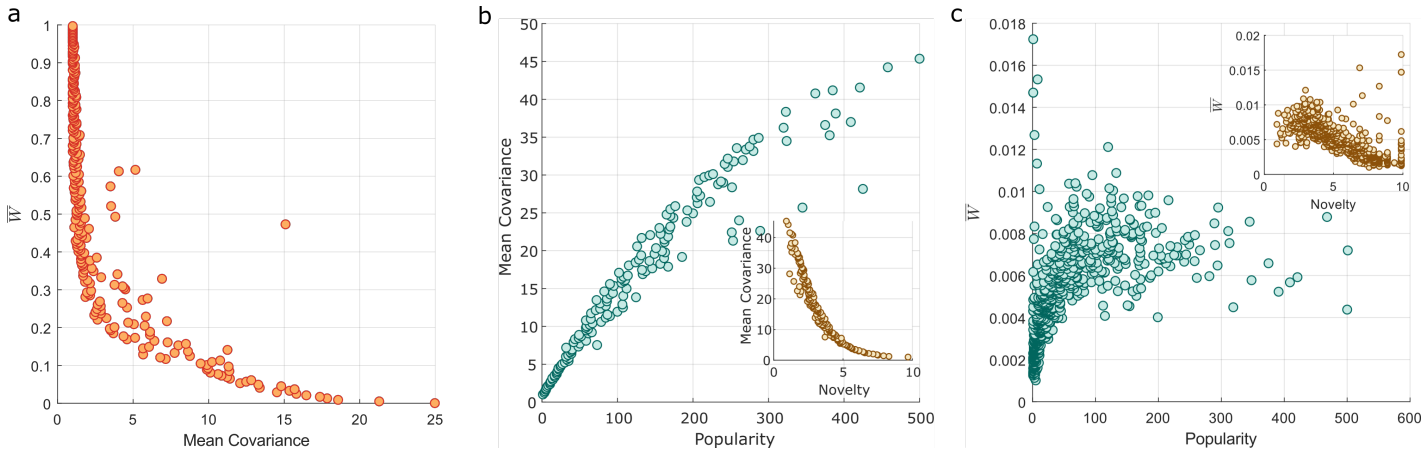

Figure 4.4: Effect of the covariance-driven regularisation in the Movielens dataset. (a) Learned weights in W as a function of the covariance values; (b) Mean covariance as a function of popularity. *Inset*. Novelty; (c) Mean learned weights in W as a function of popularity. *Inset*. Novelty.

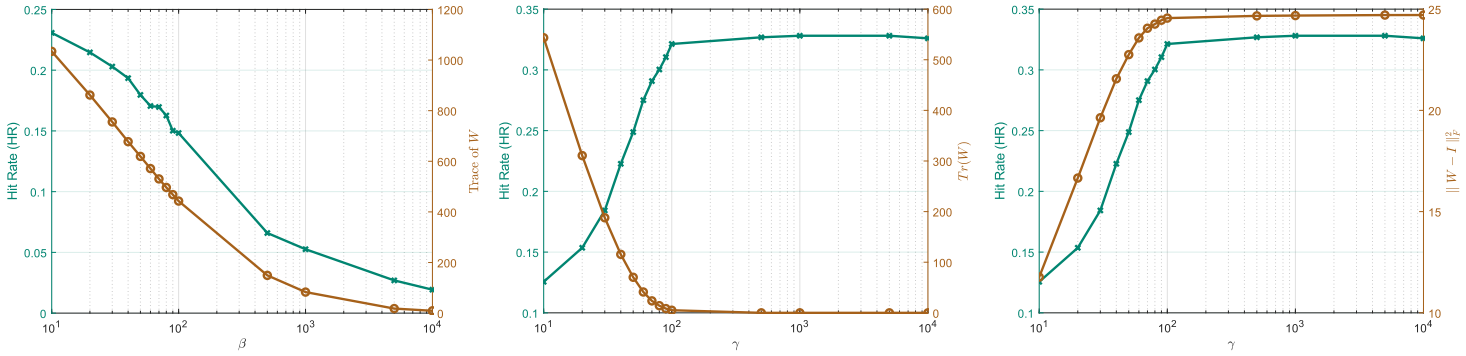

Figure 4.5: Model parameters effect on recommendation performance and null diagonal constraint (Movielens dataset) (a) Contrast between $\overline{\text{HR}}(N = 10)$ and $Tr(W)$ as a function of β. For this experiment, both $\lambda = \gamma = 0$. (b) Contrast between $\overline{\text{HR}}(N = 10)$ and $Tr(W)$ as a function of γ. For this experiment, both $\beta = \lambda = 0$. (c) Contrast between $\overline{\text{HR}}(N = 10)$ and distance from trivial solution (defined as $\| W - I \|_F^2$) as a function of γ. For this experiment, both $\beta = \lambda = 0$.

4 Model interpretability

One important feature of my model is that it is inherently interpretable[27]. This is because a given item, e.g. a movie, can be explained as a linear combination of few other items. This facilitates the explainability of how my model arrived to a given recommendation for an user. Furthermore, thanks to the theoretical guarantees of uniqueness and global minimum solution of my algorithm, the learned representations are reproducible under different random initialisation of the weights. I analysed the intrinsic algorithmic properties to understand both: (i) why my algorithm is less sensitive to the regularisation parameters and; (ii) how a particular regularisation in my algorithm drives the learning of W towards a solution that increases novelty in the recommendations.

4.1 Covariance-driven regularisation

To understand the reason why the NSEM is less sensitive to the model parameters, we need to look in detail at the multiplicative learning algorithm in Eq. 4.7. Observe that all the regularisation terms appear in the denominator of the algorithm. There is, however, an additional term in the denominator: $X^\top XW$. This term is de facto performing an additional regularisation for W, as it changes the amount by which W is updated at each iteration. The values of $X^\top XW$ rank significantly higher than zero at each iteration (Wilcoxon Signed Rank test with Bonberroni correction Significance, $p < 2.22 \times 10^{-306}$, in all the datasets). The same term also ranks significantly higher than the other regularisation terms ($\beta W + \lambda$) at each iteration (Wilcoxon Signed Rank Test with Bonberroni correction Significance, $p < 2.22 \times 10^{-306}$). To understand better the nature of this *covariance-driven regularisation*, let's consider the symmetric positive definite covariance

matrix $\mathcal{C} = X^\top X = [c_1, c_2, ..., c_m]$ given by its column vectors $c_j \, , \forall j \in \{1, 2, ..., m\}$ and the learned matrix $W = [w_1, w_2, ..., w_m]$ given by its column vectors $w_j \, , \forall j \in \{1, 2, ..., m\}$. The regularisation term $X^\top XW$ can be written as follows:

$$\mathcal{C}W = \begin{bmatrix} {c_1}^T w_1 & {c_1}^T w_2 & \dots & {c_1}^T w_m \\ {c_2}^T w_1 & {c_2}^T w_2 & \dots & {c_2}^T w_m \\ \vdots & \vdots & \vdots & \ddots \\ {c_m}^T w_1 & {c_m}^T w_2 & \dots & {c_m}^T w_m \end{bmatrix}, \tag{4.17}$$

which indicates that the amount of regularisation for a pair of items (i, j) is proportional to the dot product between the item i in the covariance (c_i) and an item j in W (w_j). This suggests that the values of W corresponding to items that co-occur with many other items (popular items) are penalised the most, whereas the values of W corresponding to items with low co-occurrence are allowed to grow. This may in principle seem counter intuitive. To understand this empirically, I analysed the regularisation term in the Movielens dataset by setting $\beta = \lambda = 0$. Fig. 4.4a shows that indeed the relationship between the covariance and W is non-linear: smaller values in the covariance correspond to higher weights in W (movies that are ranked first) and vice-versa, higher values in the covariance are related to smaller weights in W. In other words, W gives importance to relationships with possibly unpopular movies (from the long-tail) thanks to its regularisation.

4.2 W LEARNS NOVEL ITEM-ITEM RELATIONSHIPS

To further investigate the relationship between the learned weights in W and the movie popularity and novelty, I analysed the correlation of $\mathcal{C}$ and W with the popularity and nov-

elty metrics[‡]. The values in the covariance are positively correlated with the popularity - see Fig. 4.4b - (Pearson correlation $\rho = 0.96, p < 2.22 \times 10^{-308}$), and negatively correlated with the novelty (Pearson correlation $\rho = -0.83, p < 2.22 \times 10^{-308}$). While the weights in W are only moderately correlated to the popularity - see Fig. 4.4c - (Pearson correlation $\rho = 0.51, p < 5.42 \times 10^{-112}$), and less negatively correlated with novelty (Pearson correlation $\rho = -0.70, p < 8.89 \times 10^{-250}$). I observed similar trends across all the other datasets. For example, in the Netflix dataset, W is significantly correlated to the novelty (Pearson correlation $\rho = 0.41, p < 3.46 \times 10^{-160}$), whereas the covariance is negatively correlated to novelty (Pearson correlation $\rho = -0.88, p < 2.22 \times 10^{-308}$). These findings suggest that the reason behind the good performance without regularisation is due to the fact that NSEM multiplicative algorithm mitigates the bias of movie popularity and thus increases novelty.

5 High-rank vs Low-rank model

In the original SLIM publication[43], it is shown that SLIM outperforms several low-rank matrix decomposition techniques. The reason, however, has remained unclear. Low-rank matrix decomposition typically assigns a low-dimensional feature vector to each user and a low-dimensional feature vector to each item so that the user-item preference is modelled by the dot-product of the two feature vectors. This means that the $n \times m$ matrix X is the product of an $n \times k$ matrix W whose rows are the user feature vectors and a $k \times m$ matrix H whose columns are the items feature vectors. The rank of X is k — the number of features assigned to each user or item. I compared the performance of NSEM and SLIM

[‡]Popularity of a movie j was defined as the number of user that rated movie j[20]. Novelty was defined as $-\log_2(\text{popularity}/\#\text{users})$[177].

Table 4.4: High/Full rank structure of datasets

Dataset (X)	density	rank of X	Comment
Movielens	6.3%	943	full rank matrix
Netflix	1.66%	3995	full rank matrix
Office	0.45%	2420	full rank matrix
Video	0.38%	1685	full rank matrix
Sports	0.045%	18387	full rank matrix

The column corresponding to rank was calculated using the built-in Matlab R2018a function `rank(X)` using all the available data in each dataset. The rank of a matrix X is computed as the number of singular values that are larger than a predefined tolerance.

versus two popular matrix decomposition techniques: PureSVD[20] and NMF[117] across all the datasets. Figure 4.6 shows the performance of the methods in the top 10 recommendations. We can observe that while for the Movielens dataset the performance of low-rank vs high-rank models is only comparable, for Netflix, Amazon Office, Amazon Video and Amazon Sports, high-rank models are significantly better. The difference in performance between low-rank vs high-rank models seems to be related to the density of X (see Table 4.4). The sparser the dataset the more advantage there is in using a high-rank model rather than a low-rank model. This is possibly due to the fact that high-rank models capture high-level features in the data by exploiting weaker regularities (small singular values) which is normally ignored by low-rank models.

6 Conclusions and Discussion

I have introduced a novel algorithmic framework for high-rank matrix completion under self-expressive models. I have provided strong theoretical foundations regarding the objective function and the optimality of the solution using my multiplicative learning algorithm.

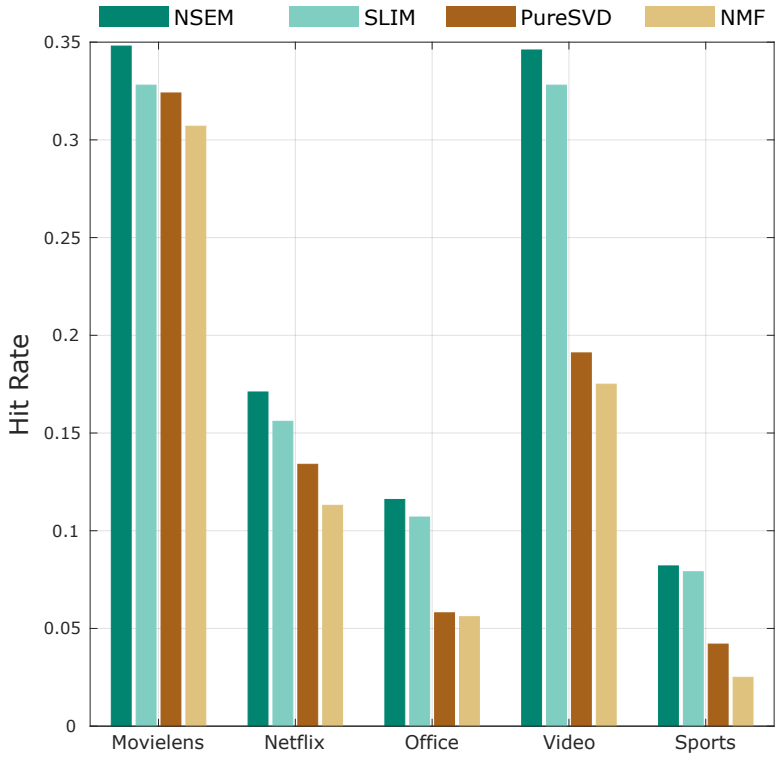

Figure 4.6: Performance of high-rank versus low-rank models in terms of $\overline{\text{HR}}$ in the top 10 recommendations.

I show that my objective function is smooth and that my learning algorithm converges to a unique global optimum solution. I also show that my NSEM algorithm, that improves upon the Sparse Linear Method (SLIM) formulation, performs better than SLIM across several real-world datasets. Finally, I have analysed the inner working of my algorithm when learning the non-negative matrix of coefficients W. I found that a covariance-driven regularisation was responsible for the robustness in performance of my algorithm and that it accounts also for learning a W that favours novelty in the recommendations.

Algorithmically, my NSEM algorithm is closely related to non-negative matrix factorisation (NMF)[29,117]. Lee and Seung[117] have shown that, by diagonally rescaling the learning rate at each iteration the resulting matrices are non-negative and that the cost function con-

verges monotonically. My algorithm is inspired by the principles of non-negative matrix factorisation. As it happens with NMF, my formulation does not requires setting a stepsize parameter nor applying a projection function to guarantee non-negativity at each iteration.

One of the limitations of NSEM (and SLIM) is that it can be slower in large-scale application that other low-rank matrix decomposition techniques. I have not addressed this problem in this chapter. However, to reduce time complexity, Ning and Karypis[43] proposed to applied feature selection when learning W. This procedure can possibly be also applied in my algorithm (using Eq. 4.13). I have shown that NSEM provides additional advantages over SLIM. For instance, when analysing the sensitivity of my algorithm under changes of the hyperparameters, I found that there is a particular type of regularisation —that I called covariance-driven regularisation — responsible for the robustness in performance observed in the experiments. This feature of my algorithm reduces the need for parameter tuning in online applications, where re-training a model under a large grid of hyperparameters can be prohibited. Conversely, I observed that SLIM requires a fine-tuning of its parameters to achieve optimal recommendation performance.

One of the main advantages of my formulation is the ease of adding additional constraints in the objective function. To preserve convexity, it is important that any new constraint on the objective function is convex term (positive semi-definite matrix). This is the case, for instance, of our models shown in chapters 2 and 3.

The main difference between my algorithm and previous high-rank matrix completion models[8] is based on the model assumption. Standard high-rank matrix completion typically assume that X is *partially observed*, and thus the learning is performed on the well-defined set of observed entries only. This is in fact, the most common assumption in matrix

completion[92,178]. However, since Cremonesi et. al.[20] showed that significant performance improvements can be obtained by simply representing missing user-item entries in X with zero values, this has become the de facto data representation for models in top-N recommendation systems. This means that both NSEM and SLIM operates on the assumption that X is *fully observed* while its entries are *noisy*[§].

There is a growing concern in the literature about model reproducibility[179]. Dacrema et. al.[180] have recently shown that less than half of deep learning models presented in top conferences could be reproduced. Of these, only one deep learning model (Mult-VAE[181]) performed better than simple heuristic methods such as those based on nearest-neighbours or graph-based models. However, even Mult-VAE did not consistently outperform SLIM in the different scenarios. Therefore, SLIM and therefore NSEM, are competitive state-of-the-art methods for top-N recommender system.

[§]The use of the Frobenius norm in the term $||X - XW||_F^2$ (in Eq. 4.4) implies that the noise is bounded.

To myself I am only a child playing on the beach, while vast oceans of truth lie undiscovered before me.

Sir Isaac Newton (1643-1727)

5

Conclusions

In this dissertation, I have addressed several important problems across distinct domains, ranging from healthcare to recommender systems. The algorithms that I presented here were mainly motivated by characteristics and limitations observed in the data. I have developed novel low-rank and high-rank models and optimisation algorithms

for completing an incomplete matrix. Throughout the chapters of this dissertation, the primary connection between the models was interpretability. I believe interpretability is one of the bridges that is needed to connect the current machine learning research to the clinical practice. Ideally, this will help scientists, doctors and policymakers to make informed decisions. I found that one way of favouring interpretability in my matrix completion models was by learning *non-negative sparse representations*. Sparsity is often a useful measure of interpretability, because humans can handle at most 7 ± 2 cognitive entities at once[182].

I will conclude by briefly summarising the key contributions once more, followed by future (some ongoing) research directions.

1 Summary of contributions

- In chapter 1, I have addressed the problem of predicting the frequencies of drug side effects. I proposed a novel non-negative matrix decomposition model that accounts for different levels of uncertainty in the data. To my knowledge, this is the first attempt to predict the frequencies of drug side effects. Importantly, I show that the signatures (or learned low-dimensional representations) of drugs and side effects are pharmacologically interpretable: they encode specific chemical perturbations at human anatomical and organ-system level.

  My novel machine learning approach can be used in the early phase, small-size clinical trials to set the direction of the risk assessment in later clinical trials or after a drug has entered the market. It can also be used in other aspects of clinical trial design, such as in the estimation of the cohort size needed for the trials. I show that the signatures of my model can be exploited to formulate a specific biological hypothesis

on the drug mechanism of action at both molecular and organ-system levels. For instance, a specific component in the signature in my model is significantly associated with arrhythmias, and therefore, can be useful for cardiotoxicity assessment.

- In chapter 2, I have addressed the problem of predicting the presence or absence of drug side effects. I proposed a low-rank model and a novel geometric high-rank model based on self-representation. Through extensive experiments, I show that my model consistently outperforms the baselines. My self-representation model also favours model interpretability: two drugs with similar self-representations tend to be clinically related or they share common protein targets.

- In chapter 3, I have addressed the problem of disease gene prediction. The disease-gene prediction problem it was a natural target of my high-rank model given that it shares many characteristics of other datasets such as sparsity and skewed distribution of diseases, in which my model have shown to work well. An important contribution of this chapter is the fact that my model can predict genes for molecularly uncharacterised diseases.

- In chapter 4, I have addressed the problem of top-N recommendation systems. Most of the theoretical development of my high-rank model is explained in this chapter. I showed that my high-rank model is motivated by the development of self-expressive models and sparse linear method. I proposed novel algorithms that can be used to solve SLIM-based objective functions. I also show that the learned self-representations can be interpreted in terms of popularity and novelty metrics.

2 Future directions

There are several research avenues for future work regarding the problems that I addressed in this dissertation, as well as follow-up ideas on the algorithmic developments presented. I summarised some of these ideas here:

- *Predicting the frequencies of side effects for novel compounds.* It would be interesting to extend the work presented here to predict the side effect of drugs by relying solely on molecular or cellular features, for instance, by exploiting the similarities in chemical structure or the activity across cell lines. A similar problem has been already addressed in the literature but to predict the presence/absence of drug side effects[111].

- *Predicting polypharmacy drug side effects using the signatures.* Recent work[183,184] have addressed the problem of predicting specific side effects that two drugs cause when taken in combination. However, these black-box approaches are far from providing any biological interpretability. It would be interesting to analyse whether my drug signatures can be used to predict polypharmacy side effects while enhancing model interpretability. Ruben Jimenez has started this work and so far, preliminary results are encouraging.

- *Drug repositioning.* Another important research avenue is drug repositioning, that is, finding new indications for old drugs[185,186]. It would be interesting to explore the power of the drug signatures to predict new drug indications. In Chapter 1, I have shown that drug signatures encode the main drug therapeutic indication. This result already indicates the potential of using drug signatures to predict new drug indications. Another idea is to apply my geometric high-rank model to drug reposition-

ing: work of Fabrizio Frasca[62], performed under my direct supervision. Our preliminary results will be presented at the Graph Representation Learning Workshop at NeurIPS 2019*.

- *Learning sparse self-representations by neural networks.* In this dissertation, I focused on developing inherently interpretable models. These are unfeasible to achieve with neural nets because the learned representations depend on uninterpreted features in other layers[28]. However, interpretability, which is always domain-specific notion[27], is only required in certain situations, for instance, when the decision suggested by a model can influence human-level decisions. It would be interesting to develop neural network models capable of learning self-representations by also considering structured data such as graph networks. There has been a recent work on low-rank matrix completion using neural networks[26,187], but we still lack high-rank matrix completion models using neural networks. Interestingly, self-representations learned by neural networks might offer the first opportunity to built more interpretable neural networks.

*https://grlearning.github.io/papers/

A

Appendix

1 Collection of the side effect frequencies

We started with 1,556 marketed drugs listed in the Side effect Resource Database (SIDER) 4.1[65] that contains 4,251 side effect terms mapped in the Medical Dictionary for Regulatory Activities (MedDRA) v20.0. In SIDER 4.1, around 40% of the drug-side effect pairs contains frequency information, whereas for the remaining 60% the frequency is unknown.

Side effect terms in SIDER were annotated with their MedDRA lowest level term (LLT) and preferred term (PT). Many LLTs may correspond to the same PT. For instance, the MedDRA LLTs Creatinine increased (C0151578), Blood creatinine increased (C0235431), Serum creatinine increased (C0700225) and Plasma creatinine increased (C0858118) corresponds to the same MedDRA PT Blood creatinine increased (C0235431). Therefore, we used PT side effect terms to collect the pair-input associations between drugs and side effects. Side effect frequencies were listed in SIDER as exact frequencies, e.g. 1%, range of frequencies, e.g. 2-5% or as frequency class e.g. very rare, rare, infrequent, frequent and very frequent. Placebo frequencies were sometimes provided as exact frequency value or range of values. Fig. A.1 summarizes the types of frequency formats in a Venn diagram depicting the three different sets of data format found. For a given input pair, multiple frequencies might also be available, for example, from clinical trials for different indications[87]. For convenience, we standardise all the frequencies into frequency classes. Exact frequencies and range of frequencies were mapped to frequency classes. The mapping between the range of side effect occurrences and the classes are regulated by the Council for International Organizations of Medical Sciences (CIOMS). For the different subsets depicted in Figure A.1, we pre-processed the frequency data as follow;

- Subset $A - (A \cup B)$. When only the exact or range frequency was available, we compute the median frequency and then map to a frequency class.
- Subset $B - (A \cup C)$. When only the frequency class label was available, we kept the labels but normalised the following terms: very rare, rare, infrequent (or uncommon), frequent (or common) and very frequent (or very common).

- Subset $C - (A \cup B)$. We have found 215 pairs for which the placebo frequency was found, but the drug frequency was not listed in SIDER 4.1. We manually checked several pairs to confirm that the drug frequency was indeed missing in the database. We discarded these pairs.

- Subset $A \cap C$. We retained pairs for which the median frequency in the intervention cohort was higher than the median frequency in the placebo cohort. For 2,474 pairs, the median frequency in both groups was comparable and for 403 out of the 2,474 pairs, the placebo frequency was higher. These associations are likely to be caused by the disease or by the so-called nocebo effect (14), i.e. patients that anticipate a side effect on medication are more likely to report it (2). We discarded these pairs to avoid possible confounders in the associations.

- Subset $A \cap B$. We compute the median frequency value and map to a frequency class. We also kept the frequency class from set B.

- Subset $B \cap C$. We discarded the placebo frequencies for which not intervention frequency was found. We kept the frequency class from set B.

- Subset $A \cap B \cap C$. We retained pairs for which the median frequency was higher in the intervention cohort than the placebo cohort. We also kept the frequency class from set B.

After mapping all the values to frequency class values, for around 13% of the associations, we could have more than one rating value for a given drug side effect due to the multiple intersections in the data. 8% of these frequency classes were inconsistent (different). These

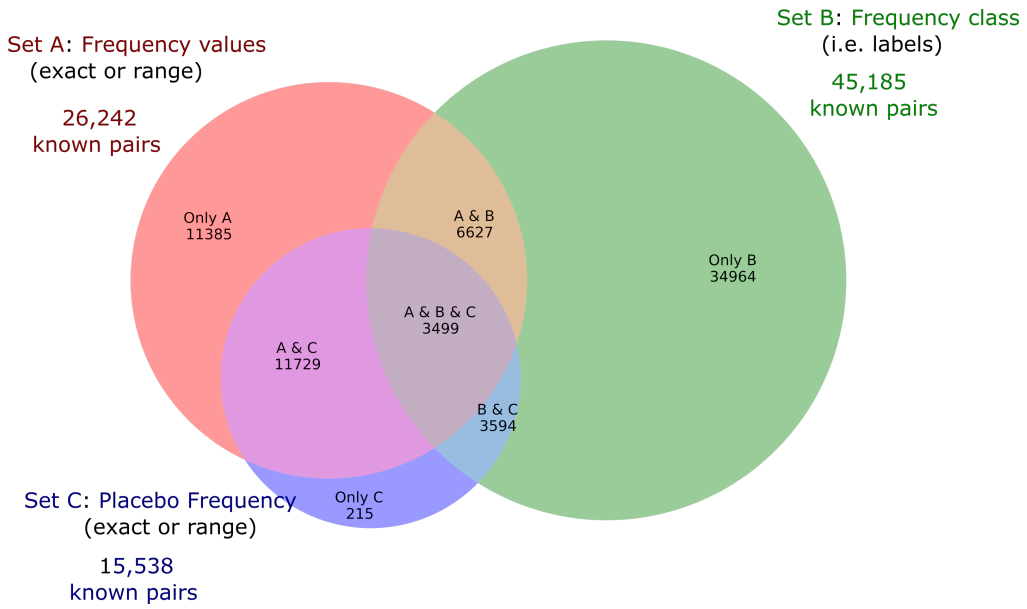

Figure A.1: Venn diagram depicting the different formats for the drug side effect frequencies in SIDER 4.1. In total, 68,514 pairs were found with frequency information. There are three overlapping sets of data formats. Set A: contains drug exact (e.g. 1%) and range frequency (e.g. 2-5%); set B contains frequency classes (e.g. very rare), and set C contains the exact and range placebo frequencies. The size of the circles is proportional to the number of drug-side effect pairs in each set.

might be due to clinical trials from different indications for the same drug[87]. For these cases, we average the rating values and then round them to the nearest highest integer. Until here, we have extracted 41,546 frequency class associations for 860 marketed drugs with around 1,011 unique side effect terms. Furthermore, we kept only drugs with known monotherapy Anatomical Therapeutic and Chemical (ATC) category according to the 2018 World Health Organization (WHO) release. Similarly, we kept only side effects with known MedDRA category of disorders. In total, our final dataset contains 759 marketed drugs with 994 side effect terms with 37,441 known rating values.

2 Additional datasets used in our study

Post-marketing side effects For our dataset of 759 drugs with 994 side-effect terms, we retrieved the binary drug-side effect associations with unknown frequencies from SIDER 4.1. We found 55,382 binary associations in SIDER. These binary associations are either from clinical trials with unreported frequency or from post-marketing reports added to drug leaflets (2). We collected 9,387 pairs with an explicit post-marketing label in SIDER. This set constitutes our post-marketing test set.

Protein targets We retrieved the known drug-target interactions from DrugBank release 5.0.5 (2016-08-17) (3). We mapped the drugs from SIDER to DrugBank using the PubChem IDs and the mapping provided in DrugBank. We retrieved molecular targets (with known or unknown pharmacological action) for 435 drugs in our dataset. In total, 1,759 associations were found between the 435 drugs and 590 unique protein targets.

Chemical fingerprints We retrieved the known drug SMILES fingerprint from DrugBank release 5.0.5 (2016-08-17) (3). 442 drugs in our dataset were mapped. We then computed the 2D Tanimoto chemical similarity based on the fingerprint using the Open-Source Cheminformatics (RDKit) (15) in python.

ATC codes and route of administration Drugs ATC codes and drugs route of administration (Adm.R) were obtained from the ATC codes WHO 2018 release. The routes of administration of the drugs can be implant, inhalation, instillation, nasal (N), oral (O), parenteral (P), rectal (R), sublingual/buccal/oromucosal (SL), transdermal (TD) and vaginal (V).

References

[1] Yungki Park and Edward M Marcotte. Flaws in evaluation schemes for pair-input computational predictions. *Nature methods*, 9(12):1134, 2012.

[2] Juwen Shen, Jian Zhang, Xiaomin Luo, Weiliang Zhu, Kunqian Yu, Kaixian Chen, Yixue Li, and Hualiang Jiang. Predicting protein–protein interactions based only on sequences information. *Proceedings of the National Academy of Sciences*, 104(11):4337–4341, 2007.

[3] Jarl ES Wikberg and Felikss Mutulis. Targeting melanocortin receptors: an approach to treat weight disorders and sexual dysfunction. *Nature Reviews Drug Discovery*, 7(4):307, 2008.

[4] Hiroaki Yabuuchi, Satoshi Niijima, Hiromu Takematsu, Tomomi Ida, Takatsugu Hirokawa, Takafumi Hara, Teppei Ogawa, Yohsuke Minowa, Gozoh Tsujimoto, and Yasushi Okuno. Analysis of multiple compound–protein interactions reveals novel bioactive molecules. *Molecular systems biology*, 7(1):472, 2011.

[5] Matteo Bellucci, Federico Agostini, Marianela Masin, and Gian Gaetano Tartaglia. Predicting protein associations with long noncoding rnas. *Nature methods*, 8(6):444, 2011.

[6] 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.

[7] MR Hurle, L Yang, Q Xie, DK Rajpal, P Sanseau, and P Agarwal. Computational drug repositioning: from data to therapeutics. *Clinical Pharmacology & Therapeutics*, 93(4):335–341, 2013.

[8] Ehsan Elhamifar. High-rank matrix completion and clustering under self-expressive models. In *Advances in Neural Information Processing Systems*, pages 73–81, 2016.

[9] Yugang Wang and Ehsan Elhamifar. High rank matrix completion with side information. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[10] Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, and Inderjit Dhillon. Large-scale multi-label learning with missing labels. In *International conference on machine learning*, pages 593–601, 2014.

[11] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, 11(Aug):2287–2322, 2010.

[12] Emmanuel J Candes and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[13] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE transactions on information theory*, 56(6):2980–2998, 2010.

[14] Yudong Chen, Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust matrix completion and corrupted columns. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 873–880, 2011.

[15] Srinadh Bhojanapalli and Prateek Jain. Universal matrix completion. In *Proceedings of the 31st International Conference on International Conference on Machine Learning-Volume 32*, pages II–1881. JMLR. org, 2014.

[16] James Bennett, Stan Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA., 2007.

[17] Anton F Fliri, William T Loging, Peter F Thadeio, and Robert A Volkmann. Analysis of drug-induced effect patterns to link structure and side effects of medicines. *Nature chemical biology*, 1(7):389, 2005.

[18] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[19] Albert-László Barabási, Natali Gulbahce, and Joseph Loscalzo. Network medicine: a network-based approach to human disease. *Nature reviews genetics*, 12(1):56, 2011.

[20] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 39–46. ACM, 2010.

[21] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[22] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM, 2002.

[23] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[24] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv:1709.05584*, 2017.

[25] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[26] Federico Monti, Michael Bronstein, and Xavier Bresson. Geometric matrix completion with recurrent multi-graph neural networks. In *Advances in Neural Information Processing Systems*, pages 3697–3707, 2017.

[27] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206, 2019.

[28] Geoffrey Hinton. Deep learning—a technology with the potential to transform health care. *Jama*, 320(11):1101–1102, 2018.

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

[30] Diego Galeano and Alberto Paccanaro. Predicting the frequency of drug side effects. *bioRxiv*, 2019.

[31] Daniel M Bean, Honghan Wu, Ehtesham Iqbal, Olubanke Dzahini, Zina M Ibrahim, Matthew Broadbent, Robert Stewart, and Richard JB Dobson. Knowledge graph prediction of unknown adverse drug reactions and validation in electronic health records. *Scientific reports*, 7(1):16416, 2017.

[32] Nir Atias and Roded Sharan. An algorithmic framework for predicting side effects of drugs. *Journal of Computational Biology*, 18(3):207–218, 2011.

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

[34] Wen Zhang, Xinrui Liu, Yanlin Chen, Wenjian Wu, Wei Wang, and Xiaohong Li. Feature-derived graph regularized matrix factorization for predicting drug side effects. *Neurocomputing*, 287:154–162, 2018.

[35] Rong Li, Yongcheng Dong, Qifan Kuang, Yiming Wu, Yizhou Li, Min Zhu, and Menglong Li. Inductive matrix completion for predicting adverse drug reactions (adrs) integrating drug–target interactions. *Chemometrics and Intelligent Laboratory Systems*, 144:71–79, 2015.

[36] Diego Galeano and Alberto Paccanaro. A recommender system approach for predicting drug side effects. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[37] Diego Ariel Galeano Galeano and Alberto Paccanaro. The geometric sparse matrix completion model for predicting drug side effects. *bioRxiv*, page 652412, 2019.

[38] Susan Dina Ghiassian, Jörg Menche, and Albert-László Barabási. A disease module detection (diamond) algorithm derived from a systematic analysis of connectivity patterns of disease proteins in the human interactome. *PLoS computational biology*, 11(4):e1004120, 2015.

[39] Fantine Mordelet and Jean-Philippe Vert. Prodige: Prioritization of disease genes with multitask machine learning from positive and unlabeled examples. *BMC bioinformatics*, 12(1):389, 2011.

[40] Oron Vanunu, Oded Magger, Eytan Ruppin, Tomer Shlomi, and Roded Sharan. Associating genes and protein complexes with disease via network propagation. *PLoS computational biology*, 6(1):e1000641, 2010.

[41] Juan J Cáceres and Alberto Paccanaro. Disease gene prediction for molecularly uncharacterized diseases. *PLoS computational biology*, 15(7):e1007078, 2019.

[42] Charu C Aggarwal et al. *Recommender systems*. Springer, 2016.

[43] Xia Ning and George Karypis. Slim: Sparse linear methods for top-n recommender systems. *2011 11th IEEE International Conference on Data Mining*, pages 497–506, 2011.

[44] David W Bates, David J Cullen, Nan Laird, Laura A Petersen, Stephen D Small, Deborah Servi, Glenn Laffel, Bobbie J Sweitzer, Brian F Shea, Robert Hallisey, et al. Incidence of adverse drug events and potential adverse drug events: implications for prevention. *Jama*, 274(1):29–34, 1995.

[45] David C Classen, Stanley L Pestotnik, R Scott Evans, James F Lloyd, and John P Burke. Adverse drug events in hospitalized patients: excess length of stay, extra costs, and attributable mortality. *Jama*, 277(4):301–306, 1997.

[46] John C Dearden. In silico prediction of drug toxicity. *Journal of computer-aided molecular design*, 17(2-4):119–127, 2003.

[47] Sean Ekins, Jordi Mestres, and Bernard Testa. In silico pharmacology for drug discovery: methods for virtual ligand screening and profiling. *British journal of pharmacology*, 152(1):9–20, 2007.

[48] Jacques Hamon, Steven Whitebread, Valerie Techer-Etienne, Helene Le Coq, Kamal Azzaoui, and Laszlo Urban. In vitro safety pharmacology profiling: what else beyond herg? *Future medicinal chemistry*, 1(4):645–665, 2009.

[49] Mark R Fielden and Kyle L Kolaja. The role of early in vivo toxicity testing in drug discovery toxicology. *Expert opinion on drug safety*, 7(2):107–110, 2008.

[50] Jeffrey A Kramer, John E Sagartz, and Dale L Morris. The application of discovery toxicology and pathology towards the design of safer pharmaceutical lead candidates. *Nature reviews drug discovery*, 6(8):636, 2007.

[51] Cindy J Gross and Jeffrey A Kramer. The role of investigative molecular toxicology in early stage drug development. *Expert opinion on drug safety*, 2(2):147–159, 2003.

[52] Annie P Chiang and Atul J Butte. Data-driven methods to discover molecular determinants of serious adverse drug events. *Clinical Pharmacology & Therapeutics*, 85(3):259–268, 2009.

[53] Jennie L Walgren and David C Thompson. Application of proteomic technologies in the drug development process. *Toxicology letters*, 149(1-3):377–385, 2004.

[54] Jeremy K Nicholson, John Connelly, John C Lindon, and Elaine Holmes. Metabonomics: a platform for studying drug toxicity and gene function. *Nature reviews Drug discovery*, 1(2):153, 2002.

[55] Wilbert HM Heijne, Anne S Kienhuis, Ben Van Ommen, Rob H Stierum, and John P Groten. Systems toxicology: applications of toxicogenomics, transcriptomics, proteomics and metabolomics in toxicology. *Expert review of proteomics*, 2(5):767–780, 2005.

[56] Eugene C Butcher. Can cell systems biology rescue drug discovery? *Nature Reviews Drug Discovery*, 4(6):461, 2005.

[57] June S Almenoff, Karol K LaCroix, Nancy A Yuen, David Fram, and William DuMouchel. Comparative performance of two quantitative safety signalling methods. *Drug safety*, 29(10):875–887, 2006.

[58] Jeffrey S Brown, Martin Kulldorff, K Arnold Chan, Robert L Davis, David Graham, Parker T Pettus, Susan E Andrade, Marsha A Raebel, Lisa Herrinton, Douglas Roblin, et al. Early detection of adverse drug events within population-based health networks: application of sequential testing methods. *Pharmacoepidemiology and drug safety*, 16(12):1275–1284, 2007.

[59] G Niklas Norén and I Ralph Edwards. Modern methods of pharmacovigilance: detecting adverse effects of drugs. *Clinical Medicine*, 9(5):486–489, 2009.

[60] Kathleen M Giacomini, Ronald M Krauss, Dan M Roden, Michel Eichelbaum, Michael R Hayden, and Yusuke Nakamura. When good drugs go bad. *Nature*, 446(7139):975, 2007.

[61] Timothy Brewer and Graham A Colditz. Postmarketing surveillance and adverse drug reactions: current perspectives and future needs. *Jama*, 281(9):824–829, 1999.

[62] Diego Galeano and Alberto Paccanaro. Predicting the frequency of drug side effects. *bioRxiv*, page 594465, 2019.

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

[64] Mary Regina Boland, Alexandra Jacunski, Tal Lorberbaum, Joseph D Romano, Robert Moskovitch, and Nicholas P Tatonetti. Systems biology approaches for identifying adverse drug reactions and elucidating their underlying biological mechanisms. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 8(2):104–122, 2016.

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

[66] Fabrizio Frasca, Diego Galeano, Guadalupe Gonzalez, Ivan Laponogov, Kirill Veselvoc, Alberto Paccanaro, and Michael Bronstein. Learning interpretable disease self-representations for drug repositioning. *arXiv preprint arXiv:1909.06609*, 2019.

[67] Stein Aerts, Diether Lambrechts, Sunit Maity, Peter Van Loo, Bert Coessens, Frederik De Smet, Leon-Charles Tranchevent, Bart De Moor, Peter Marynen, Bassem Hassan, et al. Gene prioritization through genomic data fusion. *Nature biotechnology*, 24(5):537, 2006.

[68] Committee on Ethical, Scientific Issues in Studying the Safety of Approved Drugs, Board on Population Health, Public Health Practice, and Institute of Medicine. *Ethical and scientific issues in studying the safety of approved drugs*. National Academies Press, 2012.

[69] Jeff J Guo, Swapnil Pandey, John Doyle, Boyang Bian, Yvonne Lis, and Dennis W Raisch. A review of quantitative risk–benefit methodologies for assessing drug safety and efficacy—report of the ispor risk–benefit management working group. *Value in Health*, 13(5):657–666, 2010.

[70] ZE Winters, C Griffin, R Horne, N Bidad, and P McCulloch. Barriers to accrue to clinical trials and possible solutions, 2014.

[71] Colin Begg, Mildred Cho, Susan Eastwood, Richard Horton, David Moher, Ingram Olkin, Roy Pitkin, Drummond Rennie, Kenneth F Schulz, David Simel, et al. Improving the quality of reporting of randomized controlled trials: the consort statement. *Jama*, 276(8):637–639, 1996.

[72] John Concato, Nirav Shah, and Ralph I Horwitz. Randomized, controlled trials, observational studies, and the hierarchy of research designs. *New England journal of medicine*, 342(25):1887–1892, 2000.

[73] Jacoline C Bouvy, Marie L De Bruin, and Marc A Koopmanschap. Epidemiology of adverse drug reactions in europe: a review of recent observational studies. *Drug safety*, 38(5):437–453, 2015.

[74] 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.

[75] Evelyn M Rodriguez, Judy A Staffa, and David J Graham. The role of databases in drug postmarketing surveillance. *Pharmacoepidemiology and drug safety*, 10(5):407–410, 2001.

[76] Munir Pirmohamed, Sally James, Shaun Meakin, Chris Green, Andrew K Scott, Thomas J Walley, Keith Farrar, B Kevin Park, and Alasdair M Breckenridge. Adverse drug reactions as cause of admission to hospital: prospective analysis of 18 820 patients. *Bmj*, 329(7456):15–19, 2004.

[77] Jason Lazarou, Bruce H Pomeranz, and Paul N Corey. Incidence of adverse drug reactions in hospitalized patients: a meta-analysis of prospective studies. *Jama*, 279(15):1200–1205, 1998.

[78] Yoshihiro Yamanishi, Michihiro Araki, Alex Gutteridge, Wataru Honda, and Minoru Kanehisa. Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, 24(13):i232–i240, 2008.

[79] Emmanuel Bresso, Renaud Grisoni, Gino Marchetti, Arnaud Sinan Karaboga, Michel Souchet, Marie-Dominique Devignes, and Malika Smaïl-Tabbone. Integrative relational machine-learning for understanding drug side-effect profiles. *BMC bioinformatics*, 14(1):207, 2013.

[80] 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.

[81] Alasdair Breckenridge, Kent Woods, and June Raine. Monitoring the safety of licensed medicines. *Nature Reviews Drug Discovery*, 4(7):541, 2005.

[82] Igho J Onakpoya, Carl J Heneghan, and Jeffrey K Aronson. Worldwide withdrawal of medicinal products because of adverse drug reactions: a systematic review and analysis. *Critical reviews in toxicology*, 46(6):477–489, 2016.

[83] Linda Martin, Melissa Hutchens, Conrad Hawkins, and Alaina Radnov. How much do clinical trials cost?, 2017.

[84] Michael D Ekstrand, John T Riedl, Joseph A Konstan, et al. Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction*, 4(2):81–173, 2011.

[85] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.

[86] Shimon Ullman et al. *High-level vision: Object recognition and visual cognition*, volume 2. MIT press Cambridge, MA, 1996.

[87] 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.

[88] Monica Campillos, Michael Kuhn, Anne-Claude Gavin, Lars Juhl Jensen, and Peer Bork. Drug target identification using side-effect similarity. *Science*, 321(5886):263–266, 2008.

[89] Tao Li and Chris Ding. The relationships among various nonnegative matrix factorization methods for clustering. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 362–371. IEEE, 2006.

[90] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[91] Michael W Berry, Murray Browne, Amy N Langville, V Paul Pauca, and Robert J Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational statistics & data analysis*, 52(1):155–173, 2007.

[92] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

[93] Seungjin Choi. Algorithms for orthogonal nonnegative matrix factorization. In *2008 ieee international joint conference on neural networks (ieee world congress on computational intelligence)*, pages 1828–1832. IEEE, 2008.

[94] Per Christian Hansen. Truncated singular value decomposition solutions to discrete ill-posed problems with ill-determined numerical rank. *SIAM Journal on Scientific and Statistical Computing*, 11(3):503–518, 1990.

[95] Andrea Fuentes, Moises Pineda, and Kalyan Venkata. Comprehension of top 200 prescribed drugs in the us as a resource for pharmacy teaching, training and practice. *Pharmacy*, 6(2):43, 2018.

[96] Elijah R Behr and Dan Roden. Drug-induced arrhythmia: pharmacogenomic prescribing? *European heart journal*, 34(2):89–95, 2012.

[97] 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.

[98] Ludmil B Alexandrov, Serena Nik-Zainal, David C Wedge, Samuel AJR Aparicio, Sam Behjati, Andrew V Biankin, Graham R Bignell, Niccolo Bolli, Ake Borg, Anne-Lise Børresen-Dale, et al. Signatures of mutational processes in human cancer. *Nature*, 500(7463):415, 2013.

[99] Ludmil B Alexandrov, Serena Nik-Zainal, David C Wedge, Peter J Campbell, and Michael R Stratton. Deciphering signatures of mutational processes operative in human cancer. *Cell reports*, 3(1):246–259, 2013.

[100] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[101] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[102] Emre Guney, Jörg Menche, Marc Vidal, and Albert-László Barábasi. Network-based in silico drug efficacy screening. *Nature communications*, 7:10331, 2016.

[103] Todd C Knepper and Howard L McLeod. When will clinical trials finally reflect diversity?, 2018.

[104] A Ramamoorthy, MA Pacanowski, J Bull, and L Zhang. Racial/ethnic differences in drug disposition and response: review of recently approved drugs. *Clinical Pharmacology & Therapeutics*, 97(3):263–273, 2015.

[105] James L Little, Antony J Williams, Alexey Pshenichnov, and Valery Tkachenko. Identification of "known unknowns" utilizing accurate mass data and chemspider. *Journal of the American Society for Mass Spectrometry*, 23(1):179–185, 2012.

[106] Donna M Lisi. Lotronex withdrawal. *Archives of internal medicine*, 162(1):101–101, 2002.

[107] Nazzareno Galie, Marius M Hoeper, J Simon R Gibbs, and Gerald Simonneau. Liver toxicity of sitaxentan in pulmonary arterial hypertension. *European Respiratory Journal*, 37(2):475–476, 2011.

[108] Diego Galeano and Alberto Paccanaro. Drug targets prediction using chemical similarity. In *2016 XLII Latin American Computing Conference (CLEI)*, pages 1–7. IEEE, 2016.

[109] Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328, 2004.

[110] David R Hardoon and John Shawe-Taylor. Sparse canonical correlation analysis. *Machine Learning*, 83(3):331–353, 2011.

[111] 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.

[112] Lenore Cowen, Trey Ideker, Benjamin J Raphael, and Roded Sharan. Network propagation: a universal amplifier of genetic associations. *Nature Reviews Genetics*, 18(9):551, 2017.

[113] Jicong Fan and Tommy WS Chow. Matrix completion by least-square, low-rank, and sparse self-representations. *Pattern Recognition*, 71:290–305, 2017.

[114] Andrew Y Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78. ACM, 2004.

[115] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.

[116] Vassilis Kalofolias, Xavier Bresson, Michael Bronstein, and Pierre Vandergheynst. Matrix completion on graphs. *arXiv:1408.1717*, 2014.

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

[118] Lei Xu and Michael I Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural computation*, 8(1):129–151, 1996.

[119] Craig Knox, Vivian Law, Timothy Jewison, Philip Liu, Son Ly, Alex Frolkis, Allison Pon, Kelly Banco, Christine Mak, Vanessa Neveu, et al. Drugbank 3.0: a comprehensive resource for 'omics' research on drugs. *Nucleic acids research*, 39(suppl_1):D1035–D1041, 2010.

[120] Allan Peter Davis, Cynthia Grondin Murphy, Robin Johnson, Jean M Lay, Kelley Lennon-Hopkins, Cynthia Saraceni-Richards, Daniela Sciaky, Benjamin L King, Michael C Rosenstein, Thomas C Wiegers, et al. The comparative toxicogenomics database: update 2013. *Nucleic acids research*, 41(D1):D1104–D1114, 2012.

[121] Damjan Krstajic, Ljubomir J Buturovic, David E Leahy, and Simon Thomas. Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of cheminformatics*, 6(1):10, 2014.

[122] Feixiong Cheng, István A Kovács, and Albert-László Barabási. Network-based prediction of drug combinations. *Nature communications*, 10(1):1197, 2019.

[123] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[124] Elliot Brown. *Medical Dictionary for Regulatory Activities (MedDRA®)*, chapter 13, pages 168–183. John Wiley & Sons, Ltd, 2007.

[125] Katherine Nolan, Jacqueline Kamrath, and Jacob Levitt. Lindane toxicity: a comprehensive review of the medical literature. *Pediatric dermatology*, 29(2):141–146, 2012.

[126] Daniel L Sudakin. Fatality after a single dermal application of lindane lotion. *Archives of environmental & occupational health*, 62(4):201–203, 2007.

[127] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS'00, pages 535–541, Cambridge, MA, USA, 2000. MIT Press.

[128] Yves Moreau and Léon-Charles Tranchevent. Computational tools for prioritizing candidate genes: boosting disease gene discovery. *Nature Reviews Genetics*, 13:523–536, 2012.

[129] M Oti and HG Brunner. The modular nature of genetic diseases. *Clinical Genetics*, 71(1):1–11, 2007.

[130] Xiujuan Wang, Natali Gulbahce, and Haiyuan Yu. Network-based methods for human disease gene prediction. *Briefings in Functional Genomics*, 10(5):280–293, 2011.

[131] G. M. Lathrop and J. M. Lalouel. Easy calculations of lod scores and genetic risks on small computers. *American journal of human genetics*, 36 2:460–5, 1984.

[132] Helen M Colhoun, Paul M McKeigue, and George Davey Smith. Problems of reporting genetic associations with complex outcomes. *The Lancet*, 361(9360):865 – 872, 2003.

[133] Euan A. Adie, Richard R. Adams, Kathryn L. Evans, David J. Porteous, and Ben S. Pickard. Speeding disease gene discovery by sequence based candidate prioritization. *BMC Bioinformatics*, 6(1):55, Mar 2005.

[134] Kai Wang, Mingyao Li, and Hakon Hakonarson. Annovar: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Research*, 38(16):e164, 2010.

[135] Jing Chen, Eric E. Bardes, Bruce J. Aronow, and Anil G. Jegga. Toppgene suite for gene list enrichment analysis and candidate gene prioritization. *Nucleic Acids Research*, 37(suppl2):W305–W311, 2009.

[136] Martin Oti, Berend Snel, Martijn A Huynen, and Han G Brunner. Predicting disease genes using protein–protein interactions. *Journal of medical genetics*, 43(8):691–698, 2006.

[137] Cecily J Wolfe, Isaac S Kohane, and Atul J Butte. Systematic survey reveals general applicability of" guilt-by-association" within gene coexpression networks. *BMC bioinformatics*, 6(1):227, 2005.

[138] Stephen Oliver. Proteomics: guilt-by-association goes global. *Nature*, 403(6770):601, 2000.

[139] Sebastian Köhler, Sebastian Bauer, Denise Horn, and Peter N. Robinson. Walking the interactome for prioritization of candidate disease genes. *The American Journal of Human Genetics*, 82(4):949 – 958, 2008.

[140] BoLin Chen, Min Li, JianXin Wang, and Fang-Xiang Wu. Disease gene identification by using graph kernels and markov random fields. *Science China Life Sciences*, 57(11):1054–1063, Nov 2014.

[141] Saket Navlakha and Carl Kingsford. The power of protein interaction networks for associating genes with diseases. *Bioinformatics*, 26(8):1057–1063, 2010.

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

[143] Kwang-Il Goh, Michael E. Cusick, David Valle, Barton Childs, Marc Vidal, and Albert-László Barabási. The human disease network. *Proceedings of the National Academy of Sciences*, 104(21):8685–8690, 2007.

[144] Nagarajan Natarajan and Inderjit S. Dhillon. Inductive matrix completion for predicting gene–disease associations. *Bioinformatics*, 30(12):i60–i68, 2014.

[145] Kuo Yang, Ruyu Wang, Guangming Liu, Zixin Shu, Ning Wang, Runshun Zhang, Jian Yu, Jianxin Chen, Xiaodong Li, and Xuezhong Zhou. Hergepred: Heterogeneous network embedding representation for disease gene prediction. *IEEE Journal of Biomedical and Health Informatics*, 2018.

[146] JQ Li, ZH Rong, X Chen, GY Yan, and ZH You. Mcmda: Matrix completion for mirna-disease association prediction. *Oncotarget*, 8(13):21187–21199, Mar 2017.

[147] Xing Chen, Lei Wang, Jia Qu, Na-Na Guan, and Jian-Qiang Li. Predicting mirna–disease association based on inductive matrix completion. *Bioinformatics*, 34(24):4256–4265, 2018.

[148] Chengqian Lu, Mengyun Yang, Feng Luo, Fang-Xiang Wu, Min Li, Yi Pan, Yaohang Li, and Jianxin Wang. Prediction of lncrna–disease associations based on inductive matrix completion. *Bioinformatics*, 34(19):3357–3364, 2018.

[149] Xing Chen, Jun Yin, Jia Qu, and Li Huang. Mdhgi: Matrix decomposition and heterogeneous graph inference for mirna-disease association prediction. *PLoS computational biology*, 14(8):e1006418, 2018.

[150] 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.

[151] Kasper Lage, E Olof Karlberg, Zenia M Størling, Páll I Olason, Anders G Pedersen, Olga Rigina, Anders M Hinsby, Zeynep Tümer, Flemming Pociot, Niels Tommerup, et al. A human phenome-interactome network of protein complexes implicated in genetic disorders. *Nature biotechnology*, 25(3):309, 2007.

[152] Xiangxiang Zeng, Yuanlu Liao, Yuansheng Liu, and Quan Zou. Prediction and validation of disease genes using hetesim scores. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 14(3):687–695, 2017.

[153] Yota Sato, Jun Shibasaki, Noriko Aida, Kazuya Hiiragi, Yuichi Kimura, Moe Akahira-Azuma, Yumi Enomoto, Yoshinori Tsurusaki, and Kenji Kurosawa. Novel col4a1 mutation in a fetus with early prenatal onset of schizencephaly. *Human genome variation*, 5(1):4, 2018.

[154] Amar Mukund and Shiv Kumar Sarin. Budd–chiari syndrome: a focussed and collaborative approach, 2018.

[155] Xue Chen, Xunlun Sheng, Wenjuan Zhuang, Xiantao Sun, Guohua Liu, Xun Shi, Guofu Huang, Yan Mei, Yingjie Li, Xinyuan Pan, et al. Guca1a mutation causes maculopathy in a five-generation family with a wide spectrum of severity. *Genetics in Medicine*, 19(8):945, 2017.

[156] Jae Seok Lim, Ramu Gopalappa, Se Hoon Kim, Suresh Ramakrishna, Minji Lee, Woo-il Kim, Junho Kim, Sang Min Park, Junehawk Lee, Jung-Hwa Oh, et al. Somatic mutations in tsc1 and tsc2 cause focal cortical dysplasia. *The American Journal of Human Genetics*, 100(3):454–472, 2017.

[157] George Karypis. Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 247–254. ACM, 2001.

[158] George Ning, Xia & Karypis. Sparse linear methods with side information for top-n recommendations. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 155–162. ACM, 2012.

[159] Kai-Yang Chiang, Cho-Jui Hsieh, and Inderjit S Dhillon. Matrix completion with noisy side information. In *Advances in Neural Information Processing Systems*, pages 3447–3455, 2015.

[160] Nikhil Rao, Hsiang-Fu Yu, Pradeep K Ravikumar, and Inderjit S Dhillon. Collaborative filtering with graph information: Consistency and scalable methods. In *Advances in neural information processing systems*, pages 2107–2115, 2015.

[161] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In *ICDM*, volume 10, pages 176–185. Citeseer, 2010.

[162] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

[163] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 153–162. ACM, 2016.

[164] Zhao Kang, Chong Peng, and Qiang Cheng. Top-n recommender system via matrix completion. In *AAAI*, pages 179–185, 2016.

[165] Santosh Kabbur, Xia Ning, and George Karypis. Fism: factored item similarity models for top-n recommender systems. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 659–667. ACM, 2013.

[166] Yao Cheng, Liang Yin, and Yong Yu. Lorslim: low rank sparse linear methods for top-n recommendations. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 90–99. IEEE, 2014.

[167] Yong Zheng, Bamshad Mobasher, and Robin Burke. Cslim: Contextual slim recommendation algorithms. In *Proceedings of the 8th ACM Conference on Recommender Systems*, pages 301–304. ACM, 2014.

[168] Evangelia Christakopoulou and George Karypis. Local item-item models for top-n recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 67–74. ACM, 2016.

[169] Mark Levy and Kris Jack. Efficient top-n recommendation by linear regression. In *RecSys Large Scale Recommender Systems Workshop*, 2013.

[170] Evangelia Christakopoulou and George Karypis. Hoslim: Higher-order sparse linear method for top-n recommender systems. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 38–49. Springer, 2014.

[171] Eli M Gafni and Dimitri P Bertsekas. Two-metric projection methods for constrained optimization. *SIAM Journal on Control and Optimization*, 22(6):936–964, 1984.

[172] Sparse linear methods (slim) for top-n recommender systems version 1.0, 2012.

[173] Grouplens. Movielens 100k dataset, 4 1998.

[174] Netflix dataset, 7 2009.

[175] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517. International World Wide Web Conferences Steering Committee, 2016.

[176] Information courtesy of the internet movie database, 2018.

[177] Marius Kaminskas and Derek Bridge. Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 7(1):2, 2017.

[178] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, 20(4):1956–1982, March 2010.

[179] Rickey E Carter, Zachi I Attia, Francisco Lopez-Jimenez, and Paul A Friedman. Pragmatic considerations for fostering reproducible research in artificial intelligence. *NPJ digital medicine*, 2, 2019.

[180] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 101–109. ACM, 2019.

[181] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*, pages 689–698. International World Wide Web Conferences Steering Committee, 2018.

[182] George A Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

[183] Bo Jin, Haoyu Yang, Cao Xiao, Ping Zhang, Xiaopeng Wei, and Fei Wang. Multitask dyadic prediction and its application in prediction of adverse drug-drug interaction. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

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

[185] Ted T Ashburn and Karl B Thor. Drug repositioning: Identifying and developing new uses for existing drugs. *Nature Reviews Drug Discovery*, 3(8):673–683, aug 2004.

[186] Sudeep Pushpakom, Francesco Iorio, Patrick A. Eyers, K. Jane Escott, Shirley Hopper, Andrew Wells, Andrew Doig, Tim Guilliams, Joanna Latimer, Christine McNamee, Alan Norris, Philippe Sanseau, David Cavalla, and Munir Pirmohamed. Drug repurposing: progress, challenges and recommendations. *Nature Reviews Drug Discovery*, 18:41–58, oct 2018.

[187] Rianne van den Berg, Thomas N Kipf, and Max Welling. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, 2017.


ESSE · QUAM · VIDERI