Multi-task and Multi-view Learning for Predicting Adverse Drug Reactions

By

Jintao Zhang

Submitted to the graduate degree program in Bioinformatics and the
Faculty of the Graduate School of the University of Kansas
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Dr. Jun Huan, Chairperson

Dr. Gerald H. Lushington

Committee members

Dr. Ilya Vakser

Dr. Eric Deeds

Dr. John Karanicolas

Date defended: August 7, 2012

The Dissertation Committee for Jintao Zhang certifies
that this is the approved version of the following dissertation :

Multi-task and Multi-view Learning for Predicting
Adverse Drug Reactions

Dr. Jun Huan, Chairperson

Date approved: August 7, 2012

Abstract

Adverse drug reactions (ADRs) present a major concern for drug safety and are a major obstacle in modern drug development. They account for about one-third of all late-stage drug failures, and approximately 4% of all new chemical entities are withdrawn from the market due to severe ADRs. Although off-target drug interactions are considered to be the major causes of ADRs, the adverse reaction profile of a drug depends on a wide range of factors such as specific features of drug chemical structures, its ADME/PK properties, interactions with proteins, the metabolic machinery of the cellular environment, and the presence of other diseases and drugs. Hence computational modeling for ADRs prediction is highly complex and challenging. We propose a set of statistical learning models for effective ADRs prediction systematically from multiple perspectives.

We first discuss available data sources for protein-chemical interactions and adverse drug reactions, and how the data can be represented for effective modeling. We also employ biological network analysis approaches for deeper understanding of the chemical biological mechanisms underlying various ADRs. In addition, since protein-chemical interactions are an important component for ADRs prediction, identifying these interactions is a crucial step in both modern drug discovery and ADRs prediction. The performance of common supervised learning methods for predicting protein-chemical interactions have been largely limited by insufficient availability of binding data for many proteins. We propose two multi-task learning (MTL) algorithms for jointly predicting active compounds of multiple proteins, and our methods outperform existing states of the art significantly. All these related data, methods, and preliminary

results are helpful for understanding the underlying mechanisms of ADRs and further studies.

ADRs data are complex and noisy, and in many cases we do not fully understand the molecular mechanisms of ADRs. Due to the noisy and heterogeneous data set available for some ADRs, we propose a sparse multi-view learning (MVL) algorithm for predicting a specific ADR - drug-induced QT prolongation, a major life-threatening adverse drug effect. It is crucial to predict the QT prolongation effect as early as possible in drug development. MVL algorithms work very well when complex data from diverse domains are involved and only limited labeled examples are available. Unlike existing MVL methods that use ℓ_2-norm co-regularization to obtain a smooth objective function, we propose an ℓ_1-norm co-regularized MVL algorithm for predicting QT prolongation, reformulate the objective function, and obtain its gradient in the analytic form. We optimize the decision functions on all views simultaneously and achieve 3-4 fold higher computational speedup, comparing to previous ℓ_2-norm co-regularized MVL methods that alternately optimizes one view with the other views fixed until convergence. ℓ_1-norm co-regularization enforces sparsity in the learned mapping functions and hence the results are expected to be more interpretable.

The proposed MVL method can only predict one ADR at a time. It would be advantageous to predict multiple ADRs jointly, especially when these ADRs are highly related. Advanced modeling techniques should be investigated to better utilize ADR data for more effective ADRs prediction. We study the quantitative relationship among drug structures, drug-protein interaction profiles, and drug ADRs. We formalize the modeling problem as a multi-view (drug structure data and drug-protein interaction profile data) multi-task (one drug may cause multiple ADRs and each ADR is a task) classification problem. We apply the co-regularized MVL on each ADR and use regularized MTL to increase the total sample size and improve model performance. Experimental studies on the ADR data set demonstrate the effectiveness of our MVMT

algorithm. Cluster analysis and significant feature identification using the results of our models reveal interesting hidden insight.

In summary, we use computational methods such as biological network analysis, multi-task learning, multi-view learning, and inductive multi-view multi-task learning to systematically investigate the modeling of various ADRs, and construct highly accurate models for ADRs prediction. We also have significant contribution on proposing novel supervised and semi-supervised learning algorithms, which con be applied to many other real-world applications.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Adverse drug reactions (ADRs) are undesirable effects of drugs on human bodies occurring at the proper drug doses for an appropriate indication. ADRs are a subset of more generally named *adverse drug events* or *side effects* [Hammann et al., 2010], which have increasingly attracted broad public concern on the danger to patients and major attention to the huge burden on national public health care system in the past decades. An ADR is defined as *serious* in clinical studies if the patient outcome is one or more of the following: death, life-threatening, hospitalization, disability, congenital anomaly, and requiring intervention to prevent permanent impairment or damage [Med, 2010]. In general, each drug may have multiple adverse reactions, which may or may not be serious ADRs.

Various studies have estimated that ADRs cause or contribute to a two-day increase in the average length of hospitalization, and about 6.7% of all hospitalizations with a fatality rate of 0.32% [Eichelbaum et al., 2006, Lazarou et al., 1998], that is, ADRs affect over two million hospitalized patients and cause about 100,000 deaths in U.S. annually. In addition, it is estimated that over 350,000 ADRs occur in U.S. nursing homes each year [Gurwitz et al., 2000]. ADRs are the fourth leading cause of death in U.S., following cancer and heart diseases, and cause more deaths than pulmonary disease, diabetes, and AIDS. Other western countries show similar statistics [van der Hooft et al., 2006, Leone et al., 2008]. Moreover, one out of five injuries or deaths per year to hospitalized patients may be as a result of ADRs [Leape et al., 1991], and serious ADRs account for 1.8-6.2% of all hospitalized cases at any age [Onder et al., 2002]. A two-fold greater mean length of stay, cost and mortality has been reported for hospitalized patients experiencing an ADR compared to a control group of patients without an ADR [Classen et al., 1997]. Consequentially, in U.S. the total cost of drug-related morbidity and mortality is estimated to be $136 billion annually [Johnson & Bootman, 1995], which is as much as the drug treatment itself [Eichelbaum et al., 2006], and higher than that spent on cardiovascular or diabetic care [Johnson & Bootman, 1995].

Moreover, drug discovery and development is an expensive and lengthy process, for instance, a successful drug generally costs $800 million and needs continuous investment and efforts for more than 10 years [DiMasi et al., 2003]. Unexpected serious ADRs have been one of the major obstacles in drug development and clinical applications, and hence a big nightmare to pharmaceutical companies. They account for about one-third of all late-stage (phase 2 and 3 clinical trials) drug failures [Kennedy, 1997], and approximately 4% of all new chemical entities are withdrawn from the market due to serious ADRs [Moore et al., 2007]. For instance, the numbers of reported ADRs and related deaths have increased both for about 2.6 times, and at least 34 drugs have been withdrawn from the market during the period of 1998–2005 [Need et al., 2005]. Such late-stage failures make clinical drugs unsafe for patients, and the cost associated with late-stage failures increases exponentially as the development of a drug proceeds. Lasser *et al.* [Lasser et al., 2002] conducted a detailed analysis of 548 new chemical entities approved during the period of 1975–1999, and found that 56 drugs (10.2%) acquired a new black box warning or were withdrawn, of which 45 (8.2%) acquired one or more black box warnings, and 16 (2.9%) were withdrawn from the market due to serious ADRs. Since 1993 seven drugs were approved, marketed and subsequently withdrawn from the market, and over 1,000 deaths were reported to be associated with them [Lasser et al., 2002]. DiMasi *et al.* [DiMasi, 2002] estimated that if the clinical success rate increases from 1/5 to 1/3, pharmaceutical companies can save $221 million on the drug development process and $129 million for reducing 25% of the total development and regulatory review time.

Therefore, it is of great importance to identify ADRs as early as possible in the drug discovery and development pipeline to avoid further investing and even marketing drugs with serious ADRs. However, the causes and molecular mechanisms underlying each ADR vary significantly. Drug molecules are designed to bind to target proteins and change their bioactivities to achieve therapeutic effects, and hence drug targets and off-targets are involved into the generation of ADRs. Some ADRs are also resulted from the interactions of a drug (or its metabolites) with other drugs or cellular components which are important in normal cellular functions [Lin et al., 2008], including the drug's main therapeutic target, off-target proteins, nucleic acids, and etc. The adverse reaction profile of a drug (i.e. the set of proteins that the drug binds to) depends not only on the drug's chemical structure and the ADME/PK properties, but also on the target protein structures, the binding sites, the metabolic machinery of the cellular environment, the genetic makeup of the cells, and etc.

In this thesis, we aim to establish a universally applicable and externally predictive ADRs modeling framework incorporating ADR-related data from multiple diverse sources. The rest of this thesis is organized as follows. We will discuss the background of modeling ADRs prediction as a computational problem in Chapter 2 and review related work on ADRs prediction in Chapter 3. We introduce the available data sources related to ADRs and a variety of feature extraction methods for chemical compounds, and analyze data in PubChem with biological network approaches in Chapter 4. Utilizing the protein-chemical interaction networks existing in PubChem, DrugBank and other online databases, we propose novel network features for effectively predicting potential drug targets in Chapter 5. In Chapter 6, we propose multi-task learning (MTL) approaches to jointly predicts multiple interacting proteins for a given chemical simultaneously, which is important for obtaining the interacting protein profiles for drugs.

We hypothesize that by integrating multiple sources of data related to ADRs such as drug structures and interacting protein profiles using advanced machine learning techniques, our proposed multi-task and multi-view learning algorithms can significantly improve the performance of ADRs prediction models, and provide critical clues for explaining causes of ADRs. In Chapter 7, we propose a sparse multi-view learning algorithm to effective predict a single ADR for a set of drugs, and achieve much better model performance. Here MVL is an advanced semi-supervised learning technique that learns multiple classifiers for the same data set with multiple views, one for a view. We further develop a novel multi-view multi-task (MVMT) learning algorithm to learn multiple related task jointly in Chapter 8, where each task has multi-view data. Outstanding performance improvement is achieved using our MVMT methods on several benchmark data sets. In Chapter 9, we apply the MVMT learning methods described in Chapter 8 to jointly predict multiple ADRs and achieve excellent performance. Finally, we interpret prediction results and summarize all works in this thesis.

Chapter 2

Background

Although ADRs may not be preventable, they can be anticipated and predicted. Conventional experimental methods for ADRs identification first screen compounds in animal models, and conduct a screening of preclinical drugs against a large number of enzymes and receptors *in vitro*, and are hence expensive and time-consuming [Whitebread et al., 2005, Bass et al., 2004, MacDonald et al., 2007]. Experimental methods can only enter the drug development pipeline at a relatively late stage, not as early as possible. Computational approaches are hence indispensable, complementary tools to experimental methods with less time and expenses.

There are three knowledge sources of ADRs: toxicity studies on animal models, drug clinical trials, and clinical reports in the ADR reporting systems. Animal testing and clinical trials are widely used to evaluate ADRs during the pre-market phase of a drug. Results from animal models may not be always generalizable to humans due to specie differences. Clinical trials are usually conducted on a very limited population of patients (up to 2,000). Rare ADRs occurring less than 1 out of 100,000 patients often go undetected during clinical trials. Hence a continuous post-market monitoring of ADRs is mandatory [Amery, 1999]. Once a drug is commercialized, Food and Drug Administration (FDA) primarily uses a voluntary Adverse Event Report Systems (AERS), combined with the literature to monitor ADRs. AERS is an information database designed to support the FDA's post-marketing safety surveillance program for all approved drug and therapeutic biologic products. Voluntary ADR reporting systems, however, contain a large amount of ADR information with errors, noisy, bias, and uncertainty (`http://www.fda.gov/medwatch/`). Consequentially, it is highly challenging to employ theoretic and computational principles to enable the effective ADR prediction for safe pharmaceutical care and achieve deeper understanding of the underlying molecular mechanisms of ADRs.

Off-target drug interactions are generally considered to be the major causes of ADRs [Onder et al., 2002]. However, the adverse reaction profile of a drug depends on a wide range of factors such as specific features of drug chemical structures, its ADME/PK (Absorption, Distribution, Metabolism, Excretion and Pharmacokinetics) properties, specific and non-specific interactions with proteins, the metabolic machinery of the cellular environment, the genetic makeup of the cells, the developmental stage of the hosts, and the presence of other diseases and drugs [Faulon et al., 2008, Jacob & Vert, 2008, Onder et al., 2002]. The extraordinary complexity of modeling and predicting ADRs presents serious challenges to existing data mining and machine learning approaches.

Computationally predicting ADRs is a daunting routine. First, biological effects of compounds are defined by their chemical structures, which are characterized by a large number of computed chemical descriptors (features) for constructing QSAR models. Second, the data sets of drugs that are reported to have a specific ADR are often rather small, but the additional data for related ADRs may be available. The incorporation of data from related but distinct ADRs is attractive but needs to be done with care. Third, the data for each ADR often come from multiple sources, and some of which may be useless or even present contradictory information for ADRs prediction. Identifying the right information sources is hence critical.

Several computational approaches have been proposed to predict adverse reactions of preclinical and commercial drugs either using drug chemical structures [Yap et al., 2004, Li et al., 2005, Bhavani et al., 2006] or using ligand-protein *inverse docking* by taking into account the target proteins and identifying non-therapeutic, alternative target proteins that are predicted to bind a given drug [Rockey & Elcock, 2002, Ji et al., 2006, Xie et al., 2007]. These methods suffer from the limited availability of chemical data for drugs known to cause a particular ADRs, or lack of structural data on protein targets. Traditional *in silico* approaches employed for the prediction of chemical effects *in vivo* typically either rely on the use of calculated physical properties or structural features of chemicals (i.e., chemical descriptors) or attempt to relate the results of short term assays to predict the *in vivo* outcome of chemical exposure (*in vitro* - *in vivo* correlations). Although both approaches have been somewhat successful, existing tools and models should be substantially improved to enable their reliable application in ongoing drug discovery and drug safety regulation. Recently chemogenomics approaches that integrate heterogeneous data from protein-chemical interactions, chemical ADME properties, and gene-chemical interactions have attracted attention as potential means of ADRs prediction [Faulon et al., 2008, Jacob & Vert, 2008]. So far there has been no systematic effort to employ diverse informatics approaches toward rigorous prediction of ADRs.

In this thesis we aim to systematically examine the relevant data and develop remedies for effective modeling of ADRs. We posit that the ultimate success of computational ADRs models depends on three major factors: (1) selection of the subset of ADRs for modeling and the quality of primary data used for model development, (2) comprehensive development of a variety of learning algorithms for modeling noisy and heterogeneous data to enable effective ADRs prediction, and (3) rigorous evaluation of the learning algorithms on a wide range of data sets from a variety of domains for continuous methodology improvements. Since the causes and molecular mechanisms of different ADRs vary significantly, we need to refine a subset of ADRs with similar mechanisms and develop appropriate models accordingly. Here we focus on those ADRs resulting from interactions between drug chemicals and their desired targets or other related cellular proteins, and ADRs with other mechanisms are beyond the scope of this thesis.

Chapter 3

Related Work

In this chapter, we review computational methods in existing literature that have been utilized for identifying ADRs. A straightforward way is to develop quantitative structure-activity relationship (QSAR) models for an ADR using chemoinformatics, a rapidly emerging research discipline that uniquely combines computational, statistical, and informational methodologies with key concepts from chemistry. It is distinct from other computational modeling approaches in that it uses unique representations of chemical structures in the form of multiple chemical descriptors, it has its own metrics for defining chemical similarity and diversity in chemical compound libraries, and it applies a wide array of statistical, data mining, and machine learning techniques to very large collections of chemicals in order to establish robust relationships between chemical structures and its physical or biological properties. Chemoinformatics tools have played a central role in the analysis and interpretation of structure-property data collected by means of modern high throughput screening [Schneider et al., 2008]. QSAR models establish quantitative relationships between chemical structures characterized by chemical descriptors and a target property, e.g., biological activity of chemicals in specific biological assays. Validated and externally predictive models can be applied to screen virtual chemical libraries to retrieve compounds with desired properties [Tropsha, 2010].

Some computational studies of ADRs have been mainly concerned with the prediction of individual adverse effects such as hERG inhibition [Nisius & Goller, 2009]. Chen *et al.* [Yap et al., 2004] applied support vector machine (SVM) classification to predict an important ADR - torsade de pointes (TdP) from the linear salvation energy relationships (LSER) of compounds [Pip, 2010]. They extended their study to predict genotoxicity of chemical compounds using a set of 199 selected molecular descriptors [Li et al., 2005] with various statistical learning methods such as SVM, probabilistic neural network (PNN), and k-nearest neighbors (KNN). Bhavani *et al.* [Bhavani et al., 2006] also conducted an SVM-based prediction of the TdP adverse effect with sophisticated substructure-based feature extraction methods over a large diverse drug set, and yielded high prediction accuracy. Overall, these approaches are ligand-based and predict ADRs only using drug chemical structures. It is hard to generalize them to other ADRs due to lack of target information.

Approaches which take target structures into account have thereby been presented. One such effort [Rockey & Elcock, 2002] employed a drug-docking algorithm named AutoDock to distinguish between the selectivity of different kinases for ADR virtual screening and showed very high specificity and sensitivity. Ji *et al.* [Ji et al., 2006] utilized an inverse docking approach (INVDOCK) to predict ADRs of 11 marketed anti-HIV drugs by searching for their interactions with ADR-related proteins with 86-89% prediction accuracy achieved. Moreover, Xie and coworkers [Xie et al., 2007] developed an integrated approach to studying protein-ligand interactions on a structural proteome-wide scale by taking ADR-related information such as protein functional sites, small molecule structures, and protein-ligand binding affinity profiles together into account. They take a single drug molecule and look for how it might bind to as many of the proteins encoded by the human proteome as possible in PDB with drug-target inverse docking. One limitation of these structure-based approaches is that they usually employed a relatively small amount of data due to the limited availability of the related data for each ADR. In addition, docking is an expensive process due to the huge orientation space of proteins, and also not very accurate due to the imperfection of current scoring functions.

Moreover, computational approaches for explaining ADRs at the molecular level have also been proposed. Bender *et al.* [Bender et al., 2007] developed a systematic method to predict ADRs and off-target effects using only drug chemical structures with a multi-category Bayes model. It is notable that features of the models are interpretable and back-projectable to chemical structures, and hence new hypotheses linking targets and adverse effects can be proposed to rationally engineer out those adverse effects. Scheiber *et al.* [Scheiber et al., 2009] analyzed a set of compounds sharing a common toxicity and predicted the top 5 targets for each compound. By comparing the pathways affected by toxic compounds with pathways modulated by nontoxic compounds, links between toxicity and the pathways probably being responsible for it can be established.

There are some other methods that predict ADRs from a totally different perspective. Mushiroda *et al.* [Mushiroda et al., 2005] conducted genome-wide association studies to identify genetic variations that might be associated with adverse cardiovascular events in 72 renal transplant recipients using gene-based SNPs (single-nucleotide polymorphisms), and developed a scoring system that was able to predict individual risks for cardiovascular toxicity. Lin *et al.* [Lin et al., 2008] developed a tool which integrates drug and drug target knowledge bases, and built up drug-target networks whose topological analysis can reveal drug interaction complexity for every ADR report. This method can be used for the analysis and prediction on ADRs cases and also drug target assessment in the early drug discovery process.

Although ligand-based and structure-based methods achieved good results for some ADR prediction, a new trend is to integrate a variety of ADR-related information such as drug chemical structures, protein functional sites, protein ligand binding profiles, cellular pathways, genotypic information, and drug-target knowledge bases together, and more promising and generalized prediction of ADRs has been obtained [Rockey & Elcock, 2002, Ji et al., 2006, Xie et al., 2007]. This kind of strategy is somehow similar to the basic concepts in chemogenomics. Chemogenomics by its definition is the investigation of classes of compounds (libraries) against families of functionally related proteins, and provides a systematic way of analyzing chemical-biological interactions and discovering active and/or selective ligands for biological relevant targets. Here we borrow this concept to describe that we utilize ADR-related information from both a drug and its targets and off-targets to predict its ADR profile. The problem is then converted to how to measure the similarity between two pairs of drugs and targets, e.g. (D_1, P_1) and (D_2, P_2), where D_1, D_2 are feature vectors from the two drugs, and P_1, P_2 are feature vectors from the two proteins. A typical way is to measure the similarity of the two drugs and that of the two proteins separately using drug kernels and protein kernels, say $K_d(D_1, D_2)$, and $K_p(P_1, P_2)$ respectively, and then multiply or sum K_d and K_p to obtain the similarity between the two pairs of drugs and targets.

The existing methods that include target protein structures in ADR prediction usually focus on one particular ADR class, and are often satisfied by some successful case studies. They lack a global view of the ADR prediction problem, and their predictions of ADRs based on incompletely included ADR-related information are hardly interpretable. Our understanding on ADR profiles and the underlying molecular mechanisms is still ambiguous, and there is a critical gap between the performance of current methods of ADR prediction and our goal of understanding ADRs more deeply. Without understanding ADRs and the causing mechanisms at a molecular level, it is very difficult to design more clinically safe drugs, guide lead compound identification and optimization in rational drug development, and prevent potential ADRs from threatening the health of patients. Therefore more sophisticated methods must be proposed to penetrate the gap of our knowledge on ADRs.

Our proposed methods are novel since they integrate various ADR-related information using chemogenomics concepts [Faulon et al., 2008, Jacob & Vert, 2008, Schneider et al., 2004] and advanced data mining techniques to build statistical models for ADRs prediction at the molecular level. We also realize that the ADRs associated with a drug are not independent, and some ADRs may result from the same molecular mechanism. With complete implementation of our methods using all these data mining techniques, we expect to draw a global picture of ADR prediction and gain deeper understanding on the underlying molecular mechanisms of ADRs. Our sophisticated experimental designs based on integrated ADR-related data will help understand molecular mechanisms of ADRs during the model construction process. By mining the associations of ADRs, we might be able to identify the drug functional groups that cause the ADRs and hence guide lead optimization. Finally, if some predicted side effects can be beneficial to patients, the purpose of drug development can be changed to better fit the ADR profile of the drug.

Chapter 4

Related Data, Methods and Preliminary Results

In this chapter, we discuss how we collect and analyze data related to protein-chemical interactions and adverse drug reactions, how to extract chemical features from raw data, how to select optimal sets of descriptors to achieve best model performance, and eventually systematically analyze the data in PubChem using biological network analysis. The basic concepts and methods in all these works are indispensable and critical for further studies.

4.1 Related Data

Intense growth in drug development investment in the past decade has not yet produced significant progress on the discovery of new drugs and the validation of new drug targets: the average number of novel drugs entering the global market each year has remained roughly constant (approximately 26), along with only about 6-7 novel drug targets introduced annually [Adams & Brantner, 2006, Yildirim et al., 2007, Cokol et al., 2005]. Moreover, the success rate of translating new drug candidates into FDA-approved drugs significantly decreased [Kola & Landis, 2004], mainly due to lack of efficacy or adverse drug reactions, each of which accounts for about 30% of late-stage drug failures in clinical development [Kennedy, 1997]. The increasing rate of drug attrition challenges the traditional drug design paradigm and makes the current “one disease, one gene, one drug” paradigm [Sams-Dodd, 2005] questionable.

One of the critical steps in drug discovery is the identification of chemical compounds with desired and reproducible binding activity against a specific biomolecular target [Leach, 2001]. This has become a significant challenge in the early stage of drug discovery, since any new drug must not only produce the desired medical response to the disease, but should also minimize any side effects [Wale et al., 2007]. Understanding and predicting the interactions between target proteins and small molecules is hence of great importance in pharmaceutical industry. Our knowledge about the interactions between chemical space and biological space, however, is very limited. Large amounts of data related to the molecular and cellular activities of chemicals (including drugs) were traditionally collected and analyzed only within the pharmaceutical industry. The landscape of publicly available experimental data for pharmaceutical sciences has changed dramatically in very recent years. The following are the three major sources for obtaining the data sets using throughout this thesis.

4.1.1 PubChem

Under these circumstances, the National Health Institute (NIH) launched the Molecular Libraries Initiative (MLI) in 2004 for identifying chemical probes to enhance the chemical biology understanding of therapeutically interesting genes and pathways [Austin et al., 2004]. The MLI especially focused on genes in the “undruggable” part of human genome that has not been well investigated in private sectors for identifying their functions and potential therapeutics. In 2005, NIH launched the Molecular Libraries Probe Production Centers Network (MLPCN) project. The goal of MLPCN is to profile chemicals at the molecular and cellular level such that molecules that modulate biological networks could be identified [Austin et al., 2004]. All the results are being made freely available to researchers in both public and private sectors via a web portal called PubChem (`http://pubchem.ncbi.nlm.nih.gov/`) [Zerhouni, 2003]. Tens of millions of chemicals have been deposited into the PubChem Databases, but only a very small fraction (less than 2%)

have their target protein information linked [Paolini et al., 2006].

PubChem is organized as a set of bioassays and is a good example of contemporary open access databases. Each bioassay records screening results of a set of chemicals against a protein or a cellular environment and each bioassay typically contains screening results of approximately 300,000 chemicals. The latest version of PubChem contains information for more than 50 million chemicals, more than 20,000 bioassays, and links from chemicals to bioassay descriptions, literature, and references. In our study to better predict drug ADRs, we combined data from heterogeneous databases, primarily from PubChem but also including other chemical-protein interaction databases. For each such database, we downloaded the data from the original databases and removed possible duplicates by mapping chemicals to a unique id (PubChem Compound ID).

4.1.2 Literature and Other Online Databases

There are many other sources for obtaining protein-chemical binding data sets, including data used in existing literature and other online databases such as DrugBank [Wishart et al., 2008, Wishart et al., 2006] developed by colleagues. For instance, a widely used data set consisting of 279 Factor Xa inhibitors and 156 inactives is taken from [Fontaine et al., 2005]. The following four data sets include a number of inhibitors and approximately equal number of inactives to each of four target proteins: 1) ACE; 2) COX2; 3) DHFR; and 4) THR [Sutherland et al., 2004]. Chemicals with IC_{50} < 10nM ($pIC_{50} > 8$) are defined as actives and $> 1\mu$M (or $pIC_{50} < 6$) as inactives. Some other data sets can also be extracted manually from the BindingDB database [Liu et al., 2007, Chen et al., 2002a], which contains more than 450 target proteins and their binding chemicals. Two types of binding activity parameters K_i and IC_{50} are provided, and both of them measure the inhibition power of a chemical compound to a specific target protein [Chen et al., 2002b, Chen et al., 2001].

4.1.3 ADR Databases

For effective ADRs prediction, it is important to obtain a set of compounds that are known to cause or contribute to some specific ADRs. Such data have very limited availability and the quality is somewhat unsatisfactory. For instance, a set of 142 compounds that are associated with drug-induced QT prolongation effect are obtained from the home page of University of Arizona CERT (`http://www.azcert.org/`) [Woosley, 2003] and the work by Ouillé *et al.* [Ouillé et al., 2011], by merging these two data sets and removing redundant and/or inconsistent data samples. This data set for the ADR of QT prolongation consists of 28 drug compounds labeled as "TdP risk", 40 labeled as "possible TdP risk", and 74 with other less reliable evidence of QT prolongation. A much more complete data collection for adverse drug reactions has been developed and named the Side Effect Resource (SIDER) database (`http://sideeffects.embl.de/`) [Kuhn et al., 2010]. SIDER is a public, computer-readable drug side effect resource with links between drugs and ADRs, obtained from the Drug Labels provided by the FDA with text mining tools. We mapped the drugs in SIDER to the DrugBank database [Knox et al., 2011,Wishart et al., 2008] via PubChem Compound ID and removed those drugs with trivial or too simple structures, resulting in 797 drugs associated with 1,362 ADR labels, which are represented by the COSTART controlled vocabulary. Apparently, there are much more ADRs than the number of related drugs.

After obtaining the proper data sets, the learning process consists of three components: (i) feature extraction and selection, (ii) predictive model selection, and (iii) model assessment. Feature extraction methods are first used to compute various descriptors for each sample and convert it to a vector, in which each component is a molecular descriptor, e.g. molecular weight, number of hydrogen bonds, and etc. Second, with a feature vector and the corresponding class label as a datum point, a set of such points are divided into three disjoint sets: training, validation, and testing set. Model selection is needed to choose a model of optimal performance, which in practice means selecting best learning parameters based on training and validation sets. Finally, we assess the optimal model on the unused testing set to estimate the prediction error (generalization error) of the selected model.

4.2 Chemical Descriptors

Computers can only recognize numbers. Chemical compounds must be converted into numbers in some means. We call these numbers “chemical descriptors” (features) and those conversion means “feature extraction methods”. The quality of models related to chemicals highly depends on the performance of feature extraction methods. Chemical compounds have well-defined geometric structures that can be easily converted into a connected, weighted, and undirected graph representation. Each chemical has a number of atoms represented as vertices and a number of bonds between atoms represented as edges in the molecular graph. Usually vertices are labeled with the atom element type (atomic symbol or number, e.g. carbon atoms are labeled with C or 6), and edges are labeled with the bond type (bond order or separate integers, use 1, 2, 3, 4 for single, double, triple, and aromatic bonds, respectively). Edges in a graph are undirected because chemical bonds have no associated directionality. Figure 4.1 shows an example of a chemical structure and the corresponding graph representation.

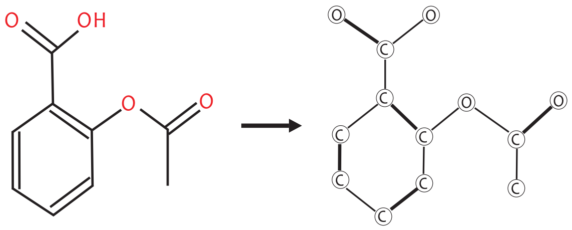

Figure 4.1: A chemical structure and the corresponding graph representation.

There have been many methods that represent a chemical by a set of descriptors based on frequency, molecular properties, topological and geometric substructures [Wale et al., 2007, Deshpande et al., 2005, Horvath et al., 2004], e.g. DRAGON descriptors [Dra, 2008], Daylight fingerprints [Day, 2007], extended connectivity fingerprints (ECFP) [Rogers et al., 2005], Maccs keys [Day, 2007], cyclic patterns and trees [Horvath et al., 2004], signature molecular descriptors [Faulon et al., 2004], and frequent subgraph-based descriptors [Huan et al., 2003, Kuramochi & Karypis, 2004]. All these feature extraction methods have been well developed and have demonstrated their effectiveness and success in many experiments and applications. They can also be used to rapidly predict the physical, chemical, and biological properties of small molecules to screen large database and identify suitable drug candidates [Agrafiotis et al., 2002, C. & Hopkins, 2004, Dobson, 2004].

4.2.1 Important Chemical Descriptors

Here we introduce some sets of commonly used chemical descriptors in detail, including the 20 sets of chemical descriptors provided by DRAGON 5.4 software [Dra, 2008], signature molecular descriptors [Faulon et al., 2004], and frequent subgraph-based descriptors [Huan et al., 2003].

4.2.1.1 DRAGON Descriptors

DRAGON is a commercial software package developed by Milano Chemometrics and QSAR Research Group [Dra, 2008] for computing chemical descriptors that can be used to evaluate molecular structure-activity or property relationships (QSAR/P), as well as for high-throughput virtual screening of chemical databases. Users need molecular structure files (in format SDF, SMILES, etc.) as input, and are given formatted output files. Although DRAGON can work on 2D structures, only much less descriptors can be calculated in this case. To make full use of DRAGON software, 3D optimized structures with all hydrogen atoms included should be used.

DRAGON 5.4, which we used in this work, computes 1664 molecular descriptors that are classified into 20 descriptor sets (or logical blocks): constitutional descriptors (DR01), topological descriptors (DR02), walk and path counts (DR03), connectivity indices (DR04), information indices (DR05), 2D autocorrelations (DR06), edge adjacency indices (DR07), Burden eigenvalues (DR08), topological charge indices (DR09), eigenvalue-based indices (DR10), Randic molecular profiles (DR11), geometrical descriptors (DR12), RDF descriptors (DR13), 3D-MoRSE descriptors (DR14), WHIM descriptors (DR15), GETAWAY descriptors (DR16), functional group counts (DR17), atom-centered fragments (DR18), charge descriptors (DR19), and molecular properties (DR20). For more detailed introduction to DRAGON software and descriptors, refer to the help manual at its homepage (`http://www.talete.mi.it/index.htm`).

For instance, in a molecule with known molecular composition and atom connectivities, functional group counts (DR17) are simply defined as the number of specific functional groups, and atom-centered fragments (DR18) are defined as the number of specific atom types. For each atom-centered fragment, its frequency occurring in the chemical structure is counted, and so far 120 atom-centered fragments defined by Ghose and Crippen [Viswanadhan et al., 1989] are included. Finally, molecular properties (DR20) include a set of heterogeneous molecular descriptors describing physico-chemical and biological properties as well as some molecular characteristics, such as hydrophilic factor, octane-water partition coefficient, molar refractivity, and etc. Since the charge descriptors (DR19) are not available to many chemicals in our data sets, they will be excluded in our study.

4.2.1.2 Signature Molecular Descriptors

Faulon *et al.* [Faulon et al., 2004] proposed an algorithm of signature molecular descriptors by enumerating all molecular signatures with a given height from chemical structures. Specifically, the signature of a molecule is a vector whose components are counts of the number of occurrences of a particular atomic signature in the molecule. An atomic signature is a canonical representation of the subgraph surrounding a particular atom. This subgraph includes all atoms and bonds up to a predefined distance, called signature height, from a given atom. The optimal signature height for chemicals is usually in range of 1-5.

To generate the signature lists of a chemical, a signature translation program named “translator” was downloaded from the homepage of Faulon *et al.* [Faulon et al., 2004]. Given a chemical structure with a predefined signature height, a list of available signatures and their occurring frequencies can be generated very efficiently. The running time for generating the signatures of most chemical structures is some seconds. Given a data set with *m* chemicals, a list of available signatures for each chemical are generated, and then all distinct signatures present in all chemicals (the union of all signature lists, e.g. totally *n* signatures) can be obtained. Finally each chemical will be associated with an n-dimensional vector, in which each component is the number of occurrence of each signature in the chemical.

4.2.1.3 Frequent Subgraph-based Descriptors

Frequent subgraph mining is widely studied since frequent subgraphs are believed to be related to some structural or functional motifs in chemical and biological structures. Huan *et al.* [Huan et al., 2003] developed a depth-first search algorithm for fast frequent subgraph mining (FFSM), which identifies all connected subgraphs that occur more frequently than a predefined frequency threshold σ called *support threshold* in a graph database. Each chemical compound is represented by a binary vector with length equal to the number of all mined subgraphs, and then each subgraph is mapped into a specific vector index. If a chemical compound contains a subgraph, the corresponding bit is set to one, and it is set to zero otherwise. [Huan et al., 2003, Smalter et al., 2008]

By mining all frequent subgraphs from a chemical database, FFSM creates an n-dimensional feature vector for each chemical, and hence provides frequent subgraph-based descriptors. A potential disadvantage of this method is that it is unclear how to select a suitable value of the *support* σ for a given problem. A very high value will fail to discover important substructures whereas a very low value will result in combinatorial explosion of frequent subgraphs. In Figure 4.2, a graph database with 3 graphs is in the top row, and the returned frequent subgraphs are listed in the bottom row with support = 2/3.

4.2.2 Comparison of Chemical Descriptors

It becomes a critical task to select high-quality feature extraction methods for computing chemical descriptors. Since the performance of different chemical descriptors vary significantly, it is necessary to do a case study on comparing the performance of different chemical descriptor sets. There are two basic hypotheses for protein-chemical interaction prediction: 1) chemicals shar-

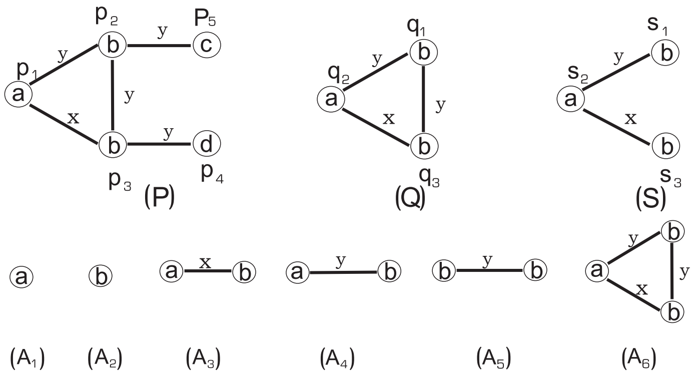

Figure 4.2: A graph database with 3 graphs in upper row, and frequent subgraphs returned by FFSM algorithm with support = 2/3 in lower row.

ing chemical similarity should also share target proteins; 2) targets sharing similar ligands should share similar biological patterns, or binding sites. The classical paradigm here is as follows: if two chemicals (proteins) are considered very similar to each other and we know one of them interacts with a protein (chemical), we would expect that it is very likely for the other to interacting with the same target (chemical) too. The key question of predicting if a chemical interacts with a given protein is how to measure the distance (or similarity) between two chemicals. Since the quality of chemical descriptors play a critical role in such predictions, it is of great importance to investigate which chemical descriptors perform better than the others.

There have been many previous studies on comparing the prediction performance of chemical descriptors. Hert *et al.* [Hert et al., 2004] compared a range of different 2D fingerprints for similarity-based virtual screening, and found that these fingerprints were notably more effective than fingerprints based on a fragment dictionary. They concluded that the combination of these fingerprints with data fusion based on similarity scores provides an effective virtual screening tool in lead discovery. Gedeck *et al.* [Gedeck et al., 2006] analyzed how the quality of QSAR predictions depended on the data sets and descriptor types, and they revealed that none of the descriptors was best for all data sets. Although 2D fragment based descriptors usually performed better than simpler descriptors based on augmented atom types, it was necessary to test them in each individual case.

In addition, Karypis *et al.* [Wale et al., 2007] introduced some new descriptor sets such as graph fragment based descriptors (GF), and conducted a comprehensive comparison of the performance of the newly developed descriptors with Daylight fingerprints [Day, 2007], extended connectivity fingerprints (ECFP) [Rogers et al., 2005], Maccs keys [Day, 2007], cyclic patterns and trees [Horvath et al., 2004] in the context of SVM-based chemical compound classification and ranked retrieval. The goal of his work was to analyze what properties of descriptor spaces were important on providing effective representations for molecular graphs, and their experiments demonstrated that descriptor class ECFP and GF consistently and statistically outperformed previously developed all other descriptor sets.

However, some other widely used chemical descriptors were not covered in these previous studies. For instance, DRAGON 5.4 [Dra, 2008] provided a collection of 20 widely used classes of chemical descriptors. Huan *et al.* [Huan et al., 2003] developed a fast frequent subgraph mining (FFSM) algorithm to generate frequent subgraph-based descriptors for chemicals. Faulon *et al.* [Faulon et al., 2004] developed an algorithm of signature molecular descriptors for both protein sequences and chemicals. A case study on comparing the performance of these 22 sets of chemical descriptors are a necessary and beneficial complement to existing studies. We conduct a detailed performance comparison for the 20 sets of DRAGON descriptors [Dra, 2008], the frequent subgraph-based descriptors [Huan et al., 2003], and the signature molecular descriptors [Faulon et al., 2004] on 14 high-quality chemical data sets collected from literatures and online databases. Here we conduct a comprehensive comparison of the above mentioned chemical descriptor sets, and our results provide important insights on how to select chemical descriptors to achieve optimal performance for protein-chemical interaction prediction.

4.2.3 Some Conclusions

The detailed experimental procedures are not described here. Interested readers may refer to the original paper [Zhang & Huan, 2010]. Experimental results show that FFSM descriptors and signature descriptors (height = 3) perform better than 17 DRAGON descriptors and equivalently to the other three: DR06(2D autocorrelations), DR07(edge adjacency indices), and DR17(functional group counts). In addition, some single DRAGON descriptor sets consistently perform better than other descriptor sets over almost all the data sets. For instance, descriptor class DR06, DR07, and DR17 generally outperform all other DRAGON descriptor sets. DR09(topological charge indices) and DR18(atom-centered fragments) also show decent accuracy compared to other DRAGON descriptors. Finally, using any single DRAGON descriptor class almost always yields better performance than using all DRAGON descriptors together, and the reason might be due to over-fitting.

With wise selection of some DRAGON descriptor sets, we can earn robust and optimal performance for protein-chemical interaction prediction. Our results shed light on selecting descriptor extraction methods wisely to obtain best prediction performance. Simply using all of them or randomly selecting some of them will diminish the prediction performance. A combination of DR06, DR07, DR17, and a few other DRAGON descriptor sets would be a robust and optimal selection. The molecular signatures and FFSM descriptors are satisfactory candidates for virtual screening and protein-chemical interaction prediction.

A major reason that the FFSM descriptors outperform most DRAGON descriptors is that FFSM has implicit functions of descriptor selection, since it only counted frequent subgraphs and infrequent descriptors were removed. In many cases frequent subgraphs have been proven to correspond to some biological or chemical structural motifs. FFSM descriptors can be a better method if approximate matching of subgraphs is allowed. In addition, signature descriptors also perform very excellently after careful tuning of the parameters, partially since they completely characterize a chemical by including geometrical and topological descriptors, atomic properties, chemical bonding and hydrodization information. Finally, through our experimental results we should keep in mind that no descriptors can be claimed the best for all data sets and all situations, and our comparisons and conclusions are made on a statistically average basis.

4.3 The BioAssay Network

PubChem provides valuable chemical genomics information in studying genes, pathways, cells and diseases, especially for identifying promising potential drug targets. The valuable data in PubChem are, however, noisy, high dimensional, with large volume, and contain outliers and errors. For instance, the activity score, which measure the biological activity level of screened compounds, are normalized in many different approaches without a consensus. Here we used network-based approaches to characterize the BioAssay data in PubChem, construct a BioAssay network and analyze its topological features using various statistical tools.

4.3.1 Related Work

Biological network analysis approaches have been intensively applied to organize complex biological data so that retrieval, analysis, and visualization of these data can be highly efficient. In addition, biological network analysis can also reveal important biological patterns and functions that are deeply hidden in mass data repository. For instance, Stelzl *et al.* [Stelzl & *et al.*, 2005] used the Y2H system to generate and analyze a human PPI network, and calculate many interesting and critical patterns and characteristics. This was viewed as an important step toward the complete human protein-protein interactome. Many follow-up studies had been performed and made the human PPI network more and more complete and sophisticated. Chaurasia *et al.* [Chaurasia et al., 2007,Chaurasia et al., 2009] built a comprehensive web platform - the Unified Human Interactome database (UniHI, `http://www.unihi.org`), for querying and accessing human protein-protein interaction data.

Using association data of approved drugs and drug targets obtained from the DrugBank database [Wishart et al., 2006, Wishart et al., 2008], Yildirim *et al.* [Yildirim et al., 2007] built a bipartite network composed of these drugs and their drug targets, which was an important step toward the complete characterization of the global relationship between protein targets of all drugs and all disease-gene products in the human protein interactome. Quantitative topological analyses of this drug-target network revealed that the targets of current drugs are highly overlapped and new drugs tend to bind previously targeted proteins [Yildirim et al., 2007]. Based on disease genes data from the Online Mendelian Inheritance in Man (OMIM) database [Hamosh et al., 2005], Goh *et al.* [Goh et al., 2007] built a bipartite human disease network consisting of 1,284 distinct diseases, 1,777 disease genes, and 2,929 disease-gene associations, and then generated two biologically relevant network projections: human disease network and disease gene network. This network-based approach revealed that genes associated similar disorders are more likely to have interactions between their products and higher expression profiling similarity between their transcripts, indicating the existence of disease-specific functional modules.

4.3.2 Data Statistics

We obtained human PPI data from various sources, including UniHI and CCSB-HI1. UniHI is a unified human PPI network containing over 250,000 human PPIs collected from 14 major PPI sources, including high-quality systematic interactome mapping and literature curation, while CCSB-HI1 is another widely used human PPI database. Data of approved drug targets were downloaded from the DrugBank database [Wishart et al., 2006, Wishart et al., 2008]. As of January 22, 2009, there are 1,493 FDA-approved drugs, and more than 800 human target proteins. In addition, for human essential genes (a human gene was defined as *essential* if a knockout of its mouse ortholog confers lethality), we first extracted mouse essential genes from the Mouse Genome Informatics Database [Eppig et al., 2005], and obtained human essential genes via the human-mouse ortholog associations. Finally, since PubChem only provides protein GI numbers for BioAssay targets, but drug targets and essential genes are identified by official gene symbols (http://www.genenames.org/) in their corresponding databases, we manually converted the GI number of each BioAssay target into official gene symbols. We then mapped BioAssay targets, drug targets, and essential genes via gene symbols into other biological databases, and identified 228, 1,339 and 2,546 entries among the total 21,051 human proteins in UniHI, respectively.

We downloaded all bioassay screening data from the PubChem BioAssay Database. As of

January 2009, 1306 bioassays (1,126 with at least one active compound) and more than 30 million compounds had been deposited into PubChem by a variety of screening centers, and the size of PubChem kept increasing continuously. For each of the 1306 bioassays, tens to hundreds of thousand of compounds were tested either against specific target proteins in vitro or within a cell for investigating disease-related mechanisms. Out of more than 30 million compounds in PubChem, there were totally 151,930 compounds that are active in at least one bioassay, and 555,859 bioassay-active compound pairs across all the bioassays. On average each active compound was active in 3.7 bioassays, and each bioassay had 493.7 active compounds, and therefore a very sparse bipartite network of bioassays and compounds can be observed.

In addition, 680 bioassays were found associated with at least one target protein and were considered target-based, and the rest 626 bioassays were hence assumed cell-based bioassays. In Fig. 4.3a, we summarized the therapeutic focus of all cell-based bioassays, and found that cancer studies (32%, anticancer and tumor growth inhibition), cell death (17%), and stem cell (10%) are the three most common types of screenings. We have found 289 distinct protein GI (gene identifier) numbers for all target-based bioassays, and 215 of them had official associated gene symbols. We then obtained the target protein subcellular locations for all target-based bioassays from the Gene Ontology database and describe their distributions in Fig. 4.3b, which demonstrates that the percentage of currently selected MLI membrane protein targets (19%) was significantly smaller than the 50% ratio of current drug targets that are membrane proteins ($\boldsymbol{P} < 0.001$, Chi-squared 1-degree test) [Drews, 2000]. In addition, we explored the molecular functions of the MLPCN targets by mapping their GI numbers into EC numbers, resulting in 113 proteins with EC numbers that were involved with 253 bioassays. We used the first two numbers of their EC numbers to classify them, as shown in Fig. 4.3c, which revealed that hydrolases (46%) and transferases (38%) were the major function classes of MLPCN target enzymes. Finally, for target-based bioassays, the distribution of organisms from which the target proteins were taken showed that 81% of MLPCN target proteins were from human beings, and the remaining targets were from animals (6%), bacteria (6%), viruses (4%), and other miscellaneous organisms (2%) (Fig. 4.3d).

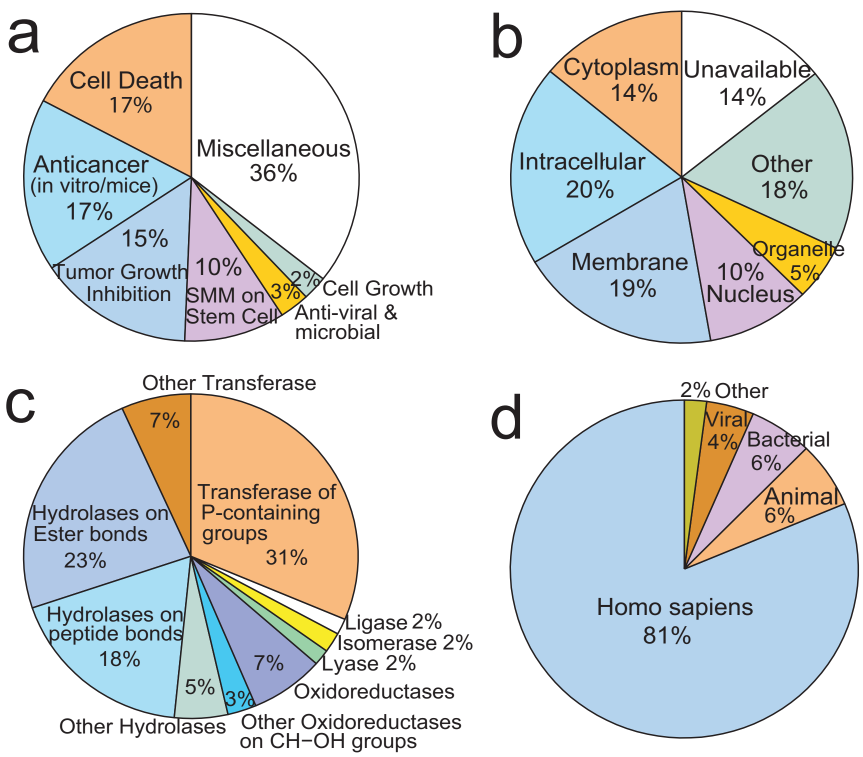

Figure 4.3: Distributions of the BioAssay data. (a) Distribution of the purposes of the 626 cell-based bioassays. (b) Distribution of the target protein cellular components of the 680 target-based bioassays. (c) Distribution of the organisms from which bioassay target proteins were taken. (d) Distribution of the function classes the 113 bioassay target emzymes.

4.3.3 Generating and Characterizing the BioAssay Network

The complexity of the BioAssay data described above showed that deep investigation will be difficult without organizing and visualizing the data in a rational matter. It is natural to generate a bipartite network for bioassays and compounds in PubChem. To visualize the BioAssay data with reasonable complexity, we can generate a BioAssay network projection as a complementary, bioassay-centered view of the bioassay space, where bioassays are represented as nodes and two bioassays are connected if they have similar binding profiles. We used *Jaccard coefficient* (the fraction of active compounds shared by two bioassays in the total number of distinct active compounds of them) to measure the similarity of bioassay binding profiles. By connecting any two bioassays that shared at least 10% active compounds, we generated a network of bioassays with 899 nodes and 6,080 edges as shown in Fig. 4.4a. Cell-based bioassays were represented by circles and colored according their screening purposes, while target-based bioassays were represented by rectangles and colored by their cellular components from the Gene Ontology database.

In Fig. 4.4a, on average a node were connected via about 6 links and had about 13.5 neighbors (the mean shortest path length = 6.09). A large number of clusters with 2-3 bioassays were observed in Fig. 4.4a. They were usually primary, confirmatory, or summary screens for the same bioassay, and certainly most of their active compounds were identical. Besides many medium-sized clusters with 4-20 nodes, there were two giant clusters composed of most bioassays, within each of which there was an extremely dense component. The two clusters corresponded to the NCI tumor cell line growth inhibition assays and the NCGC cell death related (cytotoxicity, viability, and etc.) assays. A clear trend was observed that in most cases bioassay data in the same cluster were submitted by the same organization for similar purposes or against similar target proteins. In addition, We found that there did exist a few clusters that contain both target-based and cell-based bioassays, while most clusters in the network have nodes of the same bioassay type (either target-based or cell-based). According to the definition of an edge in this network, such heterogeneous clusters revealed that the binding profiles of some target-based and some cell-based bioassays are to some extent similar (10% similar in this case), which can be helpful on understanding the protein-compound interactions within the cell-based bioassays and possible identify critical proteins in the cell-based bioassays.

We calculated the bioassay degree distribution *P(k)*, measuring the probability of a given bioassay connects with other *k* bioassays (Fig. 4.4b), and found that *P(k)* decreased gradually as the degree *k* increased. A power-law fitting with correlation coefficient as 0.955 and R^2 = 0.527 revealed that the exponent = -1.009. Hence, this is a typical scale-free network according to the definition by Barabasi *et al.* [Barabasi & Albert, 1999], in which a small fraction of nodes have most of the linked connected, and the majority of nodes have only a few links, as observed in Fig. 4.4a. In addition, the distribution of the average clustering coefficient (Fig. 4.4c) was found approximately independent on the node degree, and slightly fluctuated around the mean 0.78 (standard deviation 0.14). This answered that the BioAssay network was scale-free, but not a hierarchical

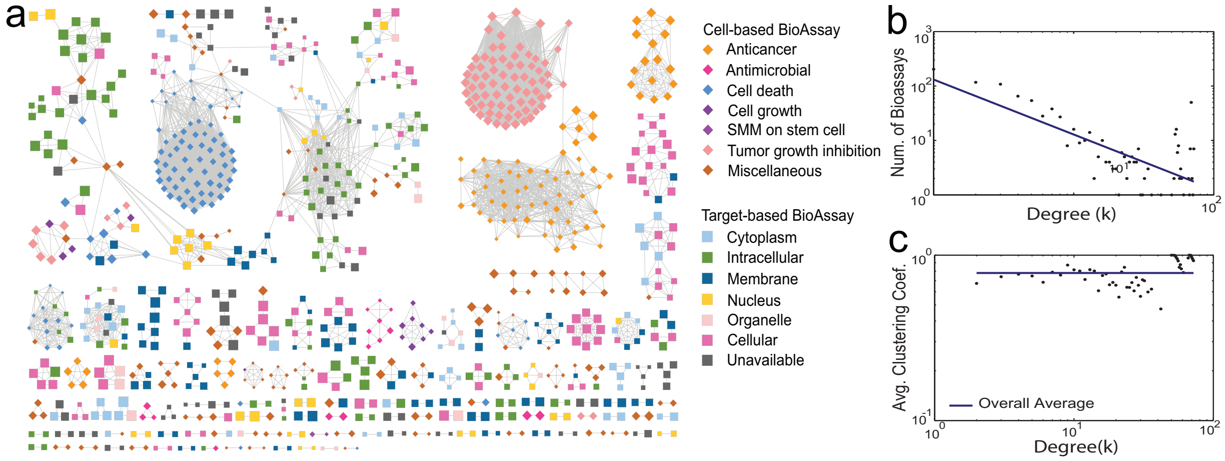

Figure 4.4: The BioAssay networks and topological distributions. (a) The BioAssay network in which nodes are bioassays and two bioassays are connected if they share at least 10% active compounds. The size of each node is proportional to the number of its active compounds, and the coloring of nodes is similar to Figure 4.4a. (b) Distribution of the network node degrees. The power-law fitting clearly shows that is a typical scale-free network. (c) Distribution of the average clustering coefficients. The almost constant fitting shows the BioAssay network is not hierarchical, not as other biological networks.

network [Ravasz et al., 2002], although typical biological networks were usually both scale-free and hierarchical. One reason could be that the edges in the bioassay network have no clear biological meaning.

4.3.4 Some Conclusions

In this work, we integrated the bioassay data from PubChem and other biological databases such as DrugBank and UniHI, and systematically analyzed them using biological network analysis approaches. We first calculated a variety of distributions of the bioassay data to identify some important trends in bioassay screenings and target selections. We then built a BioAssay network, and network topology analysis demonstrated that this network is a scale-free network but is not hierarchical, which is different from typical biological networks that are usually both scale-free and hierarchical. Some cell-based bioassays share a large portion of active compounds with target-based bioassays, which is helpful to determine the interacting proteins in the cells. By integrating other information related to adverse drug reactions, network analysis approaches provide valuable tools for further understanding the mechanisms of many ADRs systematically.

Chapter 5

Novel Network Features for Predicting Potential Drug Targets

A significant number of drug failures were attributed to the utilization of inappropriate drug targets at the early preclinical stages [Butcher, 2003]. Although great efforts have been exerted on drug research and development, only a limited number of drug targets have been identified. The majority of drug targets came from a few gene families, e.g. about 60% current drug targets are membrane proteins [Yildirim et al., 2007], while in human genome only 15$\sim$39% genes were predicted to contain transmembrane segments [Ahram et al., 2006]. To this end, the drug target space has not been fully explored, especially in the regions of other gene families. It is highly desirable for the pharmaceutical industry to use *in silico* approaches to help identify correct drug targets in the interaction network space.

5.1 Introduction

Genes in human genome have been classified into two classes: the *druggable* genome (genes that express proteins binding to drug-like molecules with potency greater than a threshold [Hajduk et al., 2005, Hopkins & Groom, 2002], e.g. 10 μM) and the remaining *undruggable* genome. According to the estimation of Hopkins *et al.* [Hopkins & Groom, 2002], there are approximately

10% human genes that can potentially become drug targets. However, the boundary between the *druggable* and *undruggable* genome is ambiguous and dynamic, and highly depends on the screening libraries. Identifying novel drug targets, especially from the currently considered *undruggable* genome, could be a promising solution to the current dilemma in drug discovery.

It is well known that proteins rarely function in isolation inside or outside cells, but rather behave as part of highly interconnected cellular networks [Chaurasia et al., 2007,Rual, 2005,Stelzl & *et al.*, 2005]. It will be advantageous to investigate proteins in the context of human protein-protein interaction (PPI) networks, with the hypothesis that topological environment of drug targets are distinct from that of non-drug-target proteins. For instance, Yildirim *et al.* [Yildirim et al., 2007] constructed a drug-target network and found that most known drug targets formed a giant cluster in the human PPI network. They concluded that drug targets were usually not essential genes, but they are close to essential genes and disease genes in the network. Therefore, known drug targets, disease genes, and essential genes are special groups of proteins in the human PPI network. It is the motivation for us to propose network topological features of proteins regarding to their relationships to these special proteins.

Machine learning algorithms have been widely used in pharmaceutical and bioinformatics studies. By integrating novel network and/or sequence features with advanced machine learning techniques, we proposed a framework to build highly accurate models to predict if a protein is likely to be a potential drug targets or not. By analyzing the relevance of proposed features, we can prioritize a set of proteins according to their predicted druggability, and thus design high-throughput screening to verify them more efficiently.

5.2 Related Work

Various computational methods have been proposed and applied for identifying potential drug targets. Zheng *et al.* [Zheng et al., 2006] reported properties of new druggable proteins based on analysis of approved drug targets, including membership of a target family, involvement in no more than two pathways, presence in no more than two tissues, and etc. In addition, Hajduk *et al.* [Hajduk et al., 2005] used the 3D structural information to predict whether a particular protein can bind with small, drug-like compounds. Although these methods achieved reasonable performance, they suffer from either poor generalization capability or limited availability of data such as protein 3D structures.

Supervised learning has also been widely used for drug target identification. For instance, Li *et al.* [Li & Lai, 2007] predicted potential drug targets based on simple sequence properties such as hydrophobicity, polarity, and etc., and achieved about 80% accuracy using SVM cross validation on the 186 selected drug targets. In addition, Bakheet *et al.* [Bakheet & Doig, 2009] proposed a comprehensive list of properties of human drug target proteins, including EC numbers, Gene Ontology terms, Glycosylation, and etc., analyzed their correlation to drug targets, and also used them as features to predict potential drug targets. Recently, Zhu *et al.* [Zhu et al., 2009] used five topological features extracted from human PPI networks to identify potential drug targets, and proposed a measure to rank proteins in the PPI network according to their potential of being drug targets.

If a protein is modulated by external stimulus, it is highly likely that its interacting partners, even the whole module, are also subject to the perturbation. Given a sufficiently complete network of high-quality PPI annotations, drug targets can be distinguished from non-drug-target proteins due to their distinct response to network perturbations, which is our basic assumption to use solely network topological features to predict potential drug targets. By investigating a set of 15 network features related to known drug targets, disease genes, and essential genes, we formalized the prediction of potential drug targets as a typical supervised learning problem. To this end, we applied sophisticated machine learning algorithms to analyze these network features and build accurate models to predict potential drug targets.

5.3 Human Protein Interaction Network

To extract network topological features for proteins, we need a human protein-protein interaction (PPI) network with as accurate and complete as possible interaction annotations. UniHI (`http://www.unihi.org`) is a unified human PPI network containing over 250,000 human PPIs collected from 14 major PPI sources with careful data integration and literature curation [Chaurasia et al., 2007, Chaurasia et al., 2009]. It also provides quality scoring systems for each data source. After careful curation, we obtained a human PPI network with 13,602 proteins as nodes and 157,349 PPIs as edges by removing redundant nodes and/or edges, merging duplicated nodes and/or edges, and excluding non-human proteins and other noises.

Approved Drug Targets: We obtained the gene symbols of the target proteins of all approved drugs from the DrugBank database [Wishart et al., 2008, Wishart et al., 2006], and use official gene symbols to cross-reference proteins in the UniHI network and in DrugBank. We mapped the genes symbols of drug targets in DrugBank to the curated UniHI network, and excluded any drug targets that have some network topological features unavailable, resulting in a set of 1,092 drug targets for positive training samples and network feature calculation.

Human Disease Genes: We downloaded all "Genes Associated with Diseases" from the GeneCards database (`http://www.genecards.org/`), and used gene symbol mapping to identify 1,521 proteins as disease genes in UniHI. In addition, we removed any disease genes that have been approved drugs, and obtained a final set of 1,157 disease genes. They will be used to calculate important topological features for proteins in the UniHI network.

Human Essential Genes: A human gene was defined as "essential" if a knockout of its mouse ortholog confers lethality. To find human essential genes, we first extracted mouse essential genes from the Mouse Genome Informatics Database [Eppig et al., 2005], and obtained 2,564 human essential genes through the human-mouse ortholog associations. Using gene symbol mapping we obtained 2,059 essential genes in the UniHI network, and finally used 1,759 of them after removing those that either have been used as drug targets or disease genes, or have some topological features unavailable.

Table 5.1: The 15 topological features extracted from the human PPI network.

Feature(Count)	Formula	Description
Degree (1)	k_i	Number of direct links to node *i*
Clustering coefficient (1)	$2\mathrm{n}_i/\mathrm{k}_i(\mathrm{k}_i+1)$	n_i is the number of links among the k_i neighbors of node *i*
Topological coefficient (1)	$\sum_j J(i,j)/k_i$	See text
Minimal SPL (3)	$min_j(d_{ij})$	Minimal SPL to drug targets, disease and essential genes
Mean SPL (3)	$\sum_j d_{ij}/\vert P\vert$	Average SPL to drug targets, disease and essential genes
Fraction of neighbors (3)	k_i^p/k_i	Fraction of node *i*'s neighbors as drug targets, disease or essential genes
Characteristic distance (3)	See text	Measure of clustering of drug targets, disease and essential genes

Putative Non-drug-target Proteins: It is indispensable to have negative samples to build an accurate model, that is, we need some proteins that can be surely determined as not drug targets. This is technically difficult since no researcher is interested in validating that a protein is definitely not a drug target. To solve this dilemma, we simply excluded any proteins that have been used as drug targets, disease genes, and essential genes from consideration [Bakheet & Doig, 2009, Li & Lai, 2007, Zhu et al., 2009]. In addition, any proteins that have some topological features unavailable were also removed, resulting in a set of 9,674 proteins. In all experiments, we randomly selected a number of proteins from this set as our negative samples. It is sure that there will exist some false negatives, however, with random sampling the error rate is acceptable (<5%) considering that only less than 10% random proteins could be drug targets [Hopkins & Groom, 2002].

5.3.1 Network Topological Features

A network is an undirected acyclic graph consisting of a number of nodes and edges. A node can represent any object, and an edge connects two nodes and usually carries some physical meaning such as interaction, similarity, and etc. In this work, we proposed 15 topological features extracted from human PPI networks, including three general topological features: degree, clustering coefficient, and topological coefficient, as summarized in Table 5.1.

The *degree* (DEG) of a node is the number of edges connecting it to other nodes. The *clustering coefficient* [Barabasi & Oltvai, 2004] of a node is defined as $\boldsymbol{C}_i = 2\boldsymbol{n}/(\boldsymbol{k}_i*(\boldsymbol{k}_i\text{-}1))$, where *n* denotes the number of direct neighbors of a given node *i*, and $\boldsymbol{k}_i$ is the number of links among the *n* neighbors of node *i*. If the clustering coefficient (CLU) of a node equals 1.0, the node is at the center of a fully connected cluster called a clique. If the clustering coefficient is close to 0, the node is in a loosely connected region. We can calculate average clustering coefficient over nodes with the same degree, and then obtain the distribution of average clustering coefficient over node degrees. The average of $\boldsymbol{C_i}$ over all nodes of a network assesses network modularity.

The *topological coefficient* (TPG) [Stelzl & *et al.*, 2005] $\boldsymbol{T_i}$ of a node *i* with $\boldsymbol{k_i}$ neighbors is computed as $\boldsymbol{T_i} = \sum_m \boldsymbol{J}(i,m)/\boldsymbol{k_i}$, where ***J(i,m)*** is defined for all nodes *m* that share at least one neighbor with node *i*. The value ***J(i,m)*** is the number of neighbors shared between the nodes *i* and *m*, plus one if there is a direct link between *i* and *m*. The topological coefficient is a relative measure for the extent to which a node shares neighbors with other nodes. Nodes that have one or no neighbors are assigned a topological coefficient of 0. Topological coefficients can be used to estimate the tendency of the nodes in the network to have shared neighbors.

5.3.1.1 Shortest Path Length-related Features

Next we defined six network features that are related to shortest path lengths (SPLs) between proteins and three pharmaceutically important sets of proteins/genes: approved drug targets, human disease genes, and essential genes. The shortest path length (SPL) between two nodes in a network is defined as the minimal number of consecutive edges between them. Given a protein in the UniHI network, the first three features are computed as its minimal SPLs to the nearest drug target, disease gene, and essential gene, not including the protein itself. These features evaluate the minimal distance between a protein and pharmaceutically important proteins. In addition, the remaining three features are the mean SPLs between a protein and all drug targets, disease genes, and essential genes in the UniHI network. These features evaluate the overall average distance between a protein and those important special proteins.

5.3.1.2 Characteristic Distance Features

The final six features are related to the clustering between proteins and the three pharmaceutically interesting sets of proteins. If we use a visual graph to view how the proteins distribute in the

UniHI network, we can find the aggregation between proteins and drug targets (disease genes, and essential gens) are different from one to another. So the first three features will be defined as the fraction of approved drug targets, disease genes, and essential gens in the direct neighbors of a given protein. For instance, if a protein has 15 direct neighbors (its degree = 15), 3 of them are known drug targets, 2 of them are disease genes, and 0 of them are essential genes, these three features will be calculated as 0.2 (3/15), 0.133(2/15), and 0.0 (0/15), respectively. These features provide a simple measure how the pharmaceutically important proteins are clustering around a protein.

To gain more understanding on the probability of proteins being potential drug targets, we developed a model to quantify the clustering of drug targets, disease genes, and essential genes surrounding other proteins. By computing the fraction $\boldsymbol{F_i}$ of drug targets, disease genes, and essential genes at each distance $\boldsymbol{d_i}$, we obtained the distribution of these three sets of proteins around each other protein in the network. We then defined the characteristic distance $\boldsymbol{D_c^p}$ as follows: $1/(\boldsymbol{D_c^p})^2 = \sum_{i=1}^{n} \boldsymbol{F_i/d_i^2}$, where n is the diameter of the network and p represents a protein from the three sets. The mechanism underlying this formula was from electrostatics and the Coulomb law. We can view each drug target (disease gene, and essential gene) as a "unit charge" that generated an electric field with field strength $\boldsymbol{F_i/d_i^2}$, and all these electric fields accumulated at the position of a given protein. Therefore, the last three features are computed as the characteristic distance between a protein and all approved drug targets, disease genes, and essential genes, respectively.

5.3.2 Classifiers

Classifiers played an important role by taking descriptor vectors and class labels as input and generate prediction models as output. There are many widely used classifiers, e.g. support vector machines (SVM), K-nearest neighbors (KNN), neural network, random forest, and etc. Each has their advantages and disadvantages. The following three classifiers are used in our study.

5.3.2.1 Support Vector Machines

Kernel classifiers, exemplified by the support vector machines (SVM), have gained popularity in many health related applications. Kernel classifiers utilize a specially designed function, called *kernel function*, to handle complex, non-linear relationships. Specifically, this is done by first mapping the original data to a new feature space and then deriving a liner relationship in the feature space. Coupled with large margin separation principle as utilized in SVM, kernel classifiers have the following advantages: (i) low chance of over-fitting in a high dimensional feature space, (ii) no need to explicitly compute the coordinates of the data and hence are computational efficient and (iii) empirically give comparable (and usually better) results to competing methods such as Neural Networks. Here we used SVMs to build highly accurate model for the baseline of comparison using tenfold cross validation and bootstrapping. The LIBSVM [Chang & Lin, 2001] implementation and RBF kernels were used in all experiments.

As a supervised learning method widely used for classification and regression, support vector machines (SVM) [Burges, 1998] view input data as two sets of vectors in an n-dimensional space, each with different class labels. SVM constructs a separating hyperplane in that space by maximizing the margin between the two data sets. To calculate the margin, two additional parallel hyperplanes are constructed, one on each side of the separating hyperplane, and are “pushed up" against the two sets of data points respectively during the optimization process. Intuitively, a good separation is achieved by the two parallel hyperplanes with the largest distance to each other.

We downloaded the state-of-the-art implementation named LIBSVM [Chang & Lin, 2001] of the SVM classifier. Signature, DRAGON, and FFSM descriptors take chemical compounds in SDF format as input, convert them into *n*-dimensional descriptor vectors. Then LIBSVM treats each *n*-dimensional vector as a point in n-dimensional space, and build a decision boundary between actives and inactive samples in that space.

5.3.2.2 Logistic Regression

Logistic regression is a generalized linear model for pattern recognition. Such models include a linear exponential part followed by a “link function”. First the linear function of input features is calculated and run through the logistic link function. The optimal coefficients of the linear function are learned from training data. Comparing with linear regression, logistic regression can be used to construct a model which estimates probabilities, e.g. for medical diagnosis and credit scoring. With proper regularization, the coefficients of a logistic regression model can be used to evaluate the relative significance of each feature. Here we used a ℓ_1-regularized logistic regression (LLR) algorithm proposed by Boyd *et al.* [Koh et al., 2007] to run our experiments.

5.3.2.3 k-Nearest Neighbors

The k-nearest neighbors (kNN) algorithm is a simple classification method based on the votes of the k nearest neighbors of a given data point in the feature space. Given a training data set with class labels, the parawise distance (or similarity) between a testing data point and each training sample will be calculated and sorted, and the top k closet (or most similar) training samples are picked, and the majority of the k class labels will be assigned to the given testing data point. This method can be used to identify the most similar known drug target to a protein that was predicted as a potential drug target when k = 1.

5.4 Results

Based on the descriptions of the 15 network features proposed for each protein, they were computed for 1,092 known drug targets extracted from DrugBank, and 9,674 putative non-drug-target proteins in the UniHI database. The 1,092 known drug targets played two roles in the experiments: some known drug targets were used to calculate the four related network features, and the others were served as training positive samples in cross validation.

5.4.1 Results Using No Drug-target-related Features

Due to the trick usage of known drug targets in feature extraction and cross validation, we first excluded the four drug-target-related features from consideration, and hence built the baseline for performance comparison using the remaining 11 features. SVMs with RBF kernels, L1-regularized logistic regression, and kNN algorithms were used to classify proteins into two classes: drug targets or not drug targets. With a highly unbalanced data set consisting of 1,092 positive and 9,674 negative samples, we randomly selected 500 positive and 1000 negative samples to make a balanced subset. We used 80% positive and 80% negative samples for training, and the rest 20% for testing. Tenfold cross validation was applied the training set to select the best model parameters ($\boldsymbol{C}$ and γ for SVMs, λ for L1-regularized logistic regression, and $\boldsymbol{k}$ for kNN).

After the optimal model parameters were identified, a single model was built on the training set and tested on the testing set to obtain the generalization accuracy as TP+TN/N, where TP = true positive, TN = true negative, and N is the total number of testing samples. We repeated the whole experimental process for 100 times to obtain stable classification results. The accuracies and optimal parameters for each classification method are summarized in Table 5.2, which showed that up to 75% accuracy was achieved by L1-regularized logistic regression using only 11 features.

Table 5.2: Summary of baseline results from experiments using 11 features.

Algorithm	Model Parameters	Testing Accuracy
SVMs	C = 4.539+/-1.15 γ = 2.406+/-0.804	0.691+/-0.030
ℓ_1-regularized Logistic Regression	λ = 0.0006+/-0.0011	0.757+/-0.028
k-Nearest Neighbors	k = 3	0.694+/-0.017

In addition, the best models obtained from L1-regularized logistic regression provide the coefficient of each feature in the linear function. From 100 cross-validation experiments, we obtained 100 sets of coefficients from the best models. According to Hastie *et al.* [Hastie et al., 2009], the Z-score is defined as the mean coefficient divided by its standard error for each of the 11 features. a Z-score whose absolute value is greater than 2.0 means the corresponding feature is significant to the model (confidence level 95%) [Hastie et al., 2009]. From Table 5.3 we found that the feature *degree*, *topological coefficient*, and *fraction of neighboring essential genes* were not significant.

Clustering coefficient, *minimal and average SPLs to disease genes*, *fraction of neighboring disease genes*, and *characteristic distance to disease genes* are features negatively correlated to the druggability of proteins, especially for minimal and average SPLs to essential genes. Meanwhile, average SPLs and characteristic distance to essential genes are positively correlated features.

Table 5.3: Z-scores of ℓ_1-regularized logistics regression model coefficients.

Network Feature	Z-score
Degree	1.111
Clustering coefficient	-3.282
Topological coefficient	-0.272
Minimal SPLs to the nearest disease gene	-3.018
Minimal SPLs to the nearest essential gene	-8.007
Average SPLs to the nearest disease gene	-3.090
Average SPLs to the nearest essential gene	3.450
Fraction of neighboring disease genes	-4.871
Fraction of neighboring essential genes	-1.028
Characteristic distance to disease genes	-3.854
Characteristic distance to essential genes	2.552

5.4.2 Results Using All Network Features

Feature Extraction: When we take the four drug-target-related features into consideration, we split the 1,092 known drug targets into two sets: one for feature extraction, and the other using for positive training samples. We randomly selected 50% known drug targets for calculating the 15 network features described in section 3.2 for the remaining 50% drug targets (546 positive samples) and all the putative non-drug-target proteins (9,674 negative samples).

Data Set Creation: Given the highly unbalanced data set with much more negative samples than positive samples, we created a balanced data set by randomly selecting 500 positive samples and 1,000 (the number of negative samples are always twice as many as positive samples). For each such data set, we randomly selected 80% positive samples and 80% negative samples as a training set, and the rest 20% data as a testing set.

Model Construction and Selection: Tenfold cross validation was applied to the training set to select the best models. We used grid search to find the best combination of model parameters, e.g.,$\boldsymbol{C}$ and γ for SVM RBF kernels, λ for L1-regularized logistic regression, and $\boldsymbol{k}$ for kNN.

Model Assessment: After we selected the best models, we used all the training samples to construct a single final model using the identified optimal parameters, and applied this model to the testing samples to calculate the testing (generalization) accuracy. We repeated the whole experimental process for 100 times using random sampling to achieve stable prediction accuracy, and reported the mean and standard deviation of testing accuracy in Table 5.4. The second row in Table 5.4 demonstrated that the best accuracy obtained using SVMs, L1-regularized LR, and kNN (k=5) is 0.774, 0.782, and 0.733, respectively.

We also repeated the experimental procedure as described above using different percentages of known drug targets for feature extraction, such as 30%, 70%, 80%, and 90%. Note that the number of positives selected in step 2 were changed accordingly since the total numbers of positive samples available were different. All resulting accuracies and standard deviations obtained using SVMs, LLR, and kNN are summarized in Table 5.4.

Table 5.4: Summary of mean accuracy (standard deviation) using different fractions of known drug targets for feature extraction.

% Drug Targets	SVMs	LLR	kNN (k=5)
30	0.770 (0.015)	0.784 (0.014)	0.736 (0.016)
50	0.774 (0.018)	0.782 (0.019)	0.733 (0.021)
70	0.762 (0.024)	0.787 (0.025)	0.727 (0.025)
80	0.752 (0.030)	0.792 (0.030)	0.720 (0.031)
90	0.732 (0.041)	0.766 (0.041)	0.696 (0.041)

To visualize the trend in the results more clearly and compare them with baseline results, we plotted the results using dot line in Fig. 5.1 with different colors, and the baseline results are represented with red horizontal lines. Overall, by introducing the four drug-target-related features, the cross validation accuracy obtained using SVMs, LLR, and kNN increased by 8.4%, 3.5%, and 4.2%, respectively. With up to 80% accuracy, the best models from L1-regularized logistic regression are considered meaningful for predicting potential drug targets.

Fig. 5.1 demonstrates that as the fraction of known drug targets used for feature extraction increases, the accuracy first slightly fluctuates (increases for SVMs and LLR, and decreases for

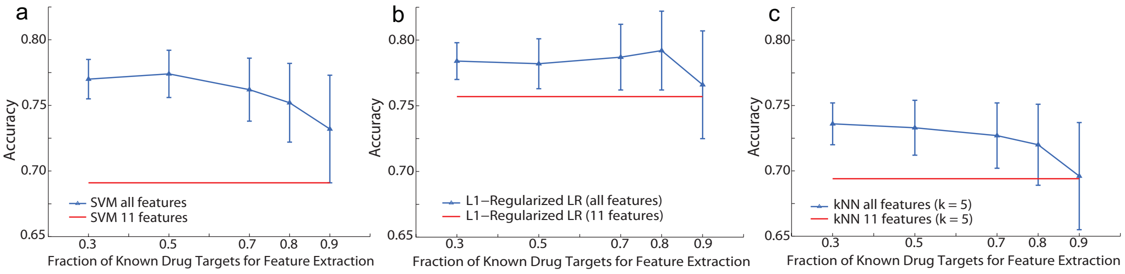

Figure 5.1: Summary of cross-validation accuracy using SVM, logistic regression, and kNN.

kNN), and then decreases significantly. The main reason for this pattern is that less training data were used to learn models when more fraction of drug targets was used for feature extraction. In addition, the standard deviation bars enlarged dramatically when the fraction of drug targets for feature extraction increased, because less training samples were used and 10 random samplings were not enough to lower the variance. There is a tradeoff between the number of training samples and the number of drug targets used for feature extraction, so the optimal partition is about 40%-60%.

5.4.3 Comparison with Previous Work

Zhu *et al.* [Zhu et al., 2009] proposed a SVM classification method to predict potential drug targets using five network topological features: degree, clustering coefficient, 1N index, shortest distance to drug targets, and average distance to drug targets, which are included in our 15 feature set. To compare the performance of our feature set with Zhu *et al.* [Zhu et al., 2009], we applied a similar cross-validation experimental procedure: a) Partitioned the set of drug targets into two parts: one for feature extraction, and the other used for training; b) Randomly selected 30%, 50%, 70%, 80% and 90% of drug targets for feature extraction, and repeated the random sampling for 10 times; c) Randomly selected a balanced data set consisting of twice as many negative as positive samples, and repeated the random sampling for 10 times. We used grid search to select the best parameters C and γ for RBF kernels.

The results for our method and Zhu *et al.*'s method [Zhu et al., 2009] were listed in Table 5.5

Table 5.5: Accuracy (standard deviation) of our method and the method by Zhu *et al.*.

% Drug Targets for	SVMs		LLR		kNN	(k= 5)
Feature Extraction	Our method	Zhu *et al.*	Our method	Zhu *et al.*	Our method	Zhu *et al.*
30	0.794(0.015)	0.695(0.017)	0.795(0.014)	0.722(0.016)	0.758(0.017)	0.677(0.021)
50	0.789(0.018)	0.696(0.024)	0.790(0.018)	0.729(0.019)	0.752(0.016)	0.664(0.023)
70	0.794(0.022)	0.691(0.028)	0.797(0.019)	0.736(0.026)	0.774(0.020)	0.665(0.033)
80	0.796(0.031)	0.680(0.034)	0.800(0.026)	0.747(0.028)	0.772(0.028)	0.667(0.040)
90	0.770(0.041)	0.666(0.039)	0.786(0.041)	0.723(0.039)	0.754(0.041)	0.654(0.050)

for comparison. For each partitioning of drug targets, our best five features (characteristic distance to known drug targets and to disease genes, minimal and mean SPL to known drug targets, and topological coefficient) outperformed the method by Zhu *et al.* for 9.3-11.6% using SVMs, 5.3-7.3% using L1-regularized logistic regression, and 8.8-10.9% using kNN at k = 5. A simple explanation of the superiority of our feature set was that our method systematically not only integrated information from both drug targets and disease genes, but also integrated information from many drug targets and/or disease genes into one feature. Our experimental results demonstrated that disease gene information did help identify potential drug targets.

In addition, we also compared our feature set with the protein sequence features used by Li *et al.* [Li & Lai, 2007]. We downloaded all protein sequences from UniProt (http://www.uniprot.org), mapped all human proteins onto the UniHI network, and obtained 1,075 drug target sequences, and 7,099 putative non-drug-target sequences. We then applied Needleman-Wunsch global alignment algorithm [Needleman & Wunsch, 1970] to calculate the pairwise sequence identities for each of the sequence sets, and iteratively removed protein sequences to cull all sequences in each set with a given identity threshold (e.g. 30% in this work). Eventually we achieved 660 drug target sequences and 5,006 non-drug-target sequences, and in each set no pairwise sequence identity are greater than 30%. We then calculated exactly the same 146 protein sequence features as in Li *et al.* [Li & Lai, 2007] using the online server PROFEAT by Chen *et al.* [Li et al., 2006].

In the experiments, both methods only used the 660 positive and 5,006 negative samples to conduct cross validation at different sizes of training set. We assured that the numbers of training samples for both methods are very close to each other. At each size of training set, we randomly selected 50 data sets with approximately twice as many negative as positive samples, and the final

Table 5.6: Accuracy (standard deviation) of our method and the method by Li *et al.*.

% Drug Targets for	SVMs		LLR		kNN	(k= 5)
Feature Extraction	Our method	Li *et al.*	Our method	Li *et al.*	Our method	Li *et al.*
30	0.754(0.022)	0.710(0.042)	0.775(0.022)	0.692(0.033)	0.723(0.020)	0.651(0.037)
50	0.753(0.025)	0.700(0.032)	0.771(0.026)	0.699(0.019)	0.713(0.023)	0.660(0.024)
70	0.731(0.031)	0.696(0.013)	0.765(0.030)	0.691(0.012)	0.703(0.030)	0.649(0.016)
80	0.737(0.041)	0.688(0.009)	0.794(0.036)	0.679(0.012)	0.710(0.037)	0.645(0.009)
90	0.732(0.046)	0.669(0.015)	0.739(0.055)	0.666(0.008)	0.689(0.050)	0.622(0.017)

accuracy and standard deviation were obtained by averaging results over the 50 experiment, as shown in Table 5.6. Clearly our feature set outperformed the 146 sequence features for 3.5-6.3% using SVMs, 7.4-11.5% using L1-regularized logistic regression, and 5.3-7.2% using kNN at k=5. The best accuracy was 79.4% using our method, but 71% using the sequence features.

5.4.4 Relevance of Network Features

Our L1-regularized logistic regression models provided not only classification accuracy, but also the coefficients of all involving features in the models. At each partitioning we repeated the experiments for 100 times, and in each experiment we calculated the Z-scores for each network feature as the mean coefficients divided by their standard errors as described in Hastie *et al.* [Hastie et al., 2009]. High Z-scores that are beyond the region [-2.0 2.0] indicate significant features. The Z-scores for three network features: degree, clustering coefficient, and topological coefficient, were shown in Fig. 5.2. We found that the Z-scores topological coefficient are within the range [-8 -2], and thus this network feature shows significantly negative correlation to prediction results. The Z-scores of feature *degree* and *clustering coefficient* are within [-2.0, 2.0], and hence have only marginal significance.

The remaining 12 features naturally fall into four groups, each with three features that are related to drug targets, disease genes, and essential genes, respectively. The mean Z-score and standard deviation bar of each feature was plotted in Fig. 5.3. We discovered some interesting patterns. First, the characteristic distance to known drug targets (Fig. 5.3a) was found the most significant predictor since its Z-scores are very negative in the range [-14,-4], showing that proteins with

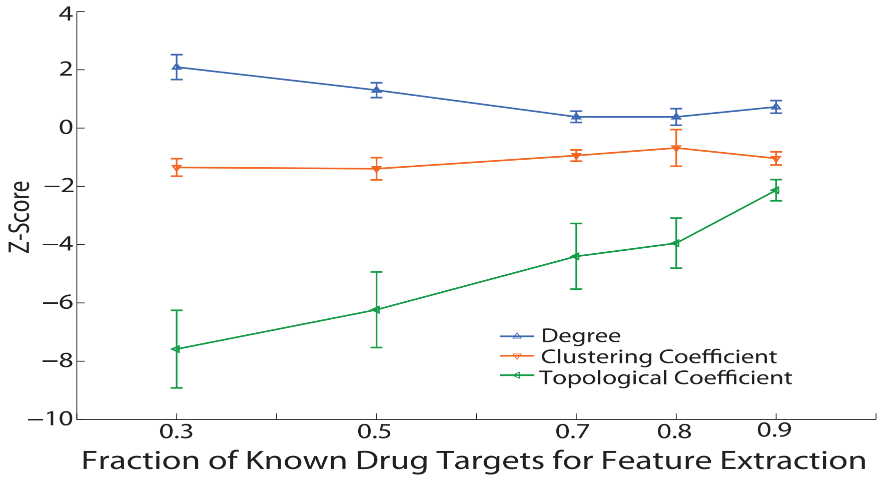

Figure 5.2: Z-scores of network features: degree, clustering coefficient, and topological coefficient.

shorter characteristic distance to known drug targets have higher chance to be drug targets, which is intuitive because shorter characteristic distance means topologically more similar. In addition, characteristic distance to disease genes were also observed significantly negatively correlated to prediction results, though characteristic distance to essential genes was unsurprisingly marginally significant predictor.

Moreover, the minimal and mean SPLs (Fig. 5.3b and c) to known drug targets are both highly significant predictors, but with opposite inferences: a protein with greater minimal SPL and shorter mean SPL to known drug targets have higher probability to be potential drug targets. Observations on disease genes are quite similar, but SPLs to disease genes have only marginal significance. For essential genes, the significance is also very marginal, although they tell that a protein with shorter minimal SPL and greater mean SPL to essential genes has higher chance to be potential drug targets.

Finally, fractions of neighboring drug targets, diseases and essential genes are three weak predictors since their Z-scores are all within [-1.0,1.0] (Fig. 5.3d). It is somehow intuitive since considering only direct neighbors in the PPI network is superficial. It is noticeable that when 30% known drug targets were used for feature extraction, the Z-scores of most network features were more significant than other values. The reason is that when small number of known drug tar-

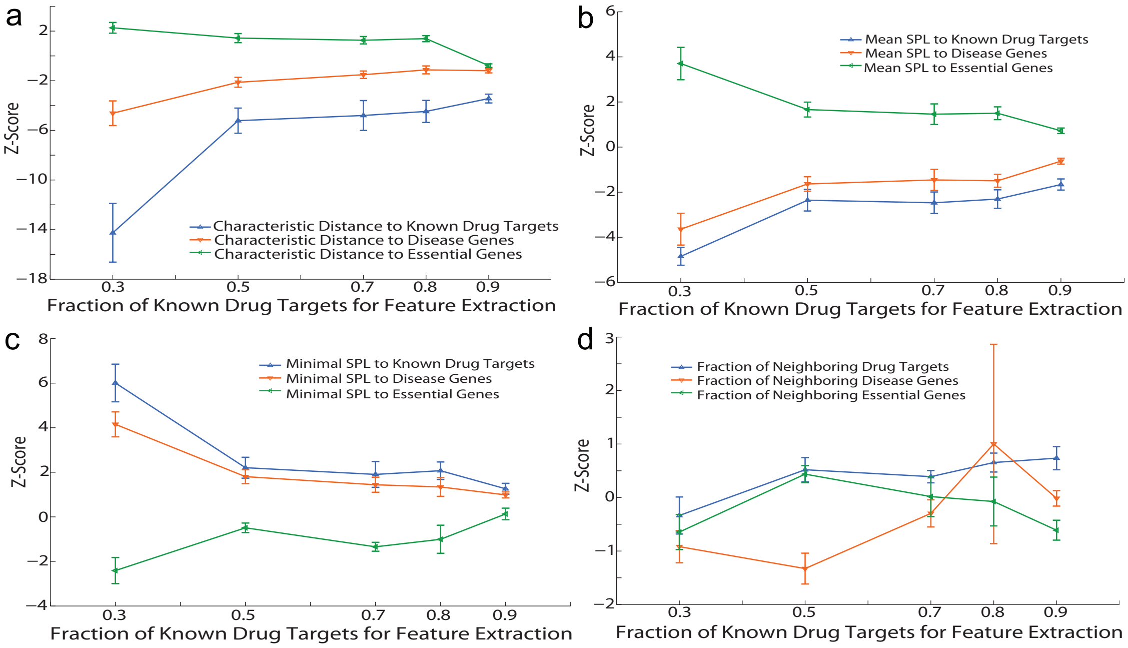

Figure 5.3: Summary of the Z-scores of four sets of topological features: (a) characteristic distance, (b) Mean SPL, (c) Minimal SPL, and (d)Fraction of neighboring special proteins. Each panel contains three features related to known drug targets, disease genes, and essential genes, respectively.

gets were used for feature extraction, the advantages of the four drug-target-related features were weakened.

5.4.5 Combining Network & Sequence Features

Now we have two sets of features for identifying potential drug targets: one extracted from protein interaction networks, and the other extracted from protein sequences, so it is natural and straightforward to integrate them together for better prediction performance. Our experimental procedures were very similar to experiments using only network features, except that fewer positive and negative samples were available. Given 660 positive and 5,006 negative samples, we still used 30%, 50%, 70%, 80%, and 90% known drug targets for calculating the four drug-target-related features, and calculated the 15 network features and 146 sequence features for the remaining positive and all negative samples. At each of the five partitioning thresholds, we randomly selected 90% positives for training and 10% for testing, with twice as many negatives as positives in each set. The other experimental steps were the same as described previously. The results of repeating the experiments for 100 times are summarized as in Table 5.7. As expected, the performance of combined features is better than sequence features, but worse than network features.

Table 5.7: Experimental results using both network features and sequence features.

% Drug Targets for	SVMs		LLR	
Feature Extraction	Combined Features	Feature Selection	Combined Features	Feature Selection
30	0.737+/-0.037	0.735+/-0.033	0.749+/-0.033	0.741+/-0.035
50	0.726+/-0.045	0.748+/-0.042	0.759+/-0.041	0.766+/-0.043
70	0.711+/-0.050	0.736+/-0.051	0.756+/-0.051	0.764+/-0.045
80	0.699+/-0.060	0.719+/-0.055	0.749+/-0.065	0.768+/-0.058
90	0.685+/-0.068	0.707+/-0.083	0.727+/-0.086	0.762+/-0.084

In addition, using Z-scores of the features calculated from L1-regularized logistic regression models, we selected four network features (degree, topological coefficient, characteristic distance to disease genes and to known drug targets) and five sequence features with Z-scores beyond the interval [-1.0, 1.0], which are considered at least marginally significant to the druggability of proteins. With the nine selected features, SVM classification performance improved slightly, which LLR performance didn't improve at all, mainly because LLR has performed feature selection with L1 regularization. The LLR prediction accuracy using all combined features can be up to 77%. Although it is slightly worse than the performance using only network features, it is much more stable. When more known drug targets are used for feature extraction, the results of using only network features fluctuated more dramatically than using combined features. Moreover, we also applied SVM recursive feature elimination (RFE) [Guyon et al., 2002] as feature selection on the combined features, but didn't achieve any improvement.

From our experiments, we concluded that the performance of network features highly depends on the completeness of the human protein interaction network and the number of known drug targets. Sometimes the interaction profiles of proteins are hard to obtain and thus are incomplete. For such proteins, our network features may not perform well, and sequence features can provide important complementary information for identifying potential drug targets. Although combined features didn't outperform network features, they provided more robust and stable results than the latter. As demonstrated in Table 5.7, when the percentage of known drug targets used for feature extraction increased, the results using combined features varied more gently than using only network features. Therefore, we here created a framework for identifying potential drug targets by integrating network features and sequence features efficiently. More sophisticated feature selection methods may also help improve the performance of the combined features.

5.5 Conclusions

We proposed a set of 15 topological features extracted from human PPI networks, applied sophisticated machine learning algorithms such as SVMs, logistic regression, and kNN to construct highly accurate models using these features to predict whether a human protein can be a drug targets or not, and achieved excellent performance with up to 80% prediction accuracy. In addition, we analyzed the correlation of each topological feature to the probability of being drug targets for human proteins by calculating the Z-scores of the model coefficients obtained by L1-regularized logistic regression, and found that some topological features were highly important to the druggability of a protein, such as characteristic distance to drug targets, shortest and average distance to drug targets and disease genes, and topological coefficients. Moreover, we compared the performance of our feature set with two previous work, and observed that our method outperformed them for 5-11% higher accuracy. Analysis demonstrated that the superiority of our features was originated from on highly integrated information from many drug targets and from both known drug targets and disease genes simultaneously. Our feature extraction only rely on the interacting profiles of proteins, and can by easily applied to many other applications. By integrating network features with sequence features, more robust and generalizable models can be built for identifying potential drug targets.

Chapter 6

Protein-chemical Interaction Prediction using Multi-task Learning

As discussed in earlier chapters, interactions between drug compounds and proteins or other cellular components are major causes of adverse drug reactions. Hence it is important to obtain the profile of interacting proteins for a given drug. However, the annotations between drug compounds and human proteins are very sparse, as shown in the data in PubChem. It is highly desirable to develop machine learning tools to predict the missing links between drugs and interacting proteins. In this chapter, we proposed two chemogenomics-based multi-task learning algorithms for protein-chemical interaction prediction.

6.1 Introduction

Interactions between proteins and small-molecule chemicals modulate many protein functions and biological processes, and identifying these interactions is a crucial step in modern drug discovery. Chemical biological assays are generally used to obtain protein-chemical interaction data, but their usage is largely restricted by the experimental expenses and time cost. *In silico* methods [Bleicher et al., 2003] provide important complementary tools for identifying active compounds for target proteins from a large database of chemical structures with low cost [Robert & Goodnow, 2006].

While the literature is rich that general machine learning algorithms (e.g. support vector machines, ℓ_1-regularized logistic regression) achieve decent performance on protein-chemical interaction prediction [Ning et al., 2009, Erhan & L'Heureux, 2006, Jacob & Vert, 2008], most of them are for a single protein with sufficient binding samples.

When only limited binding compounds are available for proteins, traditional single task learning (STL) methods that independently learn one single task at a time inevitably suffer from overfitting and are unable to build accurate models. Multi-task learning (MTL) [Caruana, 1997] is an excellent alternative approach that learns multiple related tasks jointly and transfers knowledge among them, which has been empirically as well as theoretically shown to generally improve performance of predictive models [Ando & Zhang, 2005, Argyriou et al., 2006]. In multi-task learning, the total training error of all tasks is minimized, the model parameters of all tasks are optimized jointly, and the generalization performance of models is improved.

There are other situations where MTL methods can help overcome the dilemma confronted by STL methods. Complex diseases such as Alzheimer's disease (AD) and cancers often associate with multiple target proteins. For instance, at least 43 identified proteins have been associated with AD [Stephenson et al., 2005]. Compounds that selectively inhibit only one of these disease-associated proteins are unable to effectively fight these diseases. Identifying promiscuous compounds that can inhibit multiple proteins associated with the same complex diseases is thus critical for future drug discovery.

In this chapter, we propose two complementary MTL algorithms for predicting active compounds for multiple related proteins, some of which may have very few binding samples. We hypothesize that the performance of protein-chemical interaction prediction can be highly improved by learning models for multiple related proteins jointly, integrating information from both proteins and chemicals, and selecting the most discriminative features for the final models. Moreover, our proposed MTL methods can be applied to identification of promiscuous compounds that interact with multiple related proteins. This typical multi-label classification [Tsoumakas & Katakis, 2007] problem is first split into multiple binary classification tasks with one label as a task, which are then learned jointly in the MTL framework [Ji & Ye, 2009, Wu et al., 2010].

In the first MTL method, we propose the Maximum-a-Posteriori (MAP) optimization method to integrate information from both proteins and chemicals in the same framework. Our main contribution is to exploit a covariance-coupled Gaussian prior for encoding protein features to regularize the logistic loss functions of compound features, though Gaussian prior has been used in some other learning problems [Chen & Rosenfeld, 2003, Bickel et al., 2008]. Here each task is to predict if a set of compounds interact with a given target protein. The characteristic of each task is simply represented by the features of the target protein, and the relatedness among different tasks is predefined by target protein similarity.

Generally features extracted from protein sequences and chemical structures have high dimensionality, and not all features own the same level of discriminativity. Selecting a limited number of highly discriminative features is thus critical for both improving performance and reducing computational cost. Here we adopt a boosted multi-task learning algorithm for face verification [Wang et al., 2009], and *repurpose* it for protein-chemical interaction prediction. This process is much like the drug repurposing in recent drug discovery, and significant modifications have been made to fit the problem of protein-chemical interaction prediction.

The second MTL method boosts chemical features with a number of independent boosting [Freund & Schapire, 1995, Mason et al., 2000] classifiers and learns the correlations between each task and each of these classifiers. Here base learners are two-leaf decision stumps of compound substructure features, and a boosting classifier is a weighted linear combination of these base learners. Multiple boosting classifiers and the relatedness among multiple tasks are learned jointly, and boosting and MTL are integrated into the same framework to take advantages from both techniques.

We compare our proposed MTL methods with traditional STL methods and existing MTL methods to demonstrate their performance improvement. We especially focus on the situation where some tasks have very few or even no training samples, and evaluate how our MTL methods perform and the effect of knowledge transfer among related tasks. We also investigate the performance of our methods on predicting the interactions between promiscuous compounds and their target proteins, which is pharmaceutically important for overcoming complex diseases in future drug discovery.

6.2 Related Work

MTL approaches have been widely studied in bioinformatics and chemoinformatics problems. Zhang *et al.* [Zhang et al., 2010] proposed a sparse multi-task regression (MTR) model for identifying common mechanism of responses to therapeutic targets by capturing the global structure of association using an intrinsic template shared among experimental conditions. Puniyani *et al.* [Puniyani et al., 2010] combined information from genetic markers across multiple populations to identify causal markers using $\ell_{1,2}$-regularized MTR for joint association analysis of multiple populations. [Bogojeska et al., 2010] presented an approach for the problem of predicting virological response to combination therapies by learning a separate logistic regression model for each therapy. [Xu et al., 2010] applied two MTL approaches for predicting protein subcellular localization for 20 organisms. All these existing methods deal with data from the same feature space, i.e., each sample contains features from only one type of objects.

Utilizing both protein features and compound features is a key (and to some extent ignored) issue in applying MTL for protein-chemical interaction prediction. Existing methods either totally ignored protein features, or simply integrated both sets of features using kernel tricks. Similarity between two pairs of proteins and compounds were expressed as either a weighted linear combination [Ning et al., 2009, Erhan & L'Heureux, 2006] or the tensor product [Jacob & Vert, 2008] of protein similarity and compound similarity. These data integration methods oversimplified the problem and achieved limited performance, and also might not well handle compounds with multiple interacting targets. These difficulties motivated us to develop novel MTL methods to handle these pharmaceutically important problems.

The rest of this chapter is organized as follows: in Section 2 we introduce how we collect data sets, extract features, and provide details of our proposed MTL methods. Our experimental results are presented in Section 3 with comprehensive comparisons to existing methods. In Section 4 we conclude our work with insights on potential applications of our proposed MTL methods.

6.3 Methods

In this section, we describe the two proposed MTL methods in detail. We discuss how we integrate compound and protein features into the same framework, characterize task relatedness, select the most discriminative features, and derive the problem formulation step by step.

6.3.1 Notation

Let $\mathbf{X} \in \mathbb{R}^{N \times D_1}$ denote the compound feature matrix, where N is the number of chemical compounds and D_1 is the dimensionality of compound features, and $\mathbf{P} \in \mathbb{R}^{M \times D_2}$ denote the protein feature matrix, where M is the number of proteins (tasks) and D_2 is the dimensionality of protein features. For protein-chemical interaction prediction, a data sample is a protein-compound pair and we have $N_{cp} \in [N, NM]$ data samples since each chemical interacts at least one protein and at most M proteins. $\mathbf{Y}$ denotes the $N_{cp} \times 1$ vector of sample labels. In binary classification as we study here, each element of $\mathbf{Y}$ is taken from {1,-1}, representing that the given pair of proteins and chemicals is active or inactive, respectively.

6.3.2 ℓ_2-regularized logistic regression

While Gaussian prior has been used as ℓ_2-regularization to provide prior knowledge in various machine learning applications [Chen & Rosenfeld, 2003, Bickel et al., 2008], in this work we encode protein features into the inverse of a covariance matrix (partial correlation matrix) in the prior, and the likelihood term is represented by logistic functions of compound features. Our main contribution is proposing this novel framework for composite data that each sample consists of two parties: one is encoded in the logistic loss term and the other is encoded in the Gaussian prior term.

Below we give details of a covariance-coupled ℓ_2-regularized logistic regression MTL method, labeled as *GaussMTL1*, which maximizes the posteriori with a Gaussian prior of model parameters with zero mean and constant variance. In the MTL framework, each task t is assigned a $D_1 \times 1$ vector $\mathbf{v}_t$ of logistic regression parameters, and the concatenated parameter vector $\mathbf{v}$ for M tasks is of size $MD_1 \times 1$. Similarly as existing ℓ_2-regularized logistic regression, our goal is to *maximize* the following log-posteriori function:

$$\sum_{\{x_i,y_i\}} \log[p(y_i|x_i,\mathbf{v})] - \lambda \mathbf{v}^\top \Sigma^{-1} \mathbf{v}, \tag{6.1}$$

where y_i is the binary label of the data sample pair x_i, Σ^{-1} is the inverse of a covariance matrix encoding protein features and is a function of matrix $\mathbf{P}$, and λ is the regularization parameter to be tuned. Σ is an $MD_1 \times MD_1$ matrix with 1's at the diagonal. It can be viewed as a $M \times M$ super-matrix whose elements are $D_1 \times D_1$ diagonal matrix blocks. The matrix block at position (i,j) describes the similarity kernel $k(t_i,t_j)$ between task t_i and t_j, and its diagonal elements are all set as $k(t_i,t_j)\rho$, where $\rho \in [0,1]$ is a scaling factor to control the magnitude of protein information to be considered in our models. All the remaining elements in Σ are 0's. Eventually Σ is a sparse matrix to define the task similarity using protein similarity kernels.

After plugging in the logistic functions of compound features, maximizing objective function 6.1) is equivalent to *minimize*

$$\sum_{i=1}^{N_{cp}} \log(1+\exp(-y_i \mathbf{x}_i^\top \mathbf{v})) + \lambda \mathbf{v}^\top \Sigma^{-1} \mathbf{v}, \tag{6.2}$$

where y_i is the label and $\mathbf{x}_i$ is an $MD_1 \times 1$ vector obtained by extending the compound feature vector (size $D_1 \times 1$) of sample $i, i = 1,2,...,N$. If the data sample i belongs to task $m, m = 1,2,...,M$, the m^{th} block of D_1 elements are set as the D_1 compound features of data sample i and all the remaining elements as 0's. Such treatment simplifies the problem and the model parameters of all M tasks can be optimized simultaneously. Each training sample is included in the objective cost function with a term consisting of logistic and exponential functions, so the cost function of thousands of training samples have thousands of such terms. The LBFGS (Limited-memory

Broyden-Fletcher-Goldfarb-Shanno) method is used to maximize the cost function [Matthies & Strang, 1979, Nocedal, 1980].

6.3.3 Boosted multi-task learning

When it is difficult to predefine task relatedness, an alternate method is to jointly learned task relatedness during the training process. Wang *et al.* proposed a boosted multi-task learning algorithm [Wang et al., 2009] for face verification, and achieved significant performance improvement. Here we present a boosted MTL method for protein-chemical interaction prediction by *repurposing* this method to fit our need. This process is like drug repurposing in recent drug discovery.

The basic idea of boosting is that a strong learner can be achieved by appropriately combining a set of weaker base learners. Here base learners are compound feature-based decision stump. AnyBoost [Mason et al., 2000] is an abstract boosting algorithm for finding linear combinations of functions that minimize arbitrary cost functionals, while many existing boosting methods, including AdaBoost [Freund & Schapire, 1995] which directly uses prediction errors as cost functions, are found to be just special cases of AnyBoost. AnyBoost can handle more complex problems that may contain hidden variables, with a gradient descent optimization methods. Specifically, we learn K independent boosting classifiers jointly for M target proteins, where $K < M$ to enforce clustering of similar tasks. Every task produces its own final classifier as a weighted linear combination of the K boosting classifiers. Task relatedness is thus modeled by the composition of the K boosting classifiers and their weights for each task.

6.3.3.1 Base learner for boosting

The very first step of a boosting approach is to define its base learners. A base learner should be a weak learner with prediction accuracy slightly better than random guessing. In this method a simple two-leaf decision stump derived from compound substructure features is used. Given a set of compound features $\{f_i\}$ extracted from all training samples $\{x_i\}$, a decision stump is defined as follows:

$$h(x_i) = \begin{cases} I & f_i \text{ exists in } x_i \\ -I & \text{otherwise} \end{cases} \tag{6.3}$$

where $I \in \{1, -1\}$, and the value of I is selected to minimize the prediction error using this base learner, so that it has at least 50% accuracy. A specific I value is determined for each task by applying the base learner on the training data of that task.

6.3.3.2 Boosted multi-task learning algorithm

Given M tasks (target proteins) and each task is to predict if a set of compounds interact with a given protein, we learn K independent boosting classifiers, where $K < M$. Each boosting classifier is a weighted linear combination of T baser learners defined using T boosting iterations as follows:

$$h_k(x_i) = \sum_{t=1}^{T} \alpha_{k,t} h_{k,t}(x_i), \tag{6.4}$$

where $h_{k,t}(x_i)$ and $\alpha_{k,t}$ are the base learners defined in Eq.(6.3) and the optimal boosting coefficient for boosting classifier k ($k = 1, ..., K$) obtained in iteration t. The key point for identifying the K boosting classifiers is to select $h_{k,t}(x_i)$ and $\alpha_{k,t}$ minimizing the training error.

With logistic loss functions, a boosting classifier h_k, and a compound feature x_i, the probability that a compound interacts with a protein is defined as:

$$p(y_i|x_i, h_k) = \frac{1}{1 + \exp(-y_i h_k(x_i))}, \tag{6.5}$$

where y_i = 1 denotes that the compound does interact with the target protein using feature x_i as base learner, and y_i = -1 otherwise. When we have multiple tasks to learn at the same time, the joint probability distribution is the product of the probability of each task as defined above.

Now we have K boosting classifiers $\{h_k\}$ for M tasks, the key point is to evaluate how well classifier k explains the data of task m ($m = 1, ..., M$) with a prior η_k on each classifier k. Previously [Wang et al., 2009] has proposed an expectation-maximization(EM) algorithm on multi-task learning for face recognition. We here use a similar EM algorithm with T iterations to learn $\{h_k\}$

and $\{\eta_k\}$ as follows:

$$\begin{aligned} q_{m,k}^{(t)} &= p(c_m = k|Y_m, X_m, \{h_{k'}^{(t)}\}, \{\eta_{k'}^{(t)}\}) \\ &= \frac{p(\eta_k^{(t)}|Y_m, X_m, h_k^{(t)}) p(Y_m|X_m, h_k^{(t)})}{p(Y_m|X_m, \{h_{k'}^{(t)}\}, \{\eta_{k'}^{(t)}\})} \\ &= \frac{\eta_k^{(t)} \prod_i p(y_{m,i}|x_{m,i}, h_k^{(t)})}{\sum_{k'=1}^{K} \eta_{k'}^{(t)} \prod_i p(y_{m,i}|x_{m,i}, h_{k'}^{(t)})}, \end{aligned} \tag{6.6}$$

where $c_m \in \{1, 2, ..., K\}$ is the index of the selected boosting classifier for task m, η_k serves as the prior probability of boosting classifier k. The "(t)" symbol on a variable denotes its value in the t^{th} iteration. X_m and Y_m are column vectors representing all samples of task m and their labels, while $x_{m,i}$ and $y_{m,i}$ denote an sample i of task m and its label. Thus $q_{m,k}$ stands for the correlation coefficient between task m and boosting classifier k. The Eq.(6.6) is the E-step of the EM algorithm, while the M-step is as follows:

$$\eta_k^{(t+1)} \propto \sum_m q_{m,k}^{(t)}. \tag{6.7}$$

$$h_k^{(t+1)} = \underset{h_k}{\operatorname{argmax}} \sum_{m,i} q_{m,k}^{(t)} \log[p(y_{m,i}|x_{m,i}, h_k)]. \tag{6.8}$$

Eq.(6.8) provides a complex cost function for the AnyBoost algorithm to learn $\{h_k\}$ and $\{\eta_k\}$, and it can be further written as follows by plugging in Eq.(6.5):

$$C_k^{(t+1)} = -\sum_{m,i} q_{m,k}^{(t)} \log[1 + \exp(-y_{m,i} h_k(x_{m,i}))]. \tag{6.9}$$

Let $h_k^{(t+1)} = h_k^{(t)} + \alpha_{k,t+1} h_{k,t+1}$ be the updating function of h_k. According to AnyBoost we first maximize the function $C_k^{(t+1)}$ by taking the derivative:

$$\frac{\partial C_k^{(t+1)}}{\partial h_k(x_{m,i})} = w_{m,i}^k = q_{m,k}^{(t)} \frac{y_{m,i}}{1 + \exp(y_{m,i} h_k^{(t)}(x_{m,i}))}. \tag{6.10}$$

Without detailed derivation of the AnyBoost algorithm, we find a new base learner $h_{k,t+1}$ by maximizing $\sum_{m,i} w_{m,i}^k h_{k,t+1}(x_{m,i})$, and then find its weight $\alpha_{k,t+1}$ by maximizing the cost function $C_k^{(t+1)}$.

After the optimal $\{h_k\}$ and $\{\eta_k\}$ have been learned by the EM algorithm, the final model of each task is projected into the functional space of these boosting classifiers $\{h_k\}$, and the probability of a given compound x that interacts with its target m is given as follows:

$$\begin{aligned} & p(y|x, X_m, Y_m, \{h_k\}, \{\eta_k\}) \\ = & \sum_{k=1}^{K} p(h_k|X_m, Y_m, \{h_k\}, \{\eta_k\}) p(y|x, h_k) \\ = & \sum_{k=1}^{K} q_{m,k} p(y|x, h_k). \end{aligned} \tag{6.11}$$

The final classifier for task m is defined as y = 1 if $p \geq$ 0.5 and -1 otherwise.

There are some tricks in the implementation of this boosted MTL method. Using a sequential greedy search the best base learner that maximize the cost function can be identified at each iteration. The trick is that the process could be very time-consuming if the number of candidate features is large. Removing candidate features that are identical for all training samples can significantly reduce the searching space and speed up the process. The second trick is to compute the boosting coefficient $\alpha_{k,t}$ for the previously selected base learner $\{h_k^{(t)}\}$ by maximizing the convex cost function $C_k^{(t+1)}$. The cost function can have up to thousands of logistic and/or exponential terms, and are optimized by the L-BFGS method [Matthies & Strang, 1979, Nocedal, 1980].

The above method takes only the compound features into consideration. It is tempting to ask if the performance can be improved when we integrate protein features into this framework. A simple way is to use tensor product of protein and compound features as candidate features. Given a $D_1 \times 1$ protein vector and a $D_2 \times 1$ compound vector, the tensor product can be concatenated into a $D_1 D_2 \times 1$ vector. We refer to the original boosted MTL method as *BoostMTL1* and this variant as *BoostMTL2* in the rest of this chapter.

6.4 Results

In this section we begin our discussion with an introduction to the collection, cleanup, and representation of protein-chemical interaction data, and then evaluate the performance improvement of our proposed MTL methods over three baseline methods on the three data sets.

6.4.1 Data set and feature extraction

We manually created two sets of protein-chemical interaction data from the BindingDB database [Liu et al., 2007] and the PubChem database (http://pubchem.ncbi.nlm.nih.gov). The first one (the AD set) has 10 target proteins associated with Alzheimer's Disease (AD) [Stephenson et al., 2005], and the second one (the Cancer set) consists of eight target proteins associated with various cancers such as breast, prostate, and lung cancer [Benson et al., 2006]. For data from PubChem, the binding activity for each compound has been defined. The BindingDB database provides specific IC_{50} and/or K_i values for each binding compound of a target, and a compound is defined as active (inactive) to its target if its IC_{50} or K_i value $\leq$ 100 nM ($\geq$ 10,000 nM). We have set these threshold values weaker than in literature to enhance the size of the "binding" set (and implicitly recognizing that some compounds bind even if they do not couple strongly enough to be considered a hit).

Table 6.1: Target proteins in the AD set and the Cancer set.

AD	Active	Inactive	Cancer	Active	Inactive
AChE	71	210	AR	44	61
APP	107	32	CYP19A1	275	226
BHMT	4	0	ATM	1	42
CASP	201	101	ERBB1	2	23
IL6	0	2	HER2	61	6
MHCII	4	13	RAR-α	31	33
NEP	360	59	FLT1	122	187
S100B	0	2	KDR	62	15
SNCA	25	19			
MAPT	44	65			

Note that in both sets the negative samples are real and verified by experiments. Although in reality there are much more negative than positive samples, our data sets have slightly more positive than negative samples (826 *vs.* 503 in the AD set and 598 *vs.* 597 for the Cancer set), since experimentalists are not of great interest to verify negative samples. These two data sets are nonetheless highly unbalanced, since some proteins have few binding compounds, and some of them have only positive or negative samples. For instance, in the AD set only four active, two inactive, and two inactive compounds are available for protein BHMT, IL6, and S100B, respectively. Such data characteristics are the first motivation for us to develop sophisticated MTL algorithms for protein-chemical interaction prediction. Detailed distributions of binding compounds for the first two sets are listed in Table 6.1.

The third class of proteins we are interested in is GPCR proteins, and we simply adapted an existing data set of GPCRs from [Jacob & Vert, 2008], which consists of 100 GPCR proteins, 219 compounds, and 399 positive interaction pairs. The distribution of the protein-chemical binding data is shown in Figure 6.1. Unlike the first two data sets, there are no validated negative samples for proteins in this GPCR set, and an equal number of randomly selected protein-chemical pairs that are known not to be positive are treated as putative negative samples.

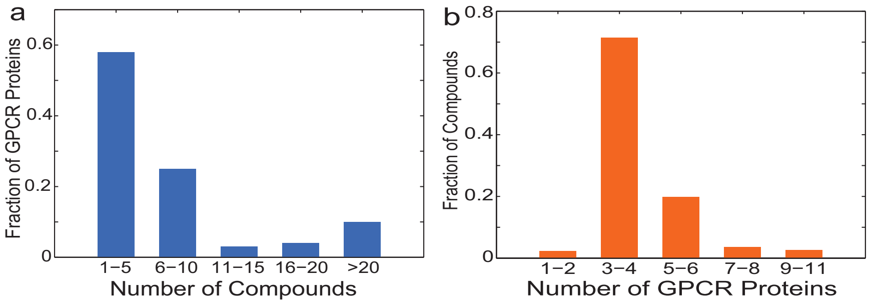

Figure 6.1: Distribution of (a) the number of binding compounds of GPCR proteins, and of (b) the number of binding GPCR proteins of compounds.

Molecular signature descriptors [Faulon et al., 2004] are used to extract features from chemical structures. We use this method because it achieves excellent performance for extracting features from chemical structures, and generally outperforms the widely used DRAGON descriptors (http://www.talete.mi.it) from our previous work [Zhang & Huan, 2010]. In detail, each element of a molecular signature vector is the number of occurrences of a particular atomic signature in the molecule. An atomic signature is a canonical representation of the chemical substructure surrounding a particular atom, including all atoms and bonds up to a predefined number of consecutive bonds from the given atom, called the signature height. For a set of chemicals or proteins, this method first extracts all distinct atomic signatures and their frequencies in each structure, and then makes a union of n features from all structures. Finally, each structure i is converted into a vector $\mathbf{v_i}$ of n elements, and each element j corresponds to the feature j. If structure i contains feature j for f_j times, we have $\mathbf{v_i}[j] = f_j$, where f_j is a non-negative integer and $f_j = 0$ means that feature j

doesn't exist in structure *i*.

Note that signature descriptors are originally developed for chemicals. When they are applied to proteins, the method simply makes a pseudo structure for a protein by replacing each amino acid with its chemical structure, since in many cases it is difficult to obtain the 3D structures for proteins. The strategy looks too simple, but in practice signature descriptors work very well for protein sequences. In this chapter we set the signature height at 2 for compounds and at 4 for proteins, and remove all signature descriptors that are existent in less than 1% compounds (proteins). We end up with $\sim$200 features for each compound and $\sim$50 features for each protein. Here we found that there are less signature descriptors for all proteins than for all small-molecule chemicals in the data sets. This looks counter-intuitive, but is reasonable since we have much more distinct chemicals (1200-1300) than proteins (8-10). Refer to the original publication [Faulon et al., 2004] for more detailed introduction to signature descriptors.

6.4.2 Baseline methods

SVMs and ℓ_1-regularized logistic regression (ℓ_1-RLR) are two excellent STL methods and selected as our comparison baselines. Note that no protein descriptors are used in these two methods. If a task has only positive or negative samples, STL methods cannot build models for it and the prediction accuracy for this task is set to 50% for comparison. The third reference method is a widely used MTL approach called *tensorSVM*, which integrates protein and compound information using tensor product of their feature vectors and computes the similarity between two pairs of proteins and compounds as the product kernels of the protein similarity and the compound similarity. We utilized the RBF kernels and the precomputed kernels in the LIBSVM package [Chang & Lin, 2001] for the SVM and *tensorSVM* method, respectively. The ℓ_1-regularized logistic regression algorithm proposed by Boyd *et al.* [Koh et al., 2007] is used for the method ℓ_1-RLR.

6.4.3 Optimization of model parameters

With 20% randomly selected samples of each task as the independent testing set, we applied 10-fold cross validation on the rest of the data, used exponential grid search to select model parameters (cost parameter C for SVM and tensorSVM, regularization strength λ for ℓ_1-RLR), and evaluated the models on the unused testing set to obtain testing accuracy. This process is repeated for 10 times, and the mean testing accuracy and F_1 score are reported. For all methods, testing accuracy, precision, recall, and F_1 score are defined as (TP+TN)/S, TP/(TP+TN), TP/(TP+FN), and 2*TP/(2*TP+TN+FN), respectively, where TP is true positive, TN is true negative, FN is false negative, and S is the total number of testing samples.

The *GaussMTL1* method involves two parameters to be tuned: the protein feature scaling factor ρ and the regularization strength λ. Tuning of ρ is expensive, and preliminary results show that the validating accuracy is not very sensitive to small changes of ρ. We apply 2D grid search for optimizing ρ and λ jointly. Here 11 values of $\rho \in [0,1]$ were uniformly selected, and at each value of ρ exponential grid search of λ was performed in each partitioning as described above. We repeat the above process for 10 times to identify the optimal λ by maximizing the mean overall accuracy of all tasks. Finally, the learned logistic regression parameters and the optimal ρ and λ were applied to the testing set to obtain the generalization accuracy. We repeated the whole experiments for 10 times and reported the mean testing accuracy and F_1 score.

The *boosted* MTL methods also have two model parameters to be discreetly selected: the number of base learners in each boosting classifier (T), and the number of boosting classifiers (K). To avoid an expensive exhaustive grid search of T and K jointly, we used an iterative search procedure as follows: we first fixed K at $M/2$ (M is the number of tasks in a data set) and searched for optimal T from six values: 5, 10, 20, 30, 40, and 50. Here the variant tensor-product boosted MTL method (*BoostMTL2*) was also investigated. Once the optimal T was identified, we fixed the T value and again searched for optimal K from values $1,2,...,M$. Theoretically we could continue the process further, but we stop after the first iteration since empirical study showed that there was no much further improvement.

Table 6.2: Prediction accuracy comparison on the AD data set and the Cancer data set. $^*\ell_1$-RLR stands for ℓ_1-regularized logistic regression. The standard deviations are included in parenthesis when available or non-zero.

Method	Overall	AChE	APP	BHMT	CASP	IL6	MHCII	NEP	S100B	SNCA
GaussMTL1	**91.0(1.0)**	**94.0(1.7)**	**95.7(3.7)**	**95.0**	**96.2(3.1)**	**95.0**	72.5(18.4)	**93.6(1.6)**	90.0(31.6)	**71.1(10.7)**
		88.2(3.1)	**90.8(2.8)**	**95.0**	**90.7(2.7)**	**95.0**	72.5(18.4)	87.5(1.6)	90.0(31.6)	**67.1(12.2)**
GaussMTL0	89.3(1.0)	91.4(1.8)	94.0(3.0)	90.0	94.4(3.1)	100.0	72.5(20.4)	92.8(2.2)	90.0	67.9(10.2)
		87.2 (2.2)	90.1(1.9)	90.0	88.7(3.5)	100.0	72.5(20.4)	83.5(1.0)	90.0	62.2(9.5)
tensorSVM	57.8	91.8(0.0)	85.1(0.1)	10.0	88.1(0.1)	25.0	68.3(0.4)	85.3(0.0)	10.0(0.3)	47.5(0.2)
		79.5(32.2)	61.7(40.1)	0.0	74.0(24.1)	0.0	10.0(31.6)	61.1(30.3)	0.0	40.0(51.6)
BoostMTL1	87.8(1.4)	88.2(2.9)	95.0(3.8)	90.0	94.3(3.3)	90.0	77.5(27.5)	89.8(3.5)	**95.0**	53.3(18.0)
		81.5(6.0)	88.1(6.8)	90.0	81.2(9.4)	90.0	77.5(27.5)	87.1(2.2)	**95.0**	56.2(11.0)
BoostMTL2	87.7(1.3)	88.4(2.9)	95.0(3.8)	90.0	94.4(2.7)	90.0	**82.5(26.5)**	89.3(3.3)	90.0	51.1(18.3)
		80.7(6.2)	86.1(7.2)	90.0	81.1(8.9)	90.0	**82.5(26.5)**	**87.6(2.0)**	90.0	53.8(10.8)
SVM	77.3	89.2(7.0)	90.8(6.2)	50.0	91.7(5.2)	50.0	50.0	82.7(6.7)	50.0	48.3(12.3)
		81.7(8.3)	88.9(5.2)	50.0	88.3(8.9)	50.0	50.0	71.1(7.3)	50.0	55.0(7.2)
ℓ_1-RLR*	78.6	90.0(3.7)	85.6(9.2)	50.0	92.4(2.5)	50.0	50.0	83.1(6.1)	50.0	54.0(17.1)
		78.3(10.9)	84.4(9.4)	50.0	82.2(11.6)	50.0	50.0	75.6(11.8)	50.0	60.6(9.2)

Method	Overall	MAPT	AR	CYP19A1	ATM	ERBB1	HER2	RAR-α	FLT1	KDR
GaussMTL1	**88.6(0.7)**	63.6(6.1)	**91.0(4.2)**	86.6(1.6)	96.7(5.4)	**100.0**	95.7(5.0)	**85.4(6.7)**	**90.5(2.5)**	79.4(10.6)
		59.1(5.6)	**84.4(5.0)**	**71.4(6.0)**	96.1(2.5)	**100.0**	**91.3(4.2)**	67.5(14.4)	70.9(7.0)	80.0(5.6)
GaussMTL0	86.5(1.0)	65.4(6.5)	90.0(3.5)	84.6(2.2)	93.7(5.4)	100.0	94.0(5.0)	83.3(5.7)	88.3(2.6)	76.2(10.6)
		56.4(7.2)	81.0(6.2)	70.5(5.6)	92.2(2.5)	100.0	90.7(3.9)	64.4(13.9)	70.8(7.0)	76.5(4.8)
tensorSVM	74.8	**67.4(0.1)**	80.2(0.1)	88.2(0.0)	73.3(0.1)	62.6	60.3(0.1)	74.1(0.1)	82.9(0.1)	76.6(0.2)
		29.2	54.2	69.2(17.4)	20.0	10.0	50.0	40.0	50.2(9.8)	13.3
BoostMTL1	84.8(4.6)	65.9(7.8)	82.4(14.7)	85.7(4.3)	**98.9(3.5)**	98.0	**96.4(6.1)**	73.8(24.4)	81.0(6.8)	**83.8(7.3)**
		57.1(7.8)	58.0(13.8)	57.1(11.9)	**98.3(2.2)**	98.0	90.6(2.3)	57.0(10.7)	66.2(4.6)	**81.2(3.6)**
BoostMTL2	85.2(4.5)	66.4(7.2)	81.4(14.3)	86.9(3.4)	98.9(3.5)	98.0	96.4(6.1)	70.8(27.1)	81.9(7.2)	83.1(8.4)
		56.2(7.6)	63.4(10.5)	55.0(9.7)	98.3(2.2)	98.0	90.6(2.3)	59.3(12.0)	67.7(4.0)	79.3(6.6)
SVM	81.2	61.2(9.2)	88.1(8.0)	**88.4(2.1)**	50.0	50.0	50.0	79.2(9.8)	90.4(3.8)	60.0(22.5)
		57.8(8.8)	79.0(13.7)	68.0(13.2)	50.0	50.0	50.0	66.0(16.5)	**71.0(9.9)**	66.0(10.8)
ℓ_1-RLR*	79.2	66.7(8.8)	85.4(8.2)	87.8(3.0)	50.0	50.0	50.0	75.6(11.2)	87.1(3.7)	60.0(11.5)
		54.4(7.3)	73.0(15.7)	55.0(12.7)	50.0	50.0	50.0	**71.0(15.2)**	71.0(7.0)	59.0(8.8)

6.4.4 MTL with substantial training data

We first tested the hypothesis that our proposed MTL methods can build more accurate models for predicting protein-chemical interactions by learning multiple related tasks (target proteins) jointly. With the optimal model parameters ρ, K and T, we used the experimental procedures described above to evaluate the performance of our methods comparing with the baseline methods, and showed the testing accuracy on the AD set and the Cancer set in Table 6.2. Note that in the tables each method has two rows of results, and the top row is for results from experiments using 80% data samples of each task, and the bottom row is for results from experiments using at most 20 data samples of each task. We highlighted the best accuracy obtained using various methods for the whole data set and for each task.

We noticed that MTL methods always perform better than STL methods and *tensorSVM* on overall testing accuracy. The overall performance of the *GaussMTL1* method and the *BoostMTL1*

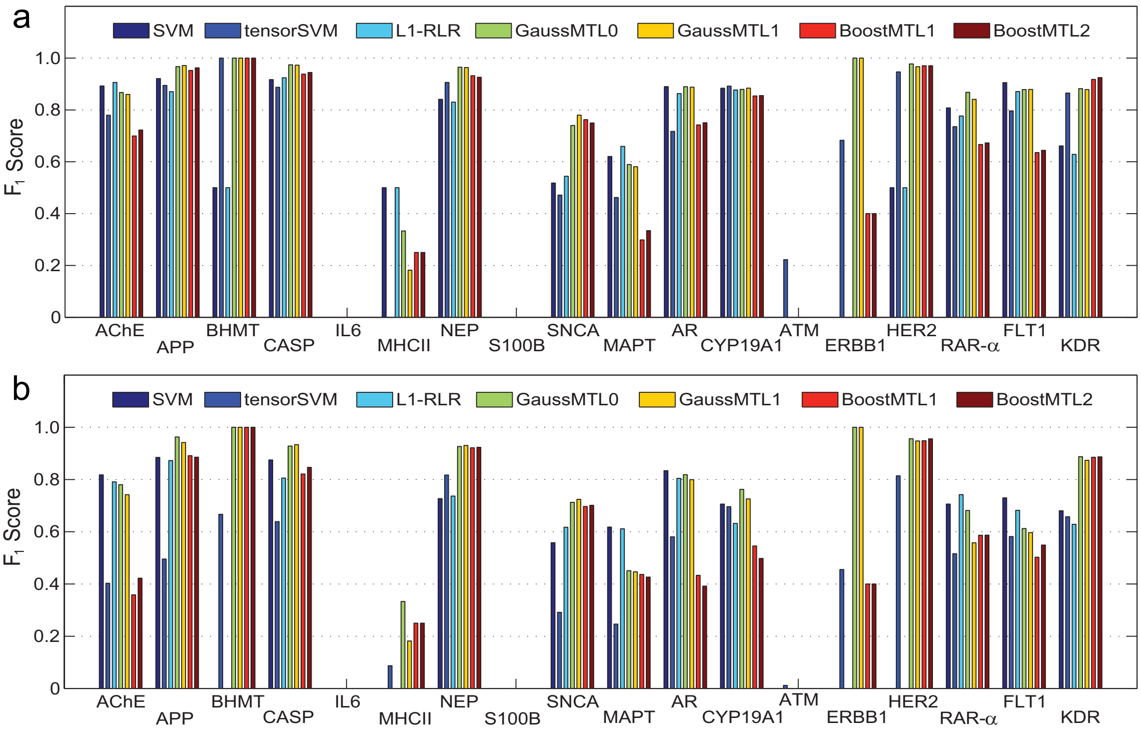

Figure 6.2: Distributions of task-by-task prediction results measured by F_1 score from experiments (a) using 80% data samples of each task, and (b) using at most 20 data samples of each task as the training set. Results are mean over 10 repeated experiments using random sampling, and deviations are not shown.

method is at least 12.4% and 9.2% better than that of the reference methods for the AD set, and 7.2% and 2.6% better for the Cancer set, respectively. We compared the results of the *GaussMTL1* method to ℓ_2-regularized logistic multi-task regression method: a special case of the *GaussMTL1* method by setting the scaling factor $\rho = 0$, which means no protein information is taken into consideration. This method is labeled as *GaussMTL0*. We observed a 1.7% improvement on the overall accuracy by incorporating protein features into the models, which reveals that protein features do help construct more accurate learning models.

Though the overall accuracy of *GaussMTL1* is 3.2∼3.8% higher than the boosted MTL methods on both data sets, there is no clear winner on accuracy of specific tasks. Explicitly (*GaussMTL*) or implicitly (*BoostMTL*) defining task relatedness does not produce significant difference in the testing results. For the two boosted MTL methods, the tensor-product boosted MTL method (*BoostMTL2*) performs almost no better than the compound feature boosted MTL method (*Boost-*

Table 6.3: Prediction accuracy using various numbers of training data. * Percentage of training samples from other tasks.

# Training Samples	80%*		20%*	
	MHCII	ERBB1	MHCII	ERBB1
0	40.0%	38.4%	32.9%	54.8%
4	63.1%	92.4%	67.7%	91.9%
8	67.8%	94.1%	67.8%	95.3%
12	78.0%	98.5%	74.0%	95.4%

MTL1), although it affords substantially more computational workload.

When we break down the results task by task, we noticed that the major improvement of our methods come from those tasks with few training samples. Since our methods are designed to have a strong regularization among tasks and hence perform knowledge transfer among related tasks, it is clear that such regularization is essential for constructing high quality models for tasks with low sample sizes. In addition, we calculated the results of F_1 score (a measure that combines precision and recall using their harmonic mean) for each task, and similar trends as using prediction accuracy holds as shown in Figure 6.2(a). Similarly, we also calculated the results of F_1 score using at most 20 training samples for each task as shown in Figure 6.2(b), and found similar patterns supporting the improved performance of our MTL methods.

To better investigate the relationship of task sample size, regularization, and testing accuracy, we conducted a small case study using two tasks with about 20 data samples: MHCII in the AD set and ERBB1 in the Cancer set. They were picked just due to the small number of labeled samples available. Experimental results on these two tasks are coherent with other tasks as shown in Table 6.2 and 6.2. We incrementally set the number of samples in the training set as 0, 4, 8, and 12 with either 80% or 20% data samples of other tasks used for training, to investigate the effect of the sample size on model accuracy. The results of the *GaussMTL1* method are listed in Table 6.3. We notice that there is a significant improvement if we have more than 4 samples for the task. If we use more than 12 training samples, our methods have been able to achieve decent prediction accuracy.

6.4.5 MTL with limited training data

In previous experiments, we used 80% of the data of each task for training and the rest 20% for testing. In order to test the robustness of our proposed MTL methods, we significantly reduced the number of training samples and repeated the experiment as described previously. In particular, for a task with more than 20 samples, we kept at most 20 samples from it and all data from other tasks in the training set, and moved the extra samples of this task into the testing set. If a task has less than 20 training samples, no more samples will be moved out of the training set. We conducted this experiment for each task in the two data set, and repeated it for 10 times to calculate the mean accuracy. All results are shown in the lower rows for each method in Table 6.2.

The results are consistent to what we reported before. For 16 out of the 18 tasks in the two data sets, our methods perform better than the reference methods, while ℓ_1-RLR and SVM outperform our methods on the RAR-α task and the FLT1 task for 3.5% and 0.1%, respectively. For the nine tasks with substantial data samples among the 16 tasks, our methods outperform reference methods for 6.0% on average. For the seven tasks with limited samples, our methods have more significant predictive performance: 92.8% for *GaussMTL1*, 94.9% for *BoostMTL1* and 95.6% for *BoostMTL2*. The performance difference is originated from the fact that our proposed MTL methods transfer knowledge among related tasks while integrating data from diverse domains.

6.4.6 MTL for multi-target PCI prediction

Multi-target drug design is a promising direction for overcoming complex diseases in future drug discovery. Due to technical difficulty and data availability, the AD set and the Cancer set consists of almost no promiscuous compounds that interact with multiple target proteins. We then adopt the GPCR data set from [Jacob & Vert, 2008], and its data distribution is shown in Figure 6.1. The 100 GPCR proteins belong to the same protein family, and many of them have been known drug targets. However, most GPCR proteins in the data set has very few binding compounds, and STL methods will not work for them. Here we use our proposed MTL methods to handle this multi-label classification problem.

Though theoretically the *GaussMTL* method can be used for this multi-label classification problem, the covariance matrix of 100 tasks are too large to be efficiently handled by computer systems, so only the *BoostMTL1* method will be investigated in this section. Compared to previous experiments, MTL for multi-target protein-chemical interaction prediction involves two critical changes: one is how to partition data into training and testing sets, and the other is how to calculate the accuracy and to evaluate the multi-label prediction performance of the *BoostMTL1* methods. There are two ways to partition data: 1) random sampling as described above, and 2) random partitioning that is restricted so that a given promiscuous compound must belong to either training set or testing set, but not both. The first partitioning method allows a compound to be in both training set (active to target α) and testing set (active to target $\beta \neq \alpha$). This is equivalent to the case that we know some targets of a compound and want to find more targets for it, while the second method is more challenging since it attempts to find binding proteins for a compound without knowing any binding information of it.

For evaluating the capability of the *BoostMTL1* method on handling multi-label classification problem, the prediction accuracy is calculated only based on promiscuous compounds with binding information to at least two target proteins. Since the negative samples of this GPCR data set is putative, we also calculate the prediction recall (number of true positives over all positives) for each method. A summary of the accuracy and recall results over all tasks is shown in Table 6.4, which demonstrates that the *BoostMTL1* method significantly outperforms *tensorSVM* using both partitioning methods described above on both prediction accuracy and recall. Unsurprisingly, the partitioning method 1) performs better than method 2) since some binding information of testing compounds is carried in training data using method 1).

Table 6.4: Prediction results for multi-target protein-chemical interactions on the GPCR data set.

Method	Boosted MTL1		TensorSVM	
	Partition 1	Partition 2	Partition 1	Partition 2
Accuracy	92.9%	79.7%	71.9%	62.2%
Recall	97.2%	85.7%	76.5%	70.3%

We further calculated the multi-label prediction accuracy and recall based on the predicting

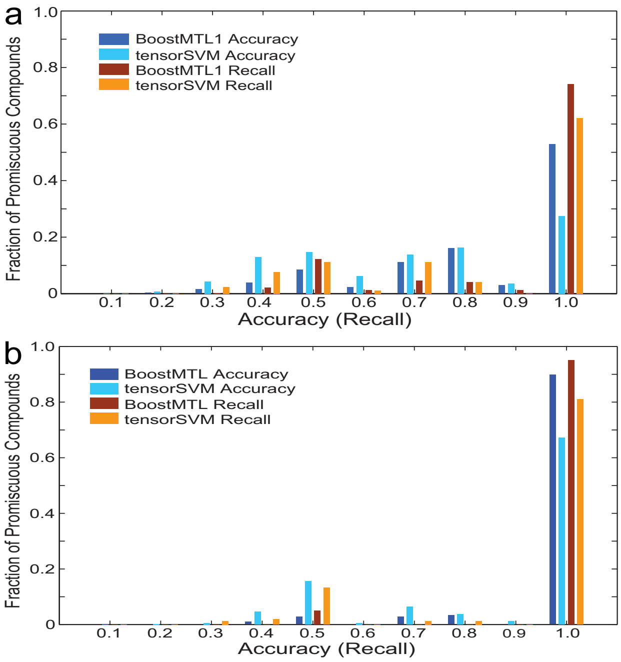

Figure 6.3: Histogram of the accuracy and recall of each promiscuous compound when (a) it can only exist in either the training or the testing set, but not both; and (b) part of the binding information of testing compounds may exist in the training set.

results of each promiscuous compound. For instance, if a compound in the testing set is interacting with three proteins and inactive toward two proteins, and the prediction reports two true positives and two true negatives, we can easily calculate the accuracy and recall on this compound as 80% and 66.7%, respectively. We calculate the accuracy and recall for each promiscuous compound, repeat the experiment for 10 times using both partitioning methods, and obtain the distribution of the accuracy and recall for promiscuous compounds in the GPCR set, as shown in Figure 6.3. In both distribution plots, most compounds have accuracy and recall in the range of 0.9∼1.0, and our *BoostMTL1* method significantly outperforms *tensorSVM* on predicting promiscuous compounds. The superiority of our boosted MTL method for multi-target protein-chemical interaction prediction is clearly demonstrated with high accuracy and recall.

6.4.7 Multi-task learning with imbalanced training data

In previous experiments, for each task we randomly selected 80% data or at most 20 samples as the training set. We didn't restrict the ratio of positive and negative samples in the training set, and literally this ratio is approximately close to the ratio of all positive and all negative samples for each task. In many real applications such as virtual screening, the data sets usually consist of much more negative than positive samples and are thus very imbalanced. In this section we investigated how the ratio of positive/negative samples influenced the prediction accuracy. First we picked task AChE and selected 60-80% data as the training set depending on the data availability, and randomly selected 80% data as the training set for all other tasks in the AD set. We enforced that the training set of the AChE task consists of 10%, 20%, 30%, 40%, and 50% positive samples. At each ratio we conducted an experiment using the four variants of our MTL methods, as described in Section 3.3. The prediction accuracies for the task AChE using training sets with various ratios of positive/negative samples were plotted in Figure 6.4, which demonstrated that although the prediction accuracy can be improved with more balanced training data, the performance of our MTL methods was still promising with highly imbalanced training sets.

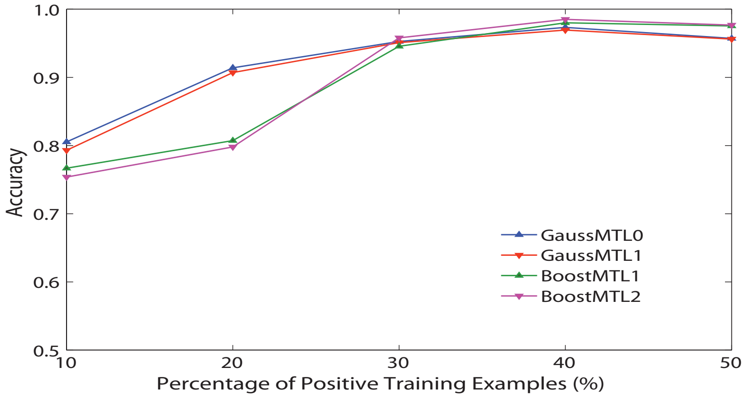

Figure 6.4: Prediction accuracy fluctuation when different percentages of positive samples in the training set of task AChE.

6.4.8 Sensitivity of model parameters

In this section we investigated the sensitivity of prediction accuracy to the changes of critical model parameters in our MTL methods: ρ in the GaussMTL method, and K and T in the BoostMTL methods. The experimental procedure as described previously is applied to show how the prediction results change when more protein information are taken into consideration. It does reveal that integrating protein features into the MTL framework improves the prediction accuracy for 1.7%. For the AD set, the accuracy fluctuates and the best performance of 91.0% is achieved at $\rho = 0.6$. When ρ increases the accuracy on the Cancer set decreases, and $\rho = 0.1$ gives the best prediction accuracy 88.6%.

The results of experiments on the parameter T for both boosted MTL methods are shown in Figure 6.5(a), which reveals that both methods achieve the best accuracy at $T = 10$ for the AD set and $T = 40$ for the Cancer set. Although introducing protein features using tensor-product kernels significantly increases the dimensionality of training data for one to two orders of magnitude, it shows almost no performance improvement compared to the BoostMTL1 method. In addition, the results of experiments on the parameter K are plotted as in Figure 6.5(b), which shows that when $K < M$ the accuracy increases while there are more boosting classifiers, and the best accuracy is obtained at $K = 9$ (87.8%) and $K = 7$ (84.8%) for the two data sets, respectively. More boosting classifiers increase the dimensionality of the final boosting models in the functional space, although more computation cost is needed.

6.5 Discussion

Traditional single-task learning methods learn each task independently and neglect potential rich information resources hidden in other related tasks, and their performance is largely restricted by not only the insufficient availability of labeled samples for many proteins, but also the emergence of more challenging problems such multi-target drug design. In this chapter, we first transform this typical multi-label classification problem into the MTL framework, and then develop novel

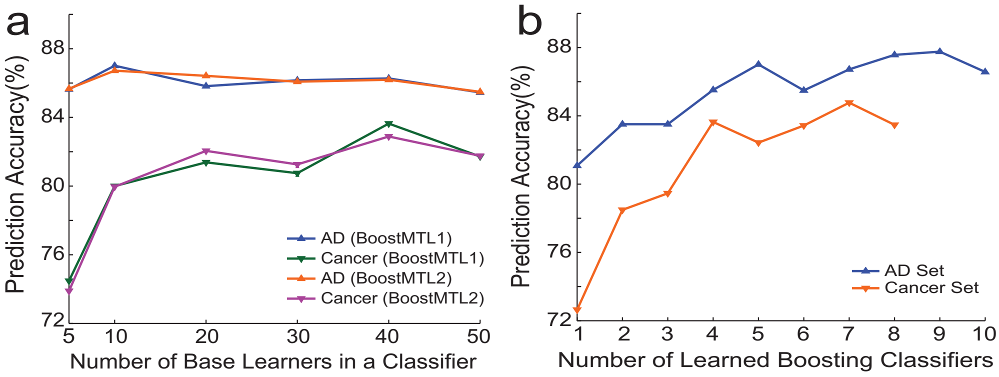

Figure 6.5: Model parameter sensitivity of the boosted MTL methods. (a) Number of boosting classifiers fixed at $M/2$ (M = number of tasks); (b) Number of base learners in each boosting classifier fixed.

approaches to integrating heterogenous data, learn multiple related tasks jointly by capturing task relatedness and transferring knowledge among them. In the first method, we encode a covariance matrix of protein similarity kernels into the Gaussian prior of logistic regression parameters so that information from both proteins and compounds is taken into consideration. In the second method we design a two-leaf decision stump based on compound substructure features as boosting base learners, and apply the AnyBoost algorithm to boost these weak learners and to select highly discriminative features. Experimental results not only demonstrate that our methods significantly outperform baseline methods with either substantial data or limited data from each task, but also reveal that the *BoostMTL1* method perform much better than the *tensorSVM* method on identifying promiscuous compounds for multi-target drug design. Our proposed MTL methods can not only be applied to protein-chemical interaction prediction, but also a wide range of other association prediction problems. They are especially developed for data samples consisting of two types of objects, such as protein-chemical/DNA/gene interactions, and gene-function annotations.

Both MTL methods have their limitations. *BoostMTL* doesn't support learning a task without training data from it, while the *GaussMTL* method can perform such tasks, though the accuracy is not highly satisfactory. *GaussMTL* has challenging time complexity since computing the inverse of of the covariance matrix increases cubically with the number of tasks increasing, while *BoostMTL* is an efficient algorithm and its time complexity is linear relative to the number of candidate features. The major limitation of our MTL methods is that the data sets we used are relatively small, and we have not tested the MTL algorithms in a virtual screening environment for evaluating their performance with highly imbalanced databases consisting of $10^4 - 10^5$ compounds. In conclusion, our proposed MTL methods provide valuable solutions to the previously mentioned challenges on predicting protein-chemical interactions. It is highly promising that integrating MTL and boosting approaches into the same framework can help overcome many other critical machine learning problems.

Chapter 7

Drug-induced QT Prolongation Prediction Using Multi-view Learning

There are much more ADRs than approved drugs, according to the SIDER database [Kuhn et al., 2010]. We first select some specific critical ADR and construct accurate models using a sparse multi-view learning algorithm to handle the noisy data. In next chapter, we then extend our method to the multi-task learning framework so that multiple ADRs can be predicted jointly. The ADR we picked is *Torsades de pointes* (TdP), which is a rare form of polymorphic ventricular tachycardia that exhibits distinct characteristics on the electrocardiogram (ECG), and is considered as a major life-threatening condition for patients as it can degenerate into ventricular fibrillation and sudden death [Ouillé et al., 2011]. In the past decade, the single most common cause of the withdrawal or restriction of the use of marketed drugs has been recognized as the prolongation of the QT interval associated with TdP [Lasser et al., 2002]. Until 2004, at least nine drugs that were marketed in the U.S. or elsewhere have been removed from the market or had their availability restricted due to the risk of prolonging QT interval, such as Astemizole [de Abajo & Rodríguez, 1999] and Grepafloxacin [Ball, 2000]. Therefore it is crucial to discover and filter out those drug candidates with potential risk of QT prolongation and/or TdP as early as possible in the drug development pipeline to save expenses and lives in clinical trials. Note that in this chapter the terms

QT prolongation and TdP risk are used exchangeably for ease of description.

7.1 Introduction

TdP is often associated with a prolonged QT interval, which may be congenital or acquired. Drug-induced QT prolongation has triggered significant attention from pharmaceutical industry. Since 2005 the U.S. Food and Drug Administration (FDA) and European regulators have required that nearly all new molecular entities must be evaluated in a thorough QT study to determine its effects on the QT interval [UCM, 2005]. Although QT interval can be influenced by many factors, understanding of the underlying genetic mechanisms of QT prolongation have been significantly improved during the past decade. At present a few ion-channel related genes have been identified, and their gene products are found to profoundly influence the balance of ion-channel currents that determine the duration of the myocyte's action potential and thus the QT interval [Moss, 2006], including three potassium ion channels $I_{kr}(K_v 11.1$, hERG, KCNH2), $I_{k1}(K_{ir}2.1$, KCNJ2), and $I_{ks}(K_v 7.1$, KCNQ1), one sodium ion channel $I_{Na}(Na_v 1.5$, SCN5A), and one calcium ion channel $I_{CaL}(Ca_v 1.2$, CACNA1C) [Ouillé et al., 2011]. Presumably, compounds that inhibit one or more these involved ion channels are more likely to obstruct the normal cardiac ion conduction and thus prolong the QT interval. This key observation is crucial for identifying compounds with potential TdP risk.

There are various means for pharmaceutical companies to obtain QT prolongation information induced by their interested compounds, such as animal models and human clinical trials, whereas the FDA Adverse Event Reporting System is another important source for such information. However, data obtained from these sources are either expensive and time-consuming or noisy, while computationally predicting this rare but serious adverse drug effect (ADE) in humans remains highly challenging. With thorough literature search we identified 142 compounds [Ouillé et al., 2011,Woosley, 2003] that are possibly associated with QT prolongation and TdP with different risk levels, including 28 compounds labeled as *risk*, 40 as *possible risk*, and 74 as *conditional or con-*

genital risk (`http://www.azcert.org`). It is not surprising that most of these drug compounds have such uncertain labels, since it is non-trivial to determine the causation between drugs and TdP risk, especially when multiple drugs are simultaneously taken by patients. With such limited and noisy data, traditional machine learning methods cannot build accurate models for predicting the QT prolongation and TdP risks of potential drug candidates. On the other hand, additional knowledge on the mechanism of QT prolongation are available but unused.

A variety of applications have limited and noisy labeled samples such as QT prolongation prediction, where traditional single-view learning (SVL) methods such as support vector machines work poorly. Multi-view learning (MVL) has been applied to many challenging machine learning and data mining problems, especially when complex data from diverse domains are involved and only limited labeled samples are available. The underlying assumption of multi-view learning [Sindhwani & Niyogi, 2005, Sindhwani & Rosenberg, 2008, Culp et al., 2009] is that different views are conditional independent and relatively complementary to one another, so that combining knowledge from multiple views can afford performance gain. Existing MVL algorithms can be classified into three categories: co-training [Yu et al., 2007], co-regularization [Sindhwani & Niyogi, 2005], and manifold regularization [Sindhwani & Rosenberg, 2008] according to how they optimize the objective function and integrate information from multiple views. Among MVL algorithms in the co-regularization framework, all previous work used ℓ_2-norm co-regularization for it is easy to optimize the smooth objective function. In this chapter, we propose an ℓ_1-norm co-regularized MVL algorithm to boost prediction performance on the important bioinformatics problem: prediction of QT prolongation effect using the limited and noisy data available. Noticeably, although this MVL algorithm is developed for predicting the effect of QT prolongation, the method itself is domain independent and could be applied to a wide range of other real-world applications with limited and noisy multi-view data.

All previous MVL algorithms that use ℓ_2-norm co-regularization have a smooth objective function optimized by an alternate optimization approach [Yu et al., 2007, Krishnapuram et al., 2004], i.e., in each iteration one view is optimized with the other views fixed until convergence of mapping functions in all views. However, we will demonstrate that our proposed ℓ_1-norm co-regularized MVL method is more advantageous after properly reformulating the objective function and computing its gradient in the analytic form. Moreover, We developed a simultaneous optimization approach that optimizes the mapping functions from all views simultaneously. With comprehensive experimental studies, simultaneous optimization is found to be much faster than previously widely used alternate optimization.

Although ℓ_1-norm regularization (lasso) methods have been widely exploited in various application domains, our work is the first attempt to apply ℓ_1-norm in multi-view learning. It seems that the mixed regularization employed here is somehow similar to the fused lasso penalty [Liu et al., 2010, Tibshirani et al., 2005] in which the coefficients and the difference between coefficients are both lasso penalized. However, our ℓ_1-norm co-regularized MVL method regularizes the difference of functional values instead of the function difference as in the fused lasso, i.e., our regularization is in the mapping space while the fused lasso penalty is in the functional space. Refer to the related references and the Methods section for more details.

For predicting drug-induced QT prolongation using multi-view learning, the first view is easily directed to drug chemical structures, while there is a few options for the second view. One choice is the general binding profiles of the drug compounds to human protein. More specifically, we can use the binding information of compounds to those human proteins associated with QT prolongation, i.e. the five ion-channel proteins mentioned previously. However, it is challenging to obtain binding data between the compounds interested and the five QT prolongation-related ion-channel proteins mentioned above. When the binding activity between a compound and one of the ion channels is unknown or unavailable, we build protein-chemical interaction prediction models to predict it until the activity matrix is filled up completely. By learning these two-view data, ℓ_1-norm co-regularization enforces sparsity in the learned mapping functions and naturally carries the functionality of feature selection, and hence the prediction results are expected to be more accurate and interpretable. Comprehensive experimental studies are designed for demonstrating the advantages of our proposed MVL method on the prediction of drug-induced QT prolongation.

7.2 Related Work

Many computational methods for facilitating the prediction of drug-induced QT prolongation and TdP risk have been developed and applied in various applications such as drug discovery and development. Yap *et al.* [Yap et al., 2004] used linear solvation energy relationships descriptors to measure the TdP-causing potential of compounds and built prediction models with support vector machines (SVMs). Their method simply ignored the different confidence levels of data and considered all *risk*, *possible risk*, and *conditional or congenital risk* compounds as positive samples, and the interpretability and meaning of their results were limited. Recognizing the fact that the hERG protein is frequently associated with QT prolongation, Yao *et al.* [Yao, 2008] and Redfarn *et al.* [Redfern, 2003] developed models using hERG inhibition activity and other relevant data to predict drug-induced QT prolongation, and investigated its relationships with various electrophysiological properties. The limitation is that hERG is not the only or necessary binding protein to induce this serious ADE. In addition, Champerous *et al.* [Champeroux et al., 2005] classified training samples into three categories according to their confidence levels, and developed a complex algorithm called TPDscreen^{TM} combined with a database from reference compounds with available clinical data to predict the risk of TdP and QT prolongation at the early stage of drug development.

Given the fact that the development of QT prolongation and TdP is associated with multiple genes [Moss, 2006], simulation studies of multiple ion channels blockade have been conducted by Mirams *et al.* [Mirams et al., 2011]. They collected multiple ion channels (hERG, I_{Na}, and I_{CaL}) data on 31 drugs associated with various risk of TdP, and performed simulations with a series of mathematical models of cardiac cells to integrate information on multi-channel block, resulting in improved prediction of TdP risk. Ouillé *et al.* [Ouillé et al., 2011] picked five TdP-associated ion channels, including three potassium channels (I_{KR}, I_{K1}, and I_{KS}), one sodium (I_{Na}) and one calcium (I_{CaL}) channel, and elucidated the mechanisms of drug-induced TdP using ion channel blocking profiles. They studied the effects of known torsadogenic and non-torsadogenic compounds on these ion channels, investigated their functions in the generation of drug-induced

TdP and QT prolongation, and identified that I_{Na} and I_{K1} play roles that are as important as I_{KR} in safety pharmacology. This study motivated us to use the blocking profiles of the five ion channels as the second view to improve our MVL models.

ℓ_2-norm co-regularization has been highly prevalent in MVL algorithms due to its ease of implementation, while our proposed ℓ_1-norm co-regularized MVL method can be more advantageous. To our best knowledge, this work is the first attempt to formulate the prediction of drug-induced QT prolongation as a MVL problem, integrate the complex relevant knowledge from two diverse domains, and build accurate models using limited and noisy available data. Finally, the rest of this chapter is organized as follows: in Section II we introduce our new MVL methods in more detail. Data collection, feature extraction, and all experimental and comparison results are described in Section III. We then conclude our work in Section IV with insightful observations.

7.3 Methods

In our data set, each sample has two view vectors. For labeled samples, the two view vectors are associated with the same label, while unlabeled samples have no label but can be helpful on improving model performance. In this section we discuss our proposed ℓ_1-norm co-regularized MVL algorithm in detail, including how we integrate information from all views and derive the problem formulation. We also derive the analytic form of the gradients of the objective functions for simultaneous optimization, as compared to the traditional alternate optimization technique.

7.3.1 ℓ_1-norm Co-regularized Multi-view Learning

Suppose we have V views and for each view $v = 1,2,...,V$, use $X_v \in R^{N\times M_v}$ to denote the feature matrix with N samples and M_v features on view v. Let x_i represents the concatenated row feature vector for sample $i = 1,2,...,N$. The N samples can be partitioned into two subsets L and U: a small number of labeled samples $\{(x_i,y_i)\}_{i\in L}$ in L, and a set of unlabeled sample $\{x_i\}_{i\in U}$ in U, where $|L| \ll |U|$ and the label $y_i \in \{-1,1\}$. The basic idea underlying co-regularized multi-view learning is to learn one mapping function h^v on each view v so that these functions agree each other on an unlabeled sample as much as possible while the error on the labeled samples is minimized. The final prediction function h is defined as:

$$h(x_i) = \frac{1}{V}\sum_{v=1}^{V} h^v(x_{i,v}), \tag{7.1}$$

where $x_{i,v}$ represents the feature vector in the v view for sample i. The view mapping functions h^1, h^2, ..., h^v are obtained by *minimizing* the following objective function over these view functions:

$$\begin{aligned} F(h^1,...,h^v) &= \sum_{i\in L} L(y_i, h(x_i)) + \sum_{v=1}^{V} \lambda_v ||h^v||_p \\ &\quad + \mu \sum_{v\neq v_1}^{V} ||h^v(X_v) - h^{v_1}(X_{v_1})||^2, \end{aligned} \tag{7.2}$$

where the first term is the loss function that penalizes the misclassification on labeled samples, the second term is to regularize the mapping function on each view with ℓ_p-norm ($1 \leq p \leq +\infty$) so that irregular solutions will be filtered out, and the final term is to penalize the disagreement among the prediction results from different view functions and to drive them to agree one another as much as possible. Here parameters $\{\lambda_j\}$ are the strength of ℓ_1-norm regularization terms in all views, and μ is the coupling parameter that regularizes the prediction disagreement using unlabeled data. By minimizing the sum of these three terms, co-regularization based MVL aims to identify a set of regular mapping functions that minimize misclassification rates on labeled samples and agree one and another on the predictions of unlabeled samples as much as possible.

For simplicity, without loss of generality we consider only two views and take least square loss on labeled samples and use ℓ_1-norm regularization on both views, resulting in the following simplified objective function,

$$\begin{aligned} F(h^1,h^2) &= \sum_{i\in L} ||y_i - h(x_i)||_2^2 + \lambda_1 ||h^1||_1 + \lambda_2 ||h^2||_1 \\ &\quad + \frac{1}{2}\mu \sum_{i\in U} ||h^1(x_i^1) - h^2(x_i^2)||_2^2, \end{aligned} \tag{7.3}$$

By applying a linear mapping function W_v on view $v = 1,2$, $h^v(X_v)$ = X_vW_v, where W_v is a $M_v \times 1$

column coefficient vector for view v, we have a metricized objective function as follows:

$$\begin{aligned} F(W_1,W_2) &= ||Y-\frac{1}{2}(X_1W_1+X_2W_2)||_2^2+\lambda_1||W_1||_1 \\ &\quad +\lambda_2||W_2||_1+\frac{1}{2}\mu||X_1W_1-X_2W_2||_2^2, \end{aligned} \tag{7.4}$$

If ℓ_2-norm co-regularization is used in Equation (7.4), this objective function is convex and smooth, and is usually optimized using the alternate optimization approach [Yu et al., 2007, Krishnapuram et al., 2004], i.e., alternately optimizing one view with the other view fixed until convergence. When we use ℓ_1-norm co-regularization as in Eqn.(7.4), the objective function becomes convex but not smooth. However, by reformulating this function properly we can optimize both views simultaneously. Let [.] denote the horizontal concatenation of two matrices $Z_1 \in R^{n\times p}$ and $Z_2 \in R^{n\times q}$, and then $[Z_1\ Z_2] \in R^{n\times(p+q)}$. We concatenate column vectors W_1,W_2 and matrices X_1,X_2 as $W = [W_1^T\ W_2^T]^T$, $X_L = [X_1\ X_2]_{x\in L}$, and $X_U = [X_1\ -X_2]_{x\in U}$. The objective function F in Eqn.(7.4) can be rewritten as

$$\begin{aligned} F(W) &= ||Y-\frac{1}{2}X_LW||^2+\lambda^T||W||_1+\frac{1}{2}||X_UW||^2, \\ &= (Y-\frac{1}{2}X_LW)^T(Y-\frac{1}{2}X_LW)+\lambda^T||W||_1 \\ &\quad +\frac{1}{2}\mu(X_UW)^TX_UW, \end{aligned} \tag{7.5}$$

where (λ) is a $(M_1+M_2)\times 1$ constant vector with the first M_1 elements as λ_1 and the remaining M_2 elements as λ_2. It is straightforward that we can derive the analytic form of the gradient of the function in Equation (7.5) as follows:

$$\begin{aligned} \frac{\partial F(W)}{\partial W} &= -X_L^T(Y-\frac{1}{2}X_LW)+\mu X_U^TX_UW \\ &\quad +\lambda * \text{sign}(W), \end{aligned} \tag{7.6}$$

We then use the LBFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno) method [Matthies & Strang, 1979, Nocedal, 1980] to minimize the objective function in Equation (7.5). With the known analytic form of the gradient as in Equation (7.6), the optimization process can be very efficient. Once the optimal value of the concatenated vector W_{opt} is obtained, the prediction on new samples with two views (x_1^{new}, x_2^{new}) is defined as

$$y_{new} = \text{sign}([x_1^{new}\ x_2^{new}]W_{opt}/2), \tag{7.7}$$

Finally the prediction accuracy, precision, and recall can be computed by comparing y_{new} with the ground truth.

7.3.2 ℓ_2-norm Co-regularized Multi-view Learning

The ℓ_2-norm co-regularized MVL method has been discussed in many previous work [Sindhwani & Niyogi, 2005,Sindhwani & Rosenberg, 2008]. The objective function of this MVL method is as follows,

$$\begin{aligned} F(W_1, W_2) &= ||Y - \frac{1}{2}(X_1W_1 + X_2W_2)||_2^2 + \lambda_1||W_1||_2 \\ &\quad +\lambda_2||W_2||_2 + \frac{1}{2}\mu||X_1W_1 - X_2W_2||_2^2, \end{aligned} \tag{7.8}$$

Some previous works use an alternate optimization method to identify optimal W_1 and W_2. The alternate process can be very time-consuming, however we can apply some reformulation tricks and implement it in a more efficient matter. Similar to ℓ_1-norm co-regularization, the gradient of this objective function can be written as,

$$\begin{aligned} \frac{\partial F(W)}{\partial W} &= -X_L^T(Y - \frac{1}{2}X_LW) + \mu X_U^T X_U W \\ &\quad + \frac{\lambda_1}{||W_{11}||_2}W_{11} + \frac{\lambda_2}{||W_{22}||_2}W_{22}, \end{aligned} \tag{7.9}$$

where variables W, Y, X_L, X_U, and U have the same meaning as in the previous subsection. W_{11} and W_{22} are vectors with the same length of W, and are obtained by appending 0's at the end of the original vector W_1 and at the beginning of W_2, respectively. With this gradient we can optimize the objective function in Equation (7.8) similarly as for our ℓ_1-norm co-regularized MVL method.

Table 7.1: Summary of data and results for the second view generation.

Ion channels	AID	Active	Inactive	# Used	Accuracy
I_{Na}	/	8	17	All	0.788
I_{CaL}	/	53	24	All	0.750
I_{kr}	376	250	1703	200	0.789
I_{ks}	2642	3878	301,738	200	0.778
I_{k1}	2032	926	1,338	200	0.739

7.4 Results

In this section, we began our discussion with an introduction to how we collected, cleaned, and represented the TdP-related drugs and those compounds with known binding activity to available human proteins, including the five TdP-associated ion-channel proteins. For performance comparison, three reference methods are selected as comparison baselines. SVMs [Burges, 1998] is a widely used single-view supervised learning algorithm with excellent performance. The LIBSVM [Chang & Lin, 2001] implementation and RBF kernels were used in all experiments. As a generalized linear method, ℓ_1-regularized logistic regression (LLR) [Koh et al., 2007] is another excellent single-view supervised learning method can construct a model that estimates probabilities, e.g. for medical diagnosis and credit scoring. In this chapter, we used the algorithm proposed by Boyd *et al.* [Koh et al., 2007] to run our experiments of the LLR method. The third reference method is the ℓ_2-norm co-regularized MVL method, which has been discussed in many previous works [Sindhwani & Niyogi, 2005, Sindhwani & Rosenberg, 2008] and introduced in the METHODS section.

7.4.1 Data set and feature extraction

We collected the compound set of drug-induced QT prolongation from the home page of University of Arizona CERT (`http://www.azcert.org/`) [Woosley, 2003] and the work by Ouillé *et. al* [Ouillé et al., 2011]. We merged these two data sets by removing redundant and/or inconsistent data points, resulting in a set of 28 drug compounds labeled as “TdP risk”, 40 labeled as “possible TdP risk”, and 74 with other less reliable evidence of QT prolongation. In our experiments we treated the 28 drugs with “TdP risk” labels as positive samples, the 40 drugs with "possible TdP risk" labels as positive or unlabeled samples, and the remaining 74 compounds as unlabeled samples whose labeled to be determined by our MVL models. For negative samples, we extracted all approved drugs from the DrugBank database [Wishart et al., 2006, Wishart et al., 2008], excluded all drug compounds that are associated with QT prolongation and/or TdP risk with even very weak evidence, and obtained a set of 1,221 drug compounds as putative negative samples.

We used molecular signature descriptors [Faulon et al., 2004] to extract features from chemical structures. In a molecular signature vector, each component is the number of occurrences of a particular atomic signature in the molecule. An atomic signature is a canonical representation of the substructure surrounding a particular atom, including all atoms and bonds up to a predefined number (called the signature height) of consecutive bonds from the given atom. We set the signature height at 2 for all compounds, filtered out all atomic signatures with frequency less than 4%, and finally obtained about 100 features for each compound. We used drug signature descriptors as the first view of our data set.

In our experimental study, we first randomly selected an equal number of negative and positive samples and combined them with unlabeled data to make a balanced subset, from which we randomly selected 20% positive and 20% negative samples for testing. We used the remaining 80% labeled and all unlabeled samples for training with 10-fold cross validation to select the best model parameters. For each experimental setting, we repeated the experiment for 50 times using random sampling, and reported the mean and standard deviation of testing accuracy, precision, and recall, which are defined as (TP+TN)/S, TP/(TP+TN), and TP/(TP+FN), respectively, where TP is true positive, TN is true negative, FN is false negative, and S is the total number of testing samples.

7.4.2 Generating the second view

In the MVL framework, each data sample has multiple feature vectors, one from each view. Our first view is taken from compound structures also using molecular signature descriptors [Faulon et al., 2004]. We set signature distance at 2 and frequency threshold at 4%, and ended up with a vector of $\sim$90 non-negative features for each compound. We first used the binding profiles of all drug compounds to the five ion-channel proteins as the second view. For the binding activity that are unknown or unavailable, we have built protein-chemical interaction prediction models to find appropriate decisions, resulting in a vector of five binary elements, in which a bit is set to 1 if a drug compound inhibits an ion channel, and 0 otherwise. To this end, we manually searched for binding compounds of the five picked ion channels from the PubChem database. For the potassium ion channels I_{KR}, I_{K1}, and I_{KS}, we identified multiple target-based bioassays that use them as target proteins, and then selected bioassays with AID 376, 2032 and 2642 for each of them, respectively. Each assay provides hundreds of active compounds and a large number of inactive compounds. For the calcium ion channel I_{CaL} and the sodium ion channel I_{Na}, we identified multiple related target-based assays with few tested compounds. We decided to use those tested compounds with activity less than 1μM as active and the rest as inactive, and obtained 8 (53) weakly active and 17 (24) inactive compounds for the ion channel I_{Na} (I_{CaL}).

We randomly selected 100 active and 100 inactive compounds for the potassium ion channels as a balanced data subset and used all active and inactive compounds identified for the sodium and calcium channel. The model construction and selection process is as follows: first 20% active and 20% inactive compounds were randomly selected as the testing set and the rest 80% as the training set, and 10-fold cross validation were applied to the training set with support vector machines (SVMs) and RBF kernels to build one model for each ion channel using its own binding compounds. In 10-fold cross validation, the training data was randomly split into 10 equal disjoint subsets. Nine subsets were used for training a model, and the rest subset was used as a validation set for validating the model. The cross-validation process repeated for 10 times with each subset used exactly once as the validation set. Each experiment was repeated for ten times, and a compound was considered to be active to an ion channel if it was predicted as active for six or more times. Eventually our models have an average accuracy of about 73-78%, as shown in Table 7.1.

Though informative, the binding profiles to the five ion-channel proteins has two limitations as the second view: the dimensionality of the resulting feature vectors is small and the feature values were obtained from predictions due to insufficient availability of experimental data. An reasonable alternative is to use the binding profiles of each drug compound to many human proteins as the second view. We obtained the protein binding profiles of the drug compounds from the STITCH [Kuhn et al., 2008, Kuhn et al., 2010] database, which integrates information about chemical-protein interactions from metabolic pathways, crystal structures, binding experiments, drug-target relationships, and text-mining results. Additional interactions in STITCH are extracted from other databases such as DrugBank [Wishart et al., 2006, Wishart et al., 2008]. Each proposed interaction is associated with a relevance score and can be traced back to the original data sources. We mapped the 1,363 drug compounds to the STITCH database and identified at least one interacting human proteins for 791 of them, and the total number of human proteins that are interacting with ten or more of the 791 drug compounds is 1,783. Hence the second view of a drug compound is represented by a binary vector of 1,783 elements, in which a bit is set on if the compound is interacting with the protein at the index, and is set off otherwise.

7.4.3 Model Construction and Selection

There are three model parameters to be optimized for the ℓ_1-norm and ℓ_2-norm co-regularized MVL methods in the training process: λ_1 and λ_2 are the strength of ℓ_1-norm regularization terms in the two views, and μ is the coupling parameter that regularizes the prediction disagreement using unlabeled data. These real-valued parameters allow flexible tradeoffs between different regularization terms, and thus finding their optimal values is crucial for model selection. Our algorithm takes four input parameters: the number of features m_1 in the first view, labeled samples X_L from two views and their labels Y, unlabeled samples X_U from both views, and outputs the mapping function W. Input parameter m_1, Y, X_L, and X_U are constant during the optimization process, while the three model parameters λ_1, λ_2 and μ are to be determined using grid search.

We used 10-fold cross validation along with an exponential grid search strategy to identify the optimal model parameters as described previously. First 20% labeled samples were randomly selected as testing data and the rest 80% was split into 10 equal subset, nine as the training set and the other as the validating set. The object function in Equation (7.5) was minimized regarding to the mapping function vector W on the training set for each given set of λ_1, λ_2, and μ values. The obtained optimal W was then applied to the validating set to calculate the validation accuracy with the given model parameters. We repeated this cross-validation process for 10 times and achieved the mean validation accuracy. Note that if the mean accuracy is less than 50%, we considered this training process was failed and a new process would be initiated. The set of model parameters (λ_1, λ_2, μ) that maximize the mean validation accuracy were selected as the best model parameters, which are applied to the unused testing set to compute the final testing accuracy, precision and recall.

7.4.4 Experimental Results

In this section, we compared our proposed ℓ_1-norm co-regularized MVL algorithm (ℓ_1-MVL) with SVMs, ℓ_1-regularized logistic regression (LLR), and the ℓ_2-norm co-regularized MVL method using both our proposed optimization (ℓ_2-MVL) and alternate optimization (ℓ_2-MVL-ALT) techniques, and evaluated their performance using testing accuracy, precision, and recall. Note that our results were obtained from 20 repeated experiments, so the performance differences between different methods should be considered significant. We tested these five methods at various experimental settings to investigate the contribution and influence of the related factors on prediction performance.

First we use the ion-channel binding profiles as the second view, and the data set consists of 28 positive, 40 possible positive, 74 unlabeled and 1,221 putative negative samples. If we used only 28 positive samples and treated the 40 possible positive samples as unlabeled, we then have 114 unlabeled samples. The results of the five methods were summarized in the first five rows in Table 7.2. We found that our ℓ_1-MVL method significantly outperformed not only the SVL methods (SVMs and LLR) for 5-6%, but also the ℓ_2-MVL method for about 3.3% on prediction accuracy. For precision, the ℓ_1-MVL method was slightly better than SVMs and LLR, and much better than ℓ_2-MVL methods. All MVL methods have much better recall than SVMs and LLR. All

Table 7.2: Prediction accuracy using ion-channel binding profiles. Standard deviations included in parenthesis.

		$\lvert X_L\rvert$ = 28				$\lvert X_L\rvert$ = 68		
Method	$\lvert X_u\rvert$	Accuracy(%)	Precision(%)	Recall(%)	$\lvert X_u\rvert$	Accuracy(%)	Precision(%)	Recall(%)
SVM	/	72.7(6.0)	77.9(9.6)	72.5(10.5)	/	68.7(4.9)	69.2(5.8)	69.6(3.9)
LLR	/	71.2(7.8)	77.9(10.5)	65.8(13.3)	/	67.6(7.4)	70.1(9.0)	65.7(8.4)
ℓ_1-MVL	114	**77.1(7.0)**	**80.2(8.5)**	**82.5(9.6)**	74	**71.6(3.8)**	70.7(5.1)	**72.3(5.4)**
ℓ_2-MVL	114	74.8(7.6)	73.9(7.7)	81.7(9.1)	74	69.3(6.2)	70.6(6.1)	68.6(7.4)
ℓ_2-MVL-ALT	114	74.4(7.5)	73.6(8.0)	80.5(7.6)	74	69.6(6.6)	69.7(5.9)	69.1(7.4)
ℓ_1-MVL	0	75.9(7.8)	78.9(7.3)	81.5(9.2)	0	69.8(6.1)	**71.1(5.5)**	69.6(6.6)
ℓ_2-MVL	0	73.1(6.1)	76.4(6.2)	78.1(11.8)	0	67.5(7.3)	68.0(6.5)	67.5(7.6)
ℓ_2-MVL-ALT	0	74.8(5.2)	77.6(7.4)	79.5(7.6)	0	68.7(6.5)	69.4(6.0)	68.9(5.7)

these performance metrics consistently reveal that our proposed ℓ_1-MVL method performs better than the reference methods. Moreover, to investigate the contribution of the unlabeled samples to prediction performance, we replaced the matrix X_U with a zero matrix of the same dimensionality, and performed similar experiments described above. We listed the results in the last three rows in Table 7.2. We observed apparent decrease of the ℓ_1-MVL method on accuracy and recall, while the precision results are still comparable or slightly better than previous experiments. We can approximately estimate the contribution of unlabeled samples in the the ℓ_1-MVL method is about 1.1%, which is non-trivial considering that the total accuracy improvement of our MVL method is only 5-6% compared to SVL methods. Although the ion-channel binding profiles has very small dimensionality as the second view, its contribution to the performance improvement is significant by comparing the results of the MVL methods with the SVL methods.

If we gave them more confidence on the 40 drug compounds labeled as "possible TdP risk"and used them as positive samples, we would have more labeled and less unlabeled samples. With 68 positive samples, we did similar experiments to investigate the influence of the number of labeled samples on prediction performance. All results with the same experimental settings as above are summarized in right columns in Table 7.2. By using those "possible positive samples" as positive samples, we observed significant performance decrease on all methods. Although the ℓ_1-norm co-regularized MVL method is still better than the baseline methods, the performance advantage shrank to 1-2%. By adding unreliable positive samples to our data set, we introduced many false positives and thus compromised the prediction accuracy. Moreover, the precision of the ℓ_1-MVL

Table 7.3: Prediction accuracy using general protein binding profiles. Standard deviations included in parenthesis.

		$\lvert X_L \rvert$	= 21			$\lvert X_L \rvert$	= 45	
Method	$\lvert X_u \rvert$	Accuracy(%)	Precision(%)	Recall(%)	$\lvert X_u \rvert$	Accuracy(%)	Precision(%)	Recall(%)
SVM	/	67.5(9.5)	69.8(11.1)	59.0(15.5)	/	64.2(5.3)	67.3(9.1)	57.2(9.2)
LLR	/	69.0(9.1)	71.6(11.0)	60.0(15.1)	/	61.9(5.6)	68.6(10.3)	49.4(11.8)
ℓ_1-MVL	71	**69.1(6.4)**	**73.0(8.7)**	**61.3(11.0)**	47	68.1(6.4)	**73.0(9.7)**	**61.3(10.8)**
ℓ_2-MVL	71	67.5(7.4)	70.1(9.0)	56.7(8.5)	47	67.5(11.4)	70.1(9.0)	56.7(10.5)
ℓ_2-MVL-ALT	71	64.7(7.3)	68.5(7.5)	56.7(7.4)	47	64.7(11.3)	68.5(8.4)	56.7(8.4)
ℓ_1-MVL	0	68.3(6.4)	72.6(5.7)	60.7(8.5)	0	**69.3(5.4)**	72.6(6.7)	60.7(8.5)
ℓ_2-MVL	0	65.0(5.7)	70.1(7.6)	54.4(6.0)	0	65.0(5.7)	70.1(8.6)	54.4(7.0)
ℓ_2-MVL-ALT	0	66.7(5.9)	71.0(8.8)	55.0(8.1)	0	66.7(7.0)	71.0(8.8)	55.0(8.1)

method decreased about 10%, which more clearly demonstrates that the data set contains much more false positives than before, according to the definition of precision. We also found that the precision of ℓ_2-MVL method dropped only about 3%. The observation showed that our proposed ℓ_1-MVL method is more sensitive to false positive samples. Finally, it is easy to notice that the performance of the MVL methods fluctuated very slightly when we have different numbers of unlabeled samples, apparently because the contribution of unlabeled samples have been offset by introducing more false positive samples.

As an independent view, the ion-channel binding profiles have very small dimensionality and most of the feature values were obtained from SVM predictions. We were then seeking to use the general protein binding profiles of the drug compounds as the second view, and the STITCH database provided valuable binding information for many human proteins. Through label mapping, the number of compounds labeled "TdP risk", "possible TdP risk", and other less reliable TdP evidence is 21, 24, and 47, respectively, and the number of negative compounds is 699. In exactly the same set of experiments as previously, we first used 21 positive, 71 unlabeled, and 21 randomly selected negative samples, and later used 45 positive, 47 unlabeled, and 45 randomly selected negative samples, and each experiment is repeated for 20 times. At the first step, 20% labeled samples were randomly selected as the testing set, and 10-fold cross validation was applied to the rest 80% labeled and all unlabeled samples for selecting optimal model parameters. We also tested the contribution of unlabeled samples by replacing them with a zero matrix of the same size in our methods, as shown in Table 7.2.

Surprisingly, the performance using this alternate second (Table 7.3) view is significantly worse than the ion-channel binding profiles in all experimental settings. One reason might be that the general protein binding profiles is not relevant to QT prolongation. Therefore, the ion-channel binding profiles provided significant relevant information about QT prolongation with just five features.

7.4.5 Time Complexity

Finally, there are totally three implementations of the ℓ_1-MVL and ℓ_2-MVL methods using simultaneous or alternate optimization, which alternately optimizes one view with the other view fixed iteratively until convergence of each mapping function. We used the same convex minimizer (LBFGS algorithm) and convergency threshold for objective functions and gradients from different MVL methods. In this section, we compared the time complexity of the ℓ_1-MVL and ℓ_2-MVL methods by varying the number of repeated experiments under identical programming and running environments. We ran all experiments using 28 positive and 114 unlabeled samples, and repeated for 1, 3, 5, 7, and 10 times on a multi-core cluster, and reported the CPU time used by each method, as plotted in Figure 7.1. From the curves of the ℓ_2-MVL, we found that the simultaneous optimization technique is 3$\sim$4-fold faster than alternate optimization. If we instead used alternate optimization to implement the ℓ_2-MVL, the ℓ_1-MVL is then about two-fold faster. When simultaneous optimization is applied to both MVL methods, the ℓ_1-MVL is two-fold slower than the ℓ_2-MVL method, maybe because the ℓ_1-MVL method has a non-smooth convex objective function.

7.4.6 Contribution of the Second View

In this section we discussed the contribution of the ion-channel protein binding profiles of the drug compounds as the second view to final models, since this view significantly improved the performance of our ℓ_1-norm co-regularized MVL method, comparing with the single-view learning methods such as SVM and LLR. In the final models, we extracted the model coefficients of these

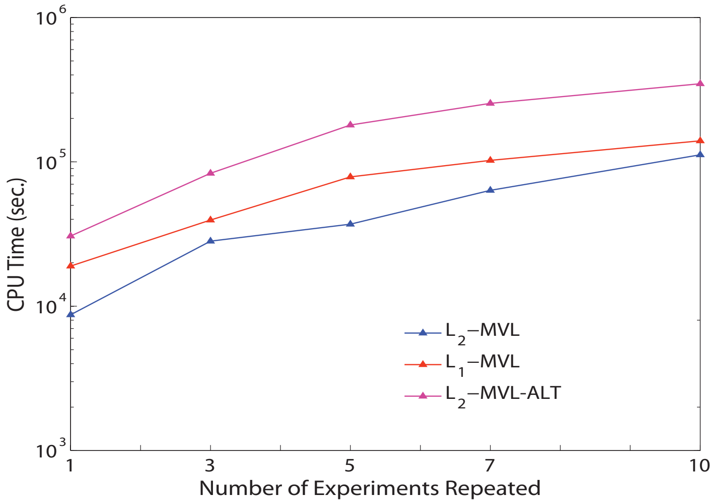

Figure 7.1: Computational time with more repeating experiments. "L_2-MVL-ALT" stands for the ℓ_2-norm MVL method implemented using alternate optimization.

five binary features from the second view, and calculated the mean and standard deviation of these coefficients under the four different experimental setting previously described. According to the fourth chapter in [Hastie et al., 2009], the mean to standard deviation ratio, called Z-score, of a feature's model coefficient measures the significance of this feature's contribution to the final models. From Table 7.4 we can find that the ion channel I_{K1} plays a relatively more important role since its Z-score $\in [0.92, 1.33]$ is much greater than other ion channels. This finding also partly matches the observation in [Ouillé et al., 2011], in which ion channel I_{Na} and I_{K1} are found to play a key role in the generation of drug-induced QT prolongation and TdP. In our models the importance of the ion channel I_{Na} was not identified, mainly because we have very few training samples (8 weak positives and 17 negatives) available for this ion channel in the data set. In addition, the Z-scores of I_{CaL} and I_{KS} are very close to zero, and can be reasonably considered as the evidence of the sparsity of our final models, i.e., two our of the five ion-channel feature was considered no significant relevance, and we can set the corresponding coefficients as zero.

Table 7.4: Z-scores of LLR model coefficients for the five ion-channel proteins.

$\lvert X_L \rvert$	$\lvert X_u \rvert$	I_{Na}	I_{CaL}	I_{KR}	I_{KS}	I_{K1}
28	114	0.308	0.061	0.300	0.082	1.327
28	0	-0.134	-0.077	0.042	-0.024	1.916
68	74	0.337	0.039	0.615	-0.090	0.923
68	0	0.434	-0.020	0.454	-0.097	0.968

7.5 Discussion

In this chapter we aim to construct accurate prediction models with very limited and noisy data that are available, and propose a novel ℓ_1-norm co-regularized multi-view learning algorithm on predicting drug-induced QT prolongation. By implementing our MVL method and the ℓ_2-norm counterpart with a reformulation trick and optimized mapping functions on all views simultaneously, we achieve 3-4 fold speed increase over the previously used alternate optimization technique [Yu et al., 2007, Krishnapuram et al., 2004], and result in more intuitive and sparse models. It is critical not to mess up our mixed regularization with the fused lasso penalty [Liu et al., 2010, Tibshirani et al., 2005] since they take place in different spaces, and have distinct working mechanisms. We conduct experiments under various settings to investigate the contribution and influence of the number of unlabeled samples and labeled samples. Experimental results show that our ℓ_1-MVL method outperforms not only two excellent SVL methods (SVMs and LLR), but also the widely used ℓ_2-norm co-regularized MVL method with the measurement of testing accuracy, precision, and recall.

To handle the challenge of limited and noisy labeled samples, we extract the second view for all TdP related drug compounds from critical related works, and pick five ion-channel proteins that have been found highly relevant to the QT prolongation effect. The contribution of the second view has been clearly demonstrated by our comprehensive experiments and significant higher prediction accuracy in both the ℓ_1-norm and ℓ_2-norm co-regularization framework. From the coefficients in the final models, we analyzed the significance of the contribution of each ion channel to QT prolongation, and our finding partly matched previous work [Ouillé et al., 2011]. We design an alternative as the second view using the general human protein binding profiles, whose dimensionality can be up to 1,783 though we use compress it to about 180 features. The experimental results are surprisingly worse than the previous five-dimension view, which reveal that the ion-channel view is more relevant.

Our ℓ_1-MVL method also has limitations. We observe that the variance of our experimental results is somehow remarkable from both Table 7.3 and 7.2, and hence the model stability of our MVL method is yet satisfactory. Due to the limited availability of ion-channel protein binding profiles, the five-dimension view is obtained from prediction. The models used to predict our second view data give prediction accuracy of 73-78%, which means that our second view data have 22-27% noise. Experiment results showed that when more false positives are introduced, the performance of our MVL method decreases dramatically. As more ion-channel binding data of drug compounds become available, our MVL method should expect even better performance.

Chapter 8

Inductive Multi-task Learning with Multiple View Data

In this chapter, we extend our MVMT learning algorithm to handle multi-view data with missing views and to model more complex task relationships. For data with missing views, the extended MVMT learning method learns a decision function only on each view present in a task. In addition, we use the task relationship learning technique to model multi-task data with more complex task relationships, where those tasks may be positively or negatively correlated, or even have no correlation. The simple ℓ_2–regularization technique will oversimplify the problem and distort the decision boundary. The extended MVMT learning methods can be applied to more applications. Our contribution in this chapter is on novel methodology development.

8.1 Introduction

In many real-world applications, it is becoming common to have data extracted from multiple diverse sources, known as "multi-view" data. Each data source is referred as a *view* [Culp et al., 2009]. Mining and learning with multi-view data (i.e. multi-view learning, or MVL) has been studied extensively [Blum & Mitchell, 1998,McCallum et al., 2000,Sindhwani & Niyogi, 2005,Sindhwani & Rosenberg, 2008,Amini et al., 2009,Culp et al., 2009]. For instance, in multi-lingual text categorization [Amini et al., 2009], each language represents a view. For scientific document categorization, each document has two views: the bag-of-words features and their citations [McCallum et al., 2000]. In classifying webpages [Blum & Mitchell, 1998], we may have three views for a given webpage: the content of the webpage, the text of any webpages linking to this webpage, and the link structure of all linked pages. The limitation of these MVL methods is that they essentially learn a single task individually at a time.

In this chapter, we study a new direction of multi-view learning where there are multiple related tasks with multi-view data (i.e. multi-view multi-task learning, or MVMT Learning). MVMT Learning has many real-world applications. For instance, sentiment classification for music reviews and news comments is two related tasks, and they both share word features from the comments or reviews. In protein functional classification, each protein has features from multiple views (e.g. protein sequences, 3D structures) and may be associated with multiple functional classes. In image annotation [Boutell et al., 2004], each image has features extracted from multiple sources and can be annotated with multiple objects such as a boat, a bird and etc. The interplay of multiple views and multiple tasks in the same learning problem motivates us to investigate multi-task learning in the multi-view framework.

Despite the wide application areas, multi-task learning (MTL) with multi-view data only caught the attention of the research community recently. Cavallanti *et al.* [Cavallanti et al., 2010] developed linear algorithms for online multi-task learning. Though there may be multiple views, each task has only one view in their settings. He *et al.* [He & Lawrence, 2011] proposed a graph-based iterative algorithm (IteM^2) for multi-view multi-task learning with applications to text classification. The IteM^2 algorithm projects any two tasks to a new Reproducing Kernel Hilbert Space (RKHS) based on the common views shared by the given two tasks. Though impressive preliminary results have been obtained in text categorization applications, the treatment of MVMT learning in IteM^2 is limited. First IteM^2 is a transductive learning method, hence it is unable to generate predictive models on independent, unknown testing samples. A consequence of the transductive learning is that the training data set must have the same fraction of positive samples as in the testing data set.

In addition, the method is designed primarily for text categorization where the feature values are all non-negative [1], since otherwise Proposition 4.1 and Theorem 4.2 in [He & Lawrence, 2011] will not hold. There are many real-world data sets with a significant portion of features with negative values that the IteM^2 method is unable to handle. In this chapter IteM^2 is the most important competing method that our proposed MVMT method will compare with.

We propose an inductive learning framework to address the general MVMT learning problem. Our starting point is a MVL framework based on co-regularization where we train multiple classifiers, one for each view, with the goal to obtain those classifiers that are in-agreement with one another on the unlabeled samples and achieve minimal classification errors on the labeled samples simultaneously [Sindhwani & Niyogi, 2005, Sindhwani & Rosenberg, 2008]. We add a couple of regularization factors (e.g. ℓ_2 regularization [Evgeniou & Pontil, 2004]) to ensure the functions that we learn from each task are similar to each other, resulting in a convex objective function, which makes the subsequent optimization easy. In addition, our algorithm is flexible when one view is completely missing for a task. In many application we may have structured view missing, and our algorithm extends to these cases naturally with the regularization function that we use. Moreover, it is quite often that not all tasks in multi-task learning are uniformly related to each other [Chen et al., 2010b, Kim & Xing, 2010]. To handle this, we introduce a positive semi-definite matrix to capture the relationship of tasks that may not be uniformly related to each other.

In summary, the main contributions of this work are two-fold. First we propose a general inductive learning framework for the challenging multi-view multi-task problems using co-regularization and task relationship learning. Secondly we design efficient algorithms to optimize the objective function with either close-form solutions or iterative optimization solutions. We conducted comprehensive experimental evaluations of our MVMT methods on three real-world MVMT data sets, and compared our MVMT methods with the state-of-the-art methods, including the competing IteM^2 method.

The rest of this chapter is organized as follows. In Section 2, we briefly review related work

[1] Refer to page 3 in He *et al.* [He & Lawrence, 2011]

on multi-view learning and multi-task learning. We explain our co-regularized MVMT algorithms and their implementations in details in Section 3. We have experimentally evaluated our methods on three real-world data sets and compared our results with those from the state-of-the-art methods. We present the results in Section 4 and conclude our work in Section 5.

8.2 Related Work

Multi-view semi-supervised learning has attracted significant research interest in recent years [Dasgupta et al., 2001, Abney, 2002, Muslea et al., 2002, Balcan & Blum, 2005, Wang & Zhou, 2007, Christoudias et al., 2008, Wang & FitzGerald, 2010]. The underlying assumption of multi-view algorithm is that each view is conditionally independent from other views [Blum & Mitchell, 1998] and is sufficient for constructing a predictive model [Amini et al., 2009, Dasgupta et al., 2001]. We briefly overview MVL methods that are widely used. Based on how information from multiple views is integrated, existing MVL algorithms can be roughly classified into a variety of categories. Co-training [Blum & Mitchell, 1998] iteratively labels unlabeled samples using the models built from existing labeled samples, and expands the pool of labeled samples until convergence of performance [Dasgupta et al., 2001, Nigam & Ghani, 2000]. Manifold co-regularization [Sindhwani & Rosenberg, 2008] was proposed based on a reproducing kernel Hilbert space with a data-dependent co-regularization norm to explore the structure of unlabeled multi-view data. Recently co-regularization [Sindhwani & Niyogi, 2005] attracted the attention of the community due to its simplicity of optimizing a single regularized objective function.

When each task has only a limited number of samples, multi-task learning has been empirically as well as theoretically shown to provide better predictive models with closely related tasks [Ando & Zhang, 2005, Argyriou et al., 2006]. Some early MTL approaches assumed that either the function parameters of different tasks are similar [Evgeniou & Pontil, 2004] or multiple tasks share a subset of features [Argyriou et al., 2006]. These MTL methods imposed a regularization term to enforce the difference between multiple task functions to be small. Recent studies on MTL

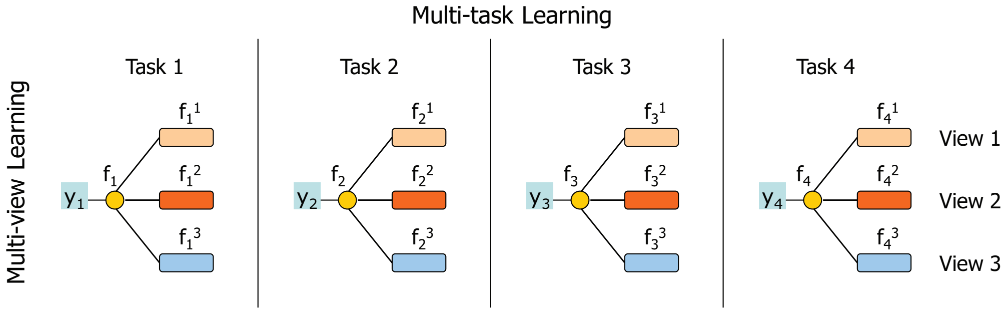

Figure 8.1: Graphical representation of the multi-view multi-task learning framework.

proposed that the relatedness of multiple tasks has a structure such as a graph or a tree [Kim & Xing, 2010, Chen et al., 2010b], or different tasks share a common subspace representation [Chen et al., 2010a]. Another group of MTL methods [Zhang et al., 2010] pose no assumption on the structure of task relatedness, and learn task relationship automatically from the input data, and provide more modeling flexibility.

8.3 Inductive Multi-view Multi-task Learning

In this section, we propose a general inductive learning framework for multi-task learning with multiple view data. We apply co-regularized MVL within each task, and the multiple related tasks are learned jointly using task regularization or task relationship learning. We handle the special case that some entire views are missing from some tasks, analyze the complexity of our MVMT methods, and develop efficient optimization algorithms to achieve optimal solutions.

8.3.1 Notations

In this chapter, we use bold capital letters (e.g. $\mathbf{X}$) to represent a matrix and bold lowercase letters (e.g. $\mathbf{x}$) to denote a vector. Lowercase letters (e.g. x) are used for scalars, and Greek letters (e.g λ) for regularization parameters. We use $[m : n]$ $(n > m)$ to denote a set of integers in the range of m to n inclusively. Unless stated otherwise, all vectors are column vectors.

Suppose we are given a set of N labeled and M unlabeled data samples for T tasks. In general we have limited supply of labeled samples but abundant supply of unlabeled samples, i.e. $M \gg N$. We use N_t and M_t denote the number of labeled and unlabeled samples in task $t \in [1:T]$, thus we have $N = \sum_t N_t$ and $M = \sum_t M_t$. Each sample has features from V views and the total number of features from all the V views is denoted as D. Let D_v be the number of features in the view $v \in [1:V]$, and we have $D = \sum_v D_v$.

For each view v present in task t, the feature matrix of the labeled samples is $\mathbf{X}_t^v \in \mathbb{R}^{N_t \times D_v}$. The feature matrix of the unlabeled samples is $\mathbf{U}_t^v \in \mathbb{R}^{M_t \times D_v}$. Let $\mathbf{y}_t \in \{1,-1\}^{N_t \times 1}$ be the label vector of the labeled samples in the task t. We write $\mathbf{X}_t = (\mathbf{X}_t^1, \mathbf{X}_t^2, \ldots, \mathbf{X}_t^V)$, and $\mathbf{U}_t = (\mathbf{U}_t^1, \mathbf{U}_t^2, \ldots, \mathbf{U}_t^V)$, corresponding to the concatenated feature matrix of the labeled and unlabeled samples for task t, respectively. It is common that in some applications not all tasks have features available from all the V views, so we introduce an indicator matrix $\mathbf{I}_d \in \{1,0\}^{T \times V}$ to mark which view is missing from which task, i.e. $\mathbf{I}_d(t,v) = 0$ if the view v is missing from task t, and = 1 otherwise. Using this notation, we only handle "structured" missing views in the sense that if a view is present in a task, it is present in all the samples in the task; if a view is missing from a task, it is missing in all the samples in the task. Throughout the chapter we use subscripts to denote tasks and superscripts to denote views.

8.3.2 Problem and Algorithm Overview

We illustrate the problem of learning multiple related tasks with multi-view data in Figure 8.1. We hypothesize that we can construct better classification models by considering information from multiple views and learning multiple related tasks jointly. In our method we learn one linear mapping function $\mathbf{f}_t^v : \mathbb{R}^{D_v} \to \{1,-1\}$ for each view v present in the task t, and search for those mapping functions based on two intuitions: (1) for a given task t, we expect that the mapping functions $\mathbf{f}_t^v$'s from all its views agree with one another as much as possible on the unlabeled samples, and (2) for a given view, we expect that all the mapping functions in different tasks behave similarly. We formalize the two intuitions using regularization functions in a supervised learning framework, and describe the details of the learning framework below.

8.3.3 Co-regularized Multi-view Learning

The basic assumption underlying multi-view learning for a single task is that the multiple views are conditionally independent and each view generates a predictive model that can be used to make predictions on data samples, while the final models are obtained from these view models. Without prior knowledge on which view contributes more to the final models than other views, we assume that all views contribute equally, following [Sindhwani & Niyogi, 2005, Sindhwani & Rosenberg, 2008]. The final models are obtained by averaging the prediction results from all view functions as follows:

$$f(\mathbf{x}) = \frac{1}{V}\sum_{v=1}^{V} f^{v}(\mathbf{x}^{v}), \tag{8.1}$$

where $\mathbf{x}$ has totally V views, $\mathbf{x}^v$ is the set of features for view v, and f^v is the view function generated on view v.

In multi-view learning, we want the models built on each single view to agree with one another as much as possible on unlabeled samples. Co-regularization is a technique to enforce such model agreement on unlabeled samples. The view functions $\mathbf{f}^v$ for all views v's are obtained from the following objective function,

$$\min_{f^v} L(\mathbf{y}, f(\mathbf{X})) + \frac{\lambda}{2}\|f\|^2 + \frac{\mu}{2}\sum_{v' \neq v} \|f^{v'}(\mathbf{U}^{v'}) - f^{v}(\mathbf{U}^{v})\|^2, \tag{8.2}$$

where $\|.\|$ denotes the ℓ_2 norm. $L(.,.)$ is the loss function that penalizes the misclassification on labeled samples. $f^v(\mathbf{U}^v)$ is the prediction results by applying the function f^v to each sample in the unlabeled data for the view v. λ is the parameter that regulates the strength of the ℓ_2-norm regularization on view functions, and μ is the coupling parameter that regularizes the disagreement of different views. By minimizing the three terms jointly, an optimal set of view functions that minimize the misclassification of labeled samples and maximize the agreement on the prediction results on unlabeled samples can be identified.

8.3.4 Task Regularization in MVMT Learning (*regMVMT*)

Given multiple related tasks with multi-view data, it is advantageous to learn these tasks in the same framework to achieve the benefits of both multi-task learning and multi-view learning. One approach to extend the co-regularized MVL framework is to learn each of the multiple task individually, as presented below.

$$\begin{aligned} \min_{f_t^v} \quad & \sum_t L(\mathbf{y_t}, f_t(\mathbf{X_t})) + \frac{\lambda}{2}\|f_t\|^2 + \\ & \frac{\mu}{2}\sum_{v' \neq v} \|f_t^v(\mathbf{U}_t^v) - f_t^{v'}(\mathbf{U}_t^{v'})\|^2, \end{aligned} \tag{8.3}$$

where f_t^v is a view specific mapping function for the view v in the task t.

Apparently the formula does not take advantage of the presence of multiple related tasks. In order to get benefit from the additional information, we apply an additional regularization function that penalizes the difference of the view specific functions on the same view across different tasks. For each view $v \in [1:V]$ in task $t \in [1:T]$, we learn a linear mapping function indexed by a parameter $\mathbf{w}_t^v \in \mathbb{R}^{D_v \times 1}$. We denote $\mathbf{w}_t \in \mathbb{R}^{D \times 1}$ as the column vector by concatenating all $\mathbf{w}_t^v$ for the task t, and we have:

$$\begin{aligned} f_t(\mathbf{X}_t) &= \frac{1}{V}\sum_{v=1}^{V} f^v(x^v) \\ &= \frac{1}{V}\sum_{v=1}^{V} \mathbf{X}_t^v \mathbf{w}_t^v = \frac{\mathbf{X}_t \mathbf{w}_t}{V}. \end{aligned} \tag{8.4}$$

Using the least square loss function, we have the objective function for the T tasks with V views for each task as follows:

$$\begin{aligned} \min_{\{\mathbf{w}_t^v\}} \quad & \sum_{t=1}^{T} \frac{1}{2}\|\mathbf{y}_t - \frac{\mathbf{X}_t \mathbf{w}_t}{V}\|^2 + \frac{\mu}{2}\sum_{v' \neq v}^{V} \|\mathbf{U}_t^v \mathbf{w}_t^v - \mathbf{U}_t^{v'} \mathbf{w}_t^{v'}\|^2 \\ & + \frac{\lambda}{2}\sum_{v=1}^{V} \|\mathbf{w}_t^v\|^2 + \frac{\gamma}{2}\sum_{t' \neq t}^{T} \|\mathbf{w}_t^v - \mathbf{w}_{t'}^v\|^2, \end{aligned} \tag{8.5}$$

where w_t^v, w_t, λ, and μ were explained before. γ is a new regularization parameter to penalize the difference of view mapping function across different tasks for the same view. We call this formalization "the regularized multi-view multi-task learning" (*regMVMT*).

To solve the related optimization problem of *regMVMT*, we denote the objective function as F, and compute its derivative regarding to each $\mathbf{w}_t^v$ as follows:

$$\begin{aligned}\frac{\partial F}{\partial \mathbf{w}_t^v} &= \frac{1}{V}\mathbf{X}_t^{vT}(\frac{\mathbf{X}_t\mathbf{w}_t}{V} - \mathbf{y}_t) + \gamma\sum_{t'\neq t}^{T}(\mathbf{w}_t^v - \mathbf{w}_{t'}^v) \\ &+\lambda\mathbf{w}_t^v + \mu\mathbf{U}_t^{vT}\sum_{v'\neq v}^{V}(\mathbf{U}_t^v\mathbf{w}_t^v - \mathbf{U}_t^{v'}\mathbf{w}_t^{v'}). \end{aligned} \quad (8.6)$$

Set Eq.(8.6) to zero, rearrange the terms and we have:

$$\begin{aligned} E_{tv} &= A_{tv}\mathbf{w}_t^v + \sum_{v'\neq v} B_{vv'}^t\mathbf{w}_t^{v'} + \sum_{t'\neq t} C_{t'v}\mathbf{w}_{t'}^v, \\ A_{tv} &= \lambda + \gamma(T-1) + \mu(V-1)\mathbf{U}_t^{vT}\mathbf{U}_t^v + \frac{\mathbf{X}_t^{vT}\mathbf{X}_t^v}{V^2}, \\ B_{vv'}^t &= \frac{\mathbf{X}_t^{vT}\mathbf{X}_t^{v'}}{V^2} - \mu\ \mathbf{U}_t^{vT}\mathbf{U}_t^{v'}, \\ C_{t'v} &= -\gamma I_{D_v}, \quad E_{tv} = \frac{\mathbf{X}_t^{vT}\mathbf{y}_t}{V}, \end{aligned} \quad (8.7)$$

where I_{D_v} is a $D_v \times D_v$ identity matrix. Note that we can have such an equation for each view v in the task t. To learn $\mathbf{w}_t^v$, we must also learn $\mathbf{w}_t^{v'}$ from other views v' ($v' \neq v$) in the task t and $\mathbf{w}_{t'}^v$ from other tasks $t' \neq t$ on the view v. Eventually we have to learn all $\mathbf{w}_t^v$'s jointly from a large set of equations as in Eq.(8.7) by taking derivatives regarding to each view in each task, as in the following linear equation system:

$$\mathscr{L}\mathscr{W} = \mathscr{R}, \quad (8.8)$$

where $\mathscr{L} \in \mathbb{R}^{TD\times TD}$ is a sparse block matrix with $TV \times TV$ blocks. Each block corresponds to a view in a task and its size is the feature dimensionality of the view.

Matrix $\mathscr{L}$ has the following form:

$$\begin{bmatrix}
A_{11} & B_{12}^1 & ... & B_{1V}^1 & \cdot\cdot\cdot & C_{T1} & 0 & ... & 0 \\
B_{21}^1 & A_{12} & ... & B_{2V}^1 & \cdot\cdot\cdot & 0 & C_{T2} & ... & 0 \\
... & ... & ... & ... & \cdot\cdot\cdot & ... & ... & ... & ... \\
B_{V1}^1 & B_{V2}^1 & ... & A_{1V} & \cdot\cdot\cdot & 0 & 0 & ... & C_{TV} \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
C_{T1} & 0 & ... & 0 & \cdot\cdot\cdot & A_{T1} & B_{12}^T & ... & B_{1V}^T \\
0 & C_{T2} & ... & 0 & \cdot\cdot\cdot & B_{21}^T & A_{T2} & ... & B_{2V}^T \\
... & ... & ... & ... & \cdot\cdot\cdot & ... & ... & ... & ... \\
0 & 0 & ... & C_{TV} & \cdot\cdot\cdot & B_{V1}^T & B_{V2}^T & ... & A_{TV}
\end{bmatrix}$$

$$\begin{aligned}
\mathscr{W} &= \mathrm{Vec}([W_1^1,...,W_1^V,E_2^1,...,E_2^V,...,E_T^1,...,E_T^V]), \\
\mathscr{R} &= \mathrm{Vec}([E_{11},...,E_{1V},E_{21},...,E_{2V},...,E_{T1},...,E_{TV}]),
\end{aligned} \tag{8.9}$$

where Vec() denotes the function stacking the column vectors in a matrix to a single column vector. Here column vector $\mathscr{W}$ and $\mathscr{R}$ are generated by stacking all column vectors $\mathbf{w}_t^v$ and E_t^v, respectively. Note that $\mathscr{L}$ is a sparse block matrix with $TV \times TV$ blocks, and block matrix A_{tv}, $B_{vv'}^t$, and $C_{t'v}$ are defined in Eq.(8.7). The analytic solution of $\mathscr{W}$ can be easily obtained from Eq.(8.8) by taking the inverse of matrix $\mathscr{L}$. When there is a new data sample from task t with $\mathbf{x}_t^*$ as the concatenated row feature vector of all the V view, the predictive outputs $\mathbf{y}_t^*$ is given by:

$$y_t^* = \mathrm{sign}(\mathbf{x}_t^* \mathbf{w}_t). \tag{8.10}$$

Note that by setting $\gamma = 0$ the *regMVMT* algorithm degenerates to the co-regularized multi-view learning algorithm. Similarly set $\mu = 0$ the *regMVMT* algorithm degenerates to the regularized multi-task learning algorithm. These two methods are special cases of our MVMT method when there is only one task or one view for each task, and we will implement them as comparison baseline methods. In Figure 8.2 we show a comparison between the *regMVMT* method and the competing transductive IteM^2 method on a data set for webpage classification. Here we could see that *regMVMT* improves significantly upon the IteM^2 method. Detailed experimental studies are in Section 4.

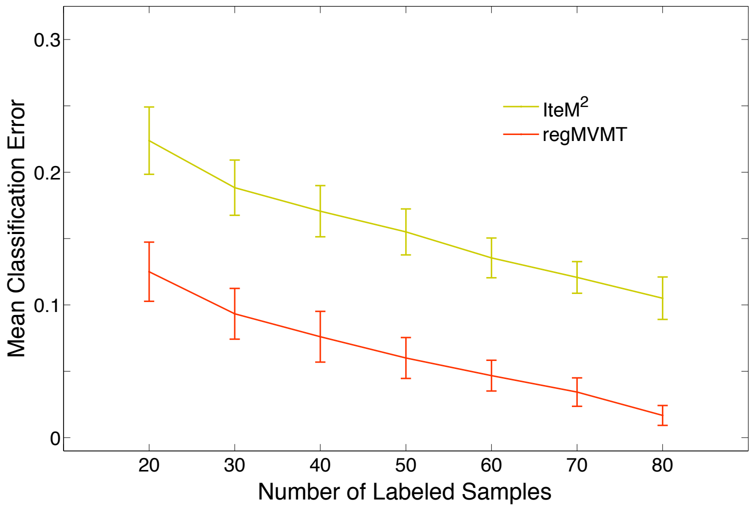

Figure 8.2: Preliminary results on the WebKB data set.

8.3.5 Learning Task Relationships in the MVMT Framework (*regMVMT+*)

In many MTL applications, tasks may not be uniformly related to each other. A remedy to those situation, as we investigate here, is to introduce a task relationship inference component [Zhang et al., 2010]. The basic idea is to use a positive semi-definite $\Omega \in \mathbb{R}^{T \times T}$ to model the similarity among T related tasks, and hence Ω must be with finite complexity. For single-view MTL problem, we may utilize Ω in the following objective function:

$$\begin{aligned} \min_{\mathbf{W},\Omega} \quad & \sum_{t}^{T} \frac{1}{2} \|\mathbf{y}_t - \mathbf{X}_t \mathbf{w}_t\|^2 + \frac{\lambda}{2} \|\mathbf{W}\|_F^2 + \frac{\gamma}{2} \mathrm{tr}(\mathbf{W}\Omega^{-1}\mathbf{W}^T), \\ \text{s.t.} \quad & \Omega \succeq 0, \ \ \mathrm{tr}(\Omega) = 1, \end{aligned} \tag{8.11}$$

where $\mathbf{W}$ is a matrix whose t_{th} column is $\mathbf{w}_t$ for task t. $\|.\|_F$ is the matrix Frobenius norm. $tr()$ is the trace of a matrix.

When we have V views for the T task, we learn a task similarity matrix Ω_v for each view v of the T tasks, and the MVMT objective function is as follows:

$$\begin{aligned}
\min_{\mathbf{W},\Omega} \quad & \sum_{t}^{T} \frac{1}{2}||\mathbf{y}_t - \mathbf{X}_t\mathbf{w}_t||^2 + \frac{\mu}{2}\sum_{v'>v}^{V} ||\mathbf{U}_t^v\mathbf{w}_t^v - \mathbf{U}_t^{v'}\mathbf{w}_t^{v'}||^2 \\
& + \frac{\lambda}{2}\sum_{v=1}^{V} ||\mathbf{w}_t^v||^2 + \frac{\gamma}{2}\sum_{v=1}^{V} \mathrm{tr}(\mathbf{W}^v\Omega_v^{-1}\mathbf{W}^{vT}), \\
\mathrm{s.t.} \quad & \Omega_v \succeq 0, \;\; \mathrm{tr}(\Omega_v) = 1, v \in [1:V],
\end{aligned} \tag{8.12}$$

where $\mathbf{w}_t$ is a column vector concatenated from $\mathbf{w}_t^v$'s for all $v \in [1:V]$, and $\mathbf{W}^v$ is a matrix whose t_{th} column is $\mathbf{w}_t^v$ for all $t \in [1:T]$. The inverse of Ω_v is a $T \times T$ matrix whose element at (i,j) is the similarity between task i and j.

Though the objective function is convex regarding to both $\{\mathbf{w}_t\}$ and $\{\Omega_v\}$, simultaneous optimization of them is technically and computationally challenging. Alternate optimization of $\mathbf{w}_t$'s and Ω_v's individually can achieve the optimal solutions efficiently. If we fix the $\{\mathbf{w}_t\}$'s and optimize the $\{\Omega_v\}$, take the partial derivative of the objective function in Eq.(8.12) with regard to $\{\Omega_v\}$, and set it to zero. The analytic solution to $\{\Omega_v\}$ is:

$$\Omega_v = \frac{(\mathbf{W}^{vT}\mathbf{W}^v)^{\frac{1}{2}}}{\mathrm{tr}((\mathbf{W}^{vT}\mathbf{W}^v)^{\frac{1}{2}})}. \tag{8.13}$$

Once Ω_v is learned, we can use the same algorithm that we have derived in Eq. 8.7. We find that only the block matrix A_{tv} and $C_{t'v}$ are different from the formulas in Eq.(8.7), while the formulas of matrix $B_{vv'}^t$ and E_{tv} remain unchanged. The new A_{tv} and C_{tv} are as follows:

$$\begin{aligned}
A_{tv} &= \lambda + \mu(V-1)\mathbf{U}_t^{vT}\mathbf{U}_t^v + \frac{\mathbf{X}_t^{vT}\mathbf{X}_t^v}{V^2}, \\
C_{t'v} &= \gamma c_{tt'}^v I_{D_v},
\end{aligned} \tag{8.14}$$

where c_{ij}^v denote the element (i,j) of the inverse of matrix Ω_v, and it is a scalar constant in each iteration. We denote this approach *regMVMT+* since it is a variant extension of the *regMVMT* method.

8.3.6 Dealing with Missing-view Data

In the previous subsections, we consider the ideal case that all tasks in a data set have complete data. When we have incomplete data, the MVMT learning problem becomes more challenging. Missing value imputation has been widely discussed, and here we are not concerned about randomly missed feature values. We aim to handel the case of "structured" missing views. In the context of this discussion, we focus on completely missing views for a task in the sense that if a view is present in a task, it is present in all the samples in the task; if a view is missing from a task, it is missing in all the samples in the task. We recognize that there is a more challenging case where we have partially observed views (i.e. some views are missing from some samples in a task). Partially observed views is beyond the scope of this chapter. A straightforward strategy to handel structured missing views is to remove any tasks with missing views, which, however, will significantly reduce the number of related tasks available and also discard useful information present in the remaining views of those tasks.

To handel structured missing views, we introduce an indicator matrix $\mathbf{I}_d \in 1,0^{T\times V}$ to mark the missing views for the T tasks with a total of V views for each task, i.e. $\mathbf{I}_d(t,v) = I_{tv} = 0$ if the view $v \in [1:V]$ is missing in the task $t \in [1:T]$, and $I_{tv} = 1$ otherwise. Let $V_t \leq V$ and $T_v \leq T$ denote the real number of views in task t and the number of tasks for view v, and we need to use V_t and T_v to replace V and T in all the above equations, respectively. In addition, the indicator scalars $I_{tv}, I_{t'v}$, and $I_{tv'}$ must be associated with the corresponding matrices from $\mathbf{X}$ and $\mathbf{U}$. In matrix Ω_v, a task is considered uncorrelated to any other tasks in terms of view v if it does not has the view v, and the dimension of Ω_v will be reduced by one row and one column. Though Ω_v may have different dimensionality for different v, we learn each Ω_v individually and we only need the trace of the resulting product matrix.

If view v is missing from task t, $\mathbf{w}_t^v$ in $\mathscr{W}$, $\mathbf{E}_t^v$ in $\mathscr{R}$, and the $((t-1)V+v)$-th block row and block column in matrix $\mathscr{L}$ are all-zero matrices. After removing these all-zero blocks from $\mathscr{L}$, $\mathscr{W}$, and $\mathscr{R}$, we have their compact versions $\mathscr{L}'$, $\mathscr{W}'$, and $\mathscr{R}'$, respectively. Eq.(8.8) is converted to:

$$\mathscr{L}'\mathscr{W}' = \mathscr{R}'. \tag{8.15}$$

Solving this equation is similar to Eq.(8.8).

8.3.7 Analysis of the *regMVMT* Algorithm

The objective function in Eq.(8.5) is convex regarding to $\mathbf{w}_t$'s, hence it has a global minimum. From the derivation of the matrix $\mathscr{L}$, we can tell that it is symmetric. Since matrix $\mathscr{L}$ is obtained from the derivative of a convex function, it must be at least positive semi-definite. If there is no all-zero row or column in matrix $\mathbf{I}_d$, i.e. each task has at least one view and each view present in at least one task, matrix $\mathscr{L}'$ is also positive semi-definite.

The implementation of the *regMVMT* algorithm is straightforward with the analytic solution. We first calculate the block matrices $A_t^v, B_{vv'}^t, C_t^v$, and the column vector E_t^v for each view v present in each task t, construct the large sparse square matrix $\mathscr{L}$ and the long column vector $\mathscr{R}$ according to the equations in Eq.(8.9). We then easily obtain the solutions $\mathbf{w}_t$'s as in Eq.(8.8) by computing the inverse of matrix $\mathscr{L}$ for a given set of regularization parameters. The parameters λ, μ, and γ will be optimized using standard cross validation. Note that the first column of each input feature matrix is an extra added unit vector to offset the intercepts (b) of linear functions. Since the intercepts are not included in any regularization terms, an identity matrix with the first diagonal element as 0 is multiplied with the constant terms ($\lambda + \gamma(T-1)$ and $-\gamma$) in Eq.(8.7) when constructing the matrix A_{tv} and C_{tv}.

Let D_T denote the number of rows or columns in $\mathscr{L}$, and we have $D_T = \sum_{t,v=1}^{T,V} I_{tv} D_v$, where D_v is the number of features in view v. When there is no missing view in any task, we have $D_T = T \times D$. Though the process of constructing square matrix $\mathscr{L}$ is complex, the speed-limiting step of the *regMVMT* algorithm is the inversion of the large sparse matrix $\mathscr{L}$ in Eq.(8.9). The construction of matrix $\mathscr{L}$ has a time complexity of $\mathbf{O}(T(T-1)D^2 + 2(N+M)D^2)$, and computing its inverse matrix is of complexity $\mathbf{O}(D_T^3)$, so the total number of features instead of the number of data samples dominates the time complexity. Matrix $\mathscr{L}$ is highly sparse, symmetric, and positive semi-definite, and the overall time complexity of the *regMVMT* algorithm is dependent on its sparsity structure.

Lemma 8.3.1. *The time complexity of the* regMVMT *algorithm is* $\mathbf{O}((1+2\bar{n}/T)D_T^2 + \frac{1}{3}D_T^3)$, *where* $\bar{n}$ *is the average number of samples in each task,* $D_T = \sum_{t,v=1}^{T,V} I_{tv} D_v$.

Proof. It is straightforward that constructing matrix $\mathscr{L}$ and vector $\mathscr{R}$ has time complexity of $\mathbf{O}((1+2\bar{n}/T)D_T^2)$, and the inversion of the positive semi-definite matrix $\mathscr{L}$ has time complexity of $\mathbf{O}(\frac{1}{3}D_T^3)$ using Chomsky decomposition. Hence we have the described overall time complexity. □

The constant factor hidden behind the asymptotic complexity depends on the sparsity structure of matrix $\mathscr{L}$. Since D_T is generally much greater than T, V, N, and M, the running time is determined by the total number of features D_T from all views present in all tasks. In addition, when we store matrices in sparse matrix format, the space complexity of our algorithm is dependent on the number of non-zero elements in matrix $\mathscr{L}$ and the space used by other much small matrices is eligible. We can easily obtain the space complexity from the structure of matrix $\mathscr{L}$.

Proposition 8.3.1. *The space complexity of the* regMVMT *algorithm is* $\mathbf{O}(\sum_t D_t^2 + D_t T(T-1))$ *approximately.*

In real-world data sets, the number of views V is usually small ($2 \sim 5$), and the total number of features D from all views is generally much greater than the number of tasks T. Generally D can be in the range of a few hundred to thousands, and feature selection approaches are needed when it is even larger. Due to the limit of the matrix size in most computer systems, our algorithm can only handle up to tens of tasks, and learning problems with hundreds of tasks or more are hence beyond the scope of this chapter.

8.3.8 Implementations of the *regMVMT* and *regMVMT*+ Algorithms

To summarize what we have discussed above, we present the efficient *regMVMT* algorithm in detail, as shown in "The *regMVMT* Algorithm" pseudo code block. The inputs are the label vector y_t and feature matrix X_t^v of labeled samples, the unlabeled feature matrix U_t^v for view v in task t, and model parameters $\{\lambda, \mu, \gamma\}$. The output of the *regMVMT* algorithm is the optimal linear decision functions for all T tasks. The parameters λ, μ, and γ will be optimized using standard five-fold cross validation. Note that the first column of each input feature matrix is an extra added

Algorithm 1 The *regMVMT* Algorithm

```
1: Input: $\{\mathbf{y}_t\}_{t=1}^T, \{\mathbf{X}_t\}_{t=1}^T, \{\mathbf{U}_t\}_{t=1}^T, \lambda, \mu, \gamma$.
2: Output: $\{\mathbf{W}_t\}_{t=1}^T$.
3: Initialize matrix $\mathscr{L}$ and vector $\mathscr{R}$ to be zero.
4: for $(t,v) \in [1:T] \times [1:V]$ do
5:     Construct matrix $A_{tv}$ and vector $E_{tv}$ as in Eq.(8.7).
6:     Construct matrix $B_{vv'}^t$ and $C_{t'v}$ as in Eq.(8.7) for each $v' \neq v, t' \neq t$.
7: end for
8: Construct square matrix $\mathscr{L}$ and column vector $\mathscr{R}$ in Eq.(8.9).
9: Compute $\mathbf{W} := \mathscr{L}^{-1}\mathscr{R}$.
10: Split $\mathbf{W}$ into $T$ vectors, and return $\{\mathbf{W}_t\}_{t=1}^T$.
```

unit vector to offset the intercepts of linear mapping functions. Since the intercepts are not included in all regularization terms, an identity matrix with the first diagonal element as 0 is multiplied with the constant terms ($\lambda + \gamma(T-1)$ and -γ) in Eq.(8.7) when constructing the matrix A_{tv} in line 5 and the matrix C_{tv} in line 6, respectively.

The implementation of the *regMVMT+* algorithm is similar to the *regMVMT* algorithm. One difference is the construction of block matrix $\mathbf{A}_{tv}$'s and $\mathbf{C}_{tv}$'s. An additional difference is that *regMVMT+* is an iterative algorithm, and within each iteration the optimization of $\mathbf{w}_t$'s uses a procedure that is very similar to the *regMVMT* algorithm. We repeat the procedure until the predefined convergence thresholds of $\mathbf{w}_t$'s and Ω_v's have been met or we have reached a maximal number of iterations. It is expected that this method needs significantly more computational time for the numerical solutions.

We design an efficient iterative algorithm to optimize both $\mathbf{w}_t$'s and Ω_v's alternately and implement it as in "The *regMVMT+* Algorithm" pseudo code block 8.3.8. First Ω_v is initialized to $\frac{1}{T}I_T$ (I_T is the $T \times T$ identity matrix) for each view v, which corresponds to the assumption that all tasks are initially uncorrelated. We optimize the convex objective function in Eq.(8.12) over $\mathbf{w}_t$'s, update Ω_v's using Eq.(8.13), and then plug in the new Ω_v's to Eq.(8.12) to optimize $\mathbf{w}_t$'s again. The procedure is repeated until the convergence of both $\mathbf{w}_t$'s and Ω_v's under a predefined threshold.

Algorithm 2 The *regMVMT+* Algorithm

```
Input: ${\mathbf{y}_t}_{t=1}^T, {\mathbf{X}_t}_{t=1}^T, {\mathbf{U}_t}_{t=1}^T, \lambda, \mu, \gamma, N_{it}, \varepsilon$
Output: ${\mathbf{W}_t}_{t=1}^T, {\Omega_v}_{v=1}^V$
Initialize $\mathbf{W}_0 := 0$ and $\Omega_{v0} := \frac{1}{T} I_T$ for $v \in [1 : V]$
for $it = 1$ to $N_{it}$ do
    for $(t, v) \in [1 : T] \times [1 : V]$ do
        Construct matrix $A_{tv}$ in Eq.(8.14) and vector $E_{tv}$ in Eq.(8.7)
        Construct matrix $B_{vv'}^t$ in Eq.(8.7) and $C_{t'v}$ in Eq.(8.14) for each $v' \neq v, t' \neq t$
    end for
    Construct square matrix $\mathscr{L}$ and column vector $\mathscr{R}$ in Eq.(8.9)
    Compute $\mathbf{W} := \mathscr{L}^{-1}\mathscr{R}$
    Update $\Omega_v$ using Eq.(8.13) for each $v \in [1 : V]$
    if $\|\mathbf{W} - \mathbf{W}_0\|_1 < \varepsilon$ & $\|\Omega_v - \Omega_{v0}\|_1 < \varepsilon$ then
        break
    else
        $\mathbf{W}_0 := \mathbf{W}$
        $\Omega_{v0} := \Omega_v$ for each $v \in [1 : V]$
    end if
end for
Split $\mathbf{W}$ into $T$ vectors, and return ${\mathbf{W}_t}_{t=1}^T$ and ${\Omega_v}_{v=1}^V$
```

8.4 Experimental Results

In this section, we present the experimental results of the two proposed MVMT methods and four baseline methods on three real-world multi-view data sets with multiple tasks. The baseline methods that we used are detailed below.

Regularized MTL (regMT): If we consider no co-regularization on different views in a given task, we convert the MVMT learning problem into the single-view MTL problem [Evgeniou & Pontil, 2004] by concatenating the feature vectors from all views to a single feature vector and merging multiple views into one view. Comparison with this baseline method can help demonstrate the benefits from using multiple views instead of a single view.

Co-regularized MVL (coMV): By ignoring the multi-task relatedness, we apply the co-regularized MVL method [Sindhwani & Niyogi, 2005] on each task. The implementation is obtained by setting the parameter $\gamma = 0$ in the *regMVMT* formulation.

Multi-task Relationship Learning (regMT+): We may learn the task relationships among the multiple related tasks from the input data when we consider no regularization on different views in a given task. The implementation is obtained by setting the parameter $\mu = 0$ in the *regMVMT+* formulation.

Iterative MVMT (IteM^2): We also compare our methods with the state-of-the-art MVMT method proposed by He *et al.* [He & Lawrence, 2011]. The authors didn't release the software, and we implemented it according to the pseudo-code provided in the IteM^2 algorithm in the original work. This is the most important competing method that our methods will compare with.

8.4.1 Data Sets

We collected three multi-view data sets with multiple tasks. The first one is the WebKB data set [Blum & Mitchell, 1998] with 1,051 webpages collected from four universities, and the goal is to classify whether each webpage is course related or not. Here each university is a task, and we have four tasks. There are three views for each webpage: the bag-of-word features from the webpage title, from the webpage main text, and from the main text of the webpage with hyperlinks to the given webpage. There is no missing views in this data set.

The second one is the email spam data set in the ECML 2006 Discovery Challenge [2], and the goal is to classify if each email is spam or not. There are three email users with 2,500 emails for each user, and each user is considered a task. Four bag-of-words views are created: one common view shared by all three tasks, and three task specific views with each for a task. The common view consists of the common vocabulary that exists in all the three tasks, while each task specific view consists of the vocabulary unique to each task, as discussed in He *et al.* [He & Lawrence, 2011]. Each task has two missing views, which are the two views specific to the other two tasks.

The third data set is extracted from the NUS-WIDE Object web image database [Chua et al., 2009] where each image is annotated by objects such as "book", "bird", and etc. We removed all images that are associated with zero or only one object, resulting in an object data set consisting of 3,545 samples in 31 tasks. We used a total of 634 features extracted from images [Chua et al.,

[2] http://www.ecmlpkdd2006.org/challenge.html

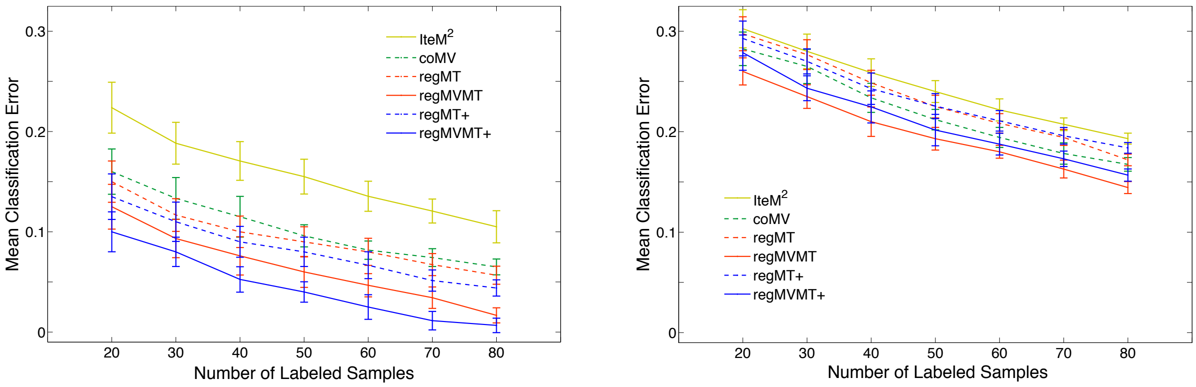

Figure 8.3: Experimental results on the WebKB data set (left) and on the NUS-WIDE Object data set (right).

2009], which can be considered as five views: 64-dim color histogram, 144-dim color correlogram, 73-dim edge direction histogram, 128-dim wavelet texture, and 225-dim block-wise color moments. We merged the five types of features for each sample into two views with the 225-dim block-wise color moments as one view and the rest as the other view. By removing those tasks with too few positive or negative samples, we obtained a multi-view data set with 11 tasks. Since this data set consists of a significant portion of negative features, when we applied the IteM^2 method to it, we added a positive constant to the values of negative features of all samples so that all negative features become non-negative. There is no missing views in this data set.

The three data sets are summarized in Table 8.1, where N_p and N_n denote the total number of positive and negative samples available in each data set. As stated in Section 3, T is the number of tasks, V is the number of views, and D is the total number of features from all views.

Table 8.1: Statistics of Data Sets Used.

Data Set	WebKB	ECML2006	NUS-WIDE Object
V	3	4	$2 \sim 5$
T	4	3	11
View Missing?	No	Yes	No
N_p	230	2,543	$389 \sim 1325$
N_n	821	2,929	$2220 \sim 3156$
D	2,096	5,597	634

8.4.2 Model Construction and Evaluation

In each MVMT data set we randomly select the same number n of labeled samples and the same number m of unlabeled samples for each task, where n is varying in the range [20,80] with the increment of 10, and m is generally $2 \sim 4$ times of n. For each subset consisting of n labeled samples and m unlabeled samples for each task, we first randomly selected 20% labeled samples as the independent testing set, and the remaining 80% labeled and all unlabeled samples are the training set.

The regularization parameters in our methods allow flexible tradeoffs between different regularization terms. We apply five-fold cross validation on the training set to optimize the parameters for each method discussed in this chapter: λ, μ and γ for *regMVMT* and *regMVMT+*, λ and γ for *regMT* and *regMT+*, and λ and μ for *coMV*. For the IteM^2 method, we performed five-fold cross validation to optimize the parameter μ and b, but found not much difference. We simply used the optimal values of parameter $\mu = 0.01$ and $b = 1$ that were set in the original work [He & Lawrence, 2011]. We applied grid searching to identify optimal values for each regularization parameter.

After obtaining the optimal parameters for each method, we construct a predictive model using all the training samples and calculate the classification error for each task on the independent testing set using the final task functions in Eq.(8.9). Each experiment was repeated for 10 times, the mean classification error for each task was calculated separately, and the mean and standard deviation of the classification errors across all tasks in each data set were reported. Here classification error $= 1-(TP+TN)/N$, where TP, TN, and N stand for the number of true positives, true negatives, and the total number of samples in the testing set, respectively.

8.4.3 Learning with Complete-view Data

We first consider the ideal case that all tasks have complete-view data, i.e. each of the T tasks has features from all the V views and the indicator matrix I_d is a unit matrix. We perform experiments on the WebKB data set which has four tasks ($T = 4$) and each task shares all the three views ($V = 3$). The left panel in Figure 8.3 shows for each method how the mean classification error changes with regard to the number of labeled samples. Note that the standard deviations are marked as error bars in each curve.

We observe a common trend that the classification errors and the variances for all methods decrease as the number of labeled samples for training increases. The two MTL baseline methods (*regMT* and *regMT*+ perform marginally better than the MVL baseline method (*coMV*), but the performance difference among the three baseline methods is not significant in a two-sample t-test. We test the significance of the results between *regMVMT* and the two baselines *regMT* and *coMV* with two-sample t-test, and find that *regMVMT* significantly outperform *regMT* and *coMV* at the 5% significance level. Similarly t-test shows that *regMVMT*+ is significantly better than the baseline *regMT*+. Our proposed inductive MVMT methods (*regMVMT* and *regMVMT*+) significantly outperform the transductive MVMT method (IteM^2) by He *et al.* with the improvement margin of up to $9 \sim 12\%$.

Next we investigate the performance of our MVMT methods on the NUS-WIDE Object data set. We conduct similar experiments and present the results in the right panel in Figure 8.3. We observe similar trends to the results of the WebKB data set. Our proposed MVMT methods (*regMVMT* and *regMVMT*+) perform better than all the the baselines, especially the IteM^2 method, and the difference is statistically significant with a two-sample t-test.

Different from the previous data set, the *regMVMT* method with ℓ_2 regularization performs slightly better than the variant MVMT method with task relationship learning (*regMVMT*+), but the difference is not significance in the t-test. The performance of the three MTL/MVL baseline methods are close and their curves are entangled. In this data set the improvement margin of our MVMT methods is up to 8% over the transductive counterpart.

8.4.4 Learning with Missing-view Data

As we discussed earlier, it is common that the multiple tasks in real-world data sets do not share all views. We would like to examine the performance of our MVMT methods compared with the IteM^2 method and other baseline methods in this setting. In the WebKB data set, we randomly

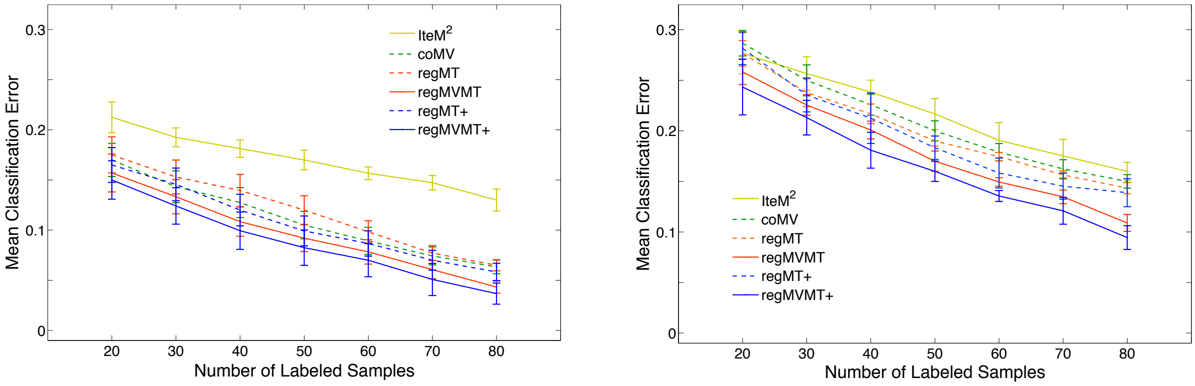

Figure 8.4: Experimental results on the WebKB data set with one missing view for each task (left) and on the ECML2006 Email data set that has two missing views for each task (right).

select one view in each task and masked it as a missing view. Hence a multi-view data set with missing views is artificially created. Here we enforce a constraint that no more than two tasks miss the same view. We perform the same experiments on this new data set using all the methods, and present the results in the left panel in Figure 8.4. First we observe that the trend of the performance curves of all methods are very similar to the left panel in Figure 8.3. Due to the information loss in the missing views, the performance of all methods decreases for about 5%. Our proposed MVMT methods build accurate models using the existing views in each task, and significantly outperform the IteM^2 method with an improvement margin of up to 12%. In addition, the *regMVMT*+ method slightly performs better than the *regMVMT* method, maybe due to the more flexible modeling of task relationships.

We then conduct experiments on the ECML2006 Email data set which consists of three task and four views. There is one common vocabulary view that is shared by all the three tasks, while each of the other three views is task specific and owned by only a particular task. Since the four views are not shared by all the three tasks, it is equivalent to the case there are two entire views missing from each of the three tasks. The experimental results of all methods are plotted in the right panel in Figure 8.4. Our MVMT methods perform better than the three baseline methods and the transductive MVMT method IteM^2 with the margin of up to 7%, and repeated measures t-test (paired two-sample t-test) demonstrates that the difference is statistically significant. All other similar trends are also observed. Since there are more missing views in each task, the overall performance of all methods shrinks more significantly.

8.4.5 Task Relationship Modeling

We perform a case study to understand how our methods learn task relationships using the NUS-WIDE Object data set with 11 tasks. For any two tasks, we calculate the pairwise correlation using the fraction of samples that are simultaneously active or inactive to the two tasks. For the *regMVMT* method, we calculate the learned relationship between any two tasks using the Gaussian kernel of the ℓ_2 norm of the difference between their decision functions as in Eq.(8.5). We make a 2D plot with each dot representing a pair of tasks, as presented in Figure 8.5. The correlation coefficient $R = 0.8978$ reveals the learned task relationships (y-axis) are highly correlated to the pairwise task correlation from the data (x-axis).

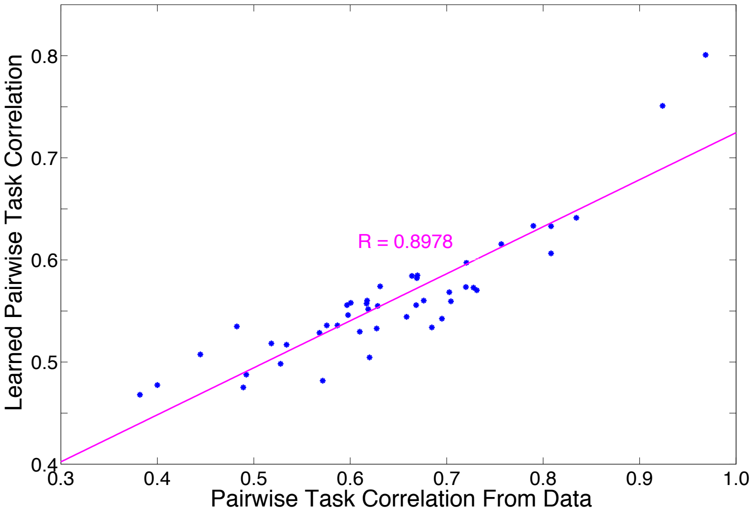

Figure 8.5: Correlation analysis (*regMVMT*) on the Object data set.

Looking into the 11 tasks, we find that they form three clusters: animals (5 tasks), vehicles (3 tasks), and plants (3 tasks). We then apply the *regMVMT*+ algorithm on this data set and obtain the matrices Ω_v's that model the task relationships of the 11 tasks. After taking the mean of all task relationship matrices $\Omega_v, v = 1,2$, we obtain the learned task relationship matrix whose element

(i, j) indicates the relationship index of task i and j. For each task in the animal cluster, we identify the four most significantly correlated tasks, which are found mostly also the tasks in the animal class with only two violations. For the vehicle class and the plant class, we also find two violations that are not in the same class as the seed task, one in each class. The experiments demonstrate that the *regMVMT+* method can well model the task relationships through the learning process.

8.5 Conclusions

In this chapter, we proposed an inductive multi-view learning algorithm for multiple related tasks. In our algorithm we developed a co-regularized framework. We utilized several regularization functions to control the complexity of the learning algorithms. We also developed two extensions. One handles structured missing views and the other handles non-uniformly related tasks. Experimental results demonstrated that our MVMT methods significantly outperform the state-of-the-art MVMT method IteM^2 and other baseline methods. In the future, we will further extend our work to different types of missing values.

Chapter 9

Applying Inductive MVMT Learning for Jointly Predicting Multiple ADRs

There are two major challenges in constructing effective models for ADR prediction. First, there are many ADR-causing factors and the molecular mechanism of ADRs are usually not clear. The adverse reaction profile of a drug depends on not only its chemical structure, but also its functional groups, binding proteins, interacting drugs, and etc. Models constructed using only part of these information sources may suffer from low accuracy and overfitting. Second, for each drug, it usually associates with multiple ADRs. The available data, however, are usually limited and noisy. For instance, the SIDER database (`http://sideeffects.embl.de`) is a public digital resource of adverse drug reactions [Kuhn et al., 2010]. SIDER contains information of 888 drugs and their connections to 1,385 ADR terms. On average each drug is associated with ~69 ADRs, and each ADR is caused by ~44 drugs. In this chapter, we present a unified framework to handel the aforementioned two challenges and jointly predict multiple ADRs using the novel machine learning approach described in Chapter 8, which we call as multi-view multi-task (MVMT) learning.

9.1 Introduction

For learning tasks with multiple sources of data (e.g. drug chemical structures and its interacting protein profiles), multi-view learning [Sindhwani & Niyogi, 2005, Culp et al., 2009, Krishnapuram et al., 2004] is a technique that aims to improve modeling quality in the presence of multiple data sources. The underlying assumption is that multiple views present conditionally independent information and the combination of the two information sources provides better modeling capability [Amini et al., 2009, Dasgupta et al., 2001]. Handling multiple labels, we utilize multi-label learning to annotate multiple ADRs that may be associated with a single drug. Technically learning from multi-label data corresponds to finding a mapping from the feature space to the power set of all labels, and the multiple labels have latent correlations between one another. In our study we treat each label as a learning task and hence transform the multi-view multi-label problem as a multi-view multi-task learning problem.

Little effort has been made toward integrating multi-view and multi-task learning into the same learning framework so that the benefits provided by both algorithms retain. Cavallanti *et al.* [Cavallanti et al., 2010] proposed linear algorithms for online multi-task learning, but the features of a single task do not form multiple views. He *et al.* [He & Lawrence, 2011] developed an efficient graphical framework for transductive MVMT learning problems. However, this method was mainly designed for text or document mining, and hence imposed a few constraints such as the signs of features and the fraction of positive samples in training and testing sets. More general and robust classification algorithms for MVMT learning problems are highly desired.

Here we propose a novel MVMT learning algorithm for predicting multiple ADRs jointly. In our method one linear mapping function is learned for each view in each task, and the final model for each task is obtained by averaging the prediction from the decision functions of all its views. We first adopt a co-regularized multi-view [Sindhwani & Niyogi, 2005, Sindhwani & Rosenberg, 2008] model on each task by optimizing a convex object function that penalizes both the misclassification on labeled samples, and the disagreement among prediction from different view functions on unlabeled samples in the same task. We then apply regularized multi-task learning [Evgeniou

& Pontil, 2004] to enforce the multiple tasks to be *similar* on the same views, and learn all task functions jointly. All the steps described above are integrated in a single objective function, and the analytic solution are derived to optimize the objective function.

The main contributions of this work is the formulation of ADR prediction as a MVMT classification problem, and we achieve significantly more accurate predictive models. The rest of this chapter is organized as follows. In Section 2, we briefly review related works on predicting ADRs and some studies on multi-view learning. We explain our co-regularized MVMT algorithm in detail and implement an efficient computer algorithm with pseudo code in Section 3. Then we conduct thorough experimental studies on a data set of adverse drug reactions in Section 4 for demonstrating the effectiveness of the proposed MVMT method, as compared to baseline learning methods. Finally we conclude our work in Section 5.

9.2 Related Work

A variety of statistical methods on predicting ADRs have been in literature, such as analysis of pharmacology data [Bender et al., 2007] or toxicity related proteins [Zhang et al., 2007]. Ursem *et al.* applied QSAR and an expert system to FDA post-market reports for estimating the mechanism of action of drug-induced hepatobiliary and urinary tract toxicities [Ursem et al., 2009]. Yang *et al.* investigated clinical trial data with adverse event frequencies for kinase inhibition-related adverse events prediction. A mini-review on this class of methods can be found in [Tatonetti et al., 2009]. In addition, Ji *et al.* exploited *in silico* protein docking method (INVDOCK) to search for putative ADR-related proteins and used these proteins as tools to facilitate adverse effect prediction [Ji et al., 2006], resulting in a valuable collection of ADR-related proteins. Hammann *et al.* developed structure-activity relationship analysis of ADRs in the central nervous system (CNS), liver, and kidney, and also of allergic reactions for identifying drugs suspected of causing adverse reactions, and then used decision tree modeling to determined the properties of compounds that predispose them to causing ADRs [Hammann et al., 2010].

The first MVMT work is proposed by He *et al.* [He & Lawrence, 2011], where a bipartite graph of the features and samples is constructed for each view in each task. However, He *et al.*'s method can only be applied to MVMT problems whose most (if not all) features have positive signs, although many real-world data sets have significant amount of negative features. This transductive learning method requires that the fraction of positive samples in training and testing sets must be the same, and cannot be used on independent unknown testing sets. He *et al.*'s method treats testing samples as unknown data in the training process, and hence the results are only in-sample prediction. Our experimental study also confirmed the advantage of our methods over *et al.*'s method.

Multi-view learning has been applied to many bioinformatics applications. For sample Culp *et al.* [Culp & Michailidis, 2009] proposed a co-training algorithm for multi-view data with applications on drug discovery data fusion, and their aim is to link drug molecules with the diseases. Sokolov *et al.* conducted multi-view prediction of protein functions in both a co-training and transductive structured SVMs framework, incorporating with the structured output learning on the multiple labels [Sokolov & Ben-Hur, 2011]. Here we report the adaptation of multi-view learning for drug ADR prediction.

9.3 Methods

In this section, we present a multi-task learning algorithm with multiple view data, i.e., the co-regularized multi-view learning algorithm that learns multiple related tasks jointly, and denote it as *regMVMT*. The diagram of MVMT learning is shown in Figure 9.1. This algorithm has been previously published in the KDD'12 conference by our group. Our contribution in this work is to apply this MVMT method for predicting multiple adverse drug reactions jointly. Hence the introduction and description of our proposed methods (*regMVMT* and *regMVMT+*) is omitted for simplicity. For details of each method present in this chapter, refer to the **Methods** section in Chapter 8. Note that the notation of symbols and methods are identical in both chapters.

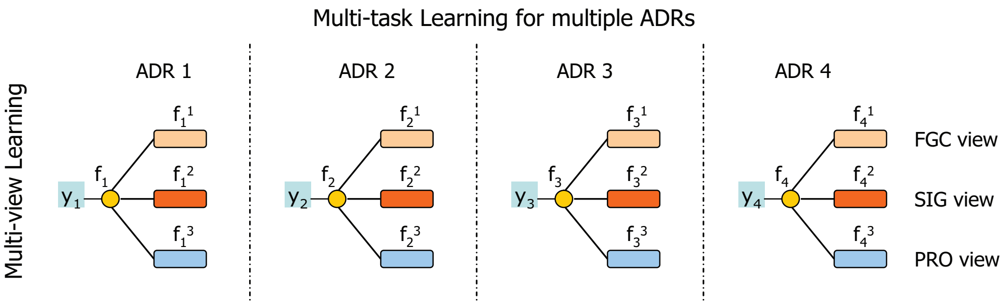

Figure 9.1: Graphical representation of the MVMT learning framework for ADR prediction. FGC: function group counts, SIG: molecular signature, PRO: protein binding profiles.

9.4 Results

In this section, we conduct a comprehensive experimental study of the *regMVMT* method and compare it with three state-of-the-art baseline methods, and present the results in detail. The baseline methods that we used are introduced below.

Regularized MTL (regMT): If we consider no co-regularization on different views in a given task, we convert the MVMT learning problem into the single-view MTL problem [Evgeniou & Pontil, 2004] by concatenating the feature vectors from all views to a single feature vector and hence merging multiple views into one view. Comparison with this baseline method can help demonstrate the benefits from using multiple views instead of a single view.

Co-regularized MVL (coMV): By ignoring the multi-task relatedness, we apply the co-regularized MVL method [Sindhwani & Niyogi, 2005] on each task. The implementation is obtained by setting the parameter $\gamma = 0$ in the *regMVMT* formulation.

Multi-task Relationship Learning (regMT+): We may learn the task relationships among the multiple related tasks from the input data when we consider no regularization on different views in a given task. The implementation is obtained by setting the parameter $\mu = 0$ in the *regMVMT+* formulation.

Iterative MVMT (IteM^2): We also compare our methods with the state-of-the-art MVMT method proposed by He *et al.*. We implemented it according to the pseudo-code provided in the *IteM*^{2} algorithm [He & Lawrence, 2011].

ℓ_2-regularized LSS (regST): If we apply no regularization on different tasks as in [Evgeniou & Pontil, 2004], we obtain the most basic single-task learning method (ℓ_2-regularized least squared loss): all views are merged into one view, and all different tasks are learned individually one by one. The implementation is obtained by setting the parameter $\mu = 0$ in the *regMT* formulation.

9.4.1 Data Sets and Feature Extraction

We collected the data set of adverse drug reactions from the Side Effect Resource (SIDER) database [Kuhn et al., 2010]. SIDER is a public, computer-readable drug side effect resource with links between drugs and ADRs, obtained from the Drug Labels provided by the FDA with text mining tools. We mapped the drugs in SIDER to the DrugBank database [Knox et al., 2011, Wishart et al., 2008] via PubChem Compound ID and removed those drugs with trivial or too simple structures, resulting in 797 drugs associated with 1,362 ADR labels, which are represented by the COSTART controlled vocabulary. It would be difficult for our methods to handle such a large number of ADRs simultaneously, and also some ADR labels have very few positive samples. We manually selected 12 ADRs which belong to three different classes: five for central nervous system (CNS) injury, four for liver injury, and three for kidney injury. There are totally 494 drug compounds associated with these ADRs, and we downloaded their chemical structures from DrugBank. To obtain unlabeled samples for multi-view learning, we selected 2,400 drug related compounds from the PubChem database, and these compounds were not found in the DrugBank or the SIDER databases. Here drug related compounds are defined as the results by searching “drug” in PubChem. There are no labels for these compounds but they are drug related. The distribution of positive and negative samples in the 12 ADRs are summarized in Table 9.1.

We computed the descriptors of functional group counts using the DRAGON software (`http://www.talete.mi.it`) and the molecular signature descriptors proposed by Faulon *et al.* [Faulon et al., 2004]. The descriptors of functional group counts focus on only a set of predefined functional groups and convert a chemical structure into a vector whose elements are the count of each functional group. The molecular signature descriptors enumerate all possible substructures with

Table 9.1: The distribution of compounds associated with the 12 ADRs.

No.	ADR Name	#Positive	#Negative
1	myocardial infarction	196	298
2	cardiac arrest	144	350
3	myocardial ischemia	68	426
4	ventricular tachycardia	85	409
5	congestive heart failure	128	366
6	diabetes mellitus	108	386
7	hepatomegaly	59	435
8	hepatitis	235	259
9	hepatic failure	121	373
10	interstitial nephritis	62	432
11	renal failure	149	345
12	renal insufficiency	56	438

the preset sizes in a chemical and convert each molecule into a vector whose elements are the count of each such substructure. Refer to related publications for more detailed introduction to these two feature extraction methods. These two sets of features are weakly correlated, and we will investigate their contributions to our models when treating each set as a view.

The third set of descriptors that we extract is the protein binding profiles of the 494 drugs. We extracted the protein binding profiles of these drugs from the STITCH database [Kuhn et al., 2009]. STITCH is a data repository of interactions of chemicals and proteins. In STITCH, chemicals are linked to proteins by evidence derived from experiments, databases and the literature. Specifically, the protein-chemical interactions present in STITCH are either imported from a set of existing databases such as DrugBank and PharmGKB, or from text mining onto a consolidated set of chemicals that has been derived from PubChem with a confidence score for each interaction assigned. All human proteins in STITCH that are interacting with at least one of the 494 drug compounds are identified and are merged into the same set. Each drug is converted into a binary vector whose *i*-th element is 1 if the drug interact with the *i*-th protein in our list and 0 otherwise.

In summary, we obtained three set of descriptors for each compound: 116 functional group counts, 341 molecular signature descriptors, and binding activity with 470 proteins. In the subsequent discussion, we call them the FGC view, the SIG view, and the PRO view, respectively.

9.4.2 Model Construction and Evaluation

In our experimental study, we use different numbers of labeled samples in the range of 100-400 and about four times of labeled samples. For each subset, we first randomly select 20% labeled samples as the independent testing set, and the remaining 80% labeled and all unlabeled samples are for the training set. We apply five-fold cross validation and exponential grid searching on the training set to optimize the parameters for each method discussed in this chapter: λ, μ and γ for *regMVMT*, λ and γ for *regMT*, and λ and μ for *coMV*. For the IteM^2 method, we simply use the optimal values of parameter $\mu = 0.01$ and $b = 1$ that are set in the original paper [He & Lawrence, 2011].

The set of model parameters that return the lowest mean classification error over all ADRs are selected as the best parameters. We then use the full training data set to train a model and apply it to the testing set and calculate the testing classification error. We repeat each experiments for 10 times, and report the mean classification errors and standard deviations with different numbers of labeled samples. Here the *classification error* is defined as the ratio of the number of misclassified samples over all samples in the testing set, i.e. $1-(TP+TN)/N$, where TP, TN and N stands for the true positives, true negatives, and total number of testing samples, respectively.

9.4.3 Preliminary Results

We first conduct a brief case study on the MVMT methods, the four baseline methods, and also a few excellent single-view single-task learning methods such as support vector machines (SVMs), decision tree (C4.5), and naive bayes. For each multi-view learning method we use 200 labeled samples and 800 unlabeled samples, while for single-view methods all three views are merged into one view and no unlabeled samples are used. We report the mean classification errors of each method on each ADR as in Table 9.2. The *regMVMT+* and *regMVMT* methods perform better than other methods on almost all ADRs. Since SVM, decision tree, and naive bayes don't provide valuable comparison baselines, they are excluded from the subsequent experimental studies.

Table 9.2: Classification errors using all three views with 200 labeled samples on the 12 ADRs.

Methods	ADR1	ADR2	ADR3	ADR4	ADR5	ADR6	ADR7	ADR8	ADR9	ADR10	ADR11	ADR12
regMVMT+	0.3012	0.2200	0.1020	0.1930	0.2300	0.2017	0.1175	0.2489	0.2100	0.0890	0.2650	0.1215
regMVMT	0.3100	0.2350	0.1150	0.2050	0.2400	0.2100	0.1075	0.2600	0.2050	0.0875	0.2850	0.1075
regMT+	0.3260	0.2415	0.1321	0.2219	0.2415	0.2430	0.1400	0.2732	0.2525	0.1200	0.3200	0.1155
regMT	0.3300	0.2575	0.1425	0.2350	0.2525	0.2550	0.1400	0.2850	0.2475	0.1250	0.3100	0.1225
coMV	0.3275	0.2563	0.1375	0.2394	0.2750	0.2156	0.1294	0.2531	0.2506	0.1031	0.3000	0.1319
IteM^2	0.3125	0.2975	0.1425	0.2650	0.2475	0.2175	0.1425	0.3175	0.2750	0.1325	0.2850	0.1475
regST	0.3550	0.2920	0.1698	0.2545	0.2700	0.2900	0.1720	0.2805	0.2775	0.1450	0.2950	0.1550
SVMs	0.3400	0.2800	0.1600	0.2475	0.2800	0.2925	0.1650	0.2725	0.2675	0.1300	0.3050	0.1450
C4.5	0.4101	0.3715	0.2344	0.3200	0.3623	0.3879	0.2436	0.3900	0.3580	0.2010	0.3775	0.1966
Naive Bayes	0.3588	0.3412	0.2100	0.2946	0.3320	0.3460	0.2202	0.3567	0.3200	0.1830	0.3505	0.1646

9.4.4 Comprehensive Experimental Study

We then conduct comprehensive experiments to compare our proposed MVMT methods with the three baseline methods. In each experiment, we vary the number of labeled samples nL = 100, 150, ..., 400, and select unlabeled samples from the 2,400 drug related unlabeled samples with the number $nU = 4 * nL$. No unlabeled samples were used for the *regMT* method. We calculate the mean and the standard deviation of the mean classification errors of all ADRs, and make plots of the final mean classification error *vs.* the number of labeled samples for each method. Note that the standard deviations are marked as error bars in each figure, and they represent the performance variances of each method on different ADRs.

There are three views in our data set: functional group counts (FGC view), molecular signatures (SIG view), and protein binding profiles (PRO view). We use different combinations of these three views to obtain four subsets. For each subset we conduct experiments according to the protocol described above with different numbers of labeled samples. First, all three views are used and the results are presented in the upper left panel in Figure 9.2. Clearly the *regMVMT* method achieves the best performance, and it outperforms the baseline methods with a margin of 4-5% classification errors. When there are more labeled samples, the classification errors of all methods gradually decrease, though the overall maximal magnitude is only about 3-5%, which demonstrates that using more labeled samples improves the final models but the improvement is limited. The *regMT* method generally performs slightly better than the *coMV* method. The competing method $IteM^2$ performs poorly with small and medium numbers of labeled samples, and it performs better than *regMT* and *coMV* with large numbers of labeled samples. The *regMVMT* method consistently

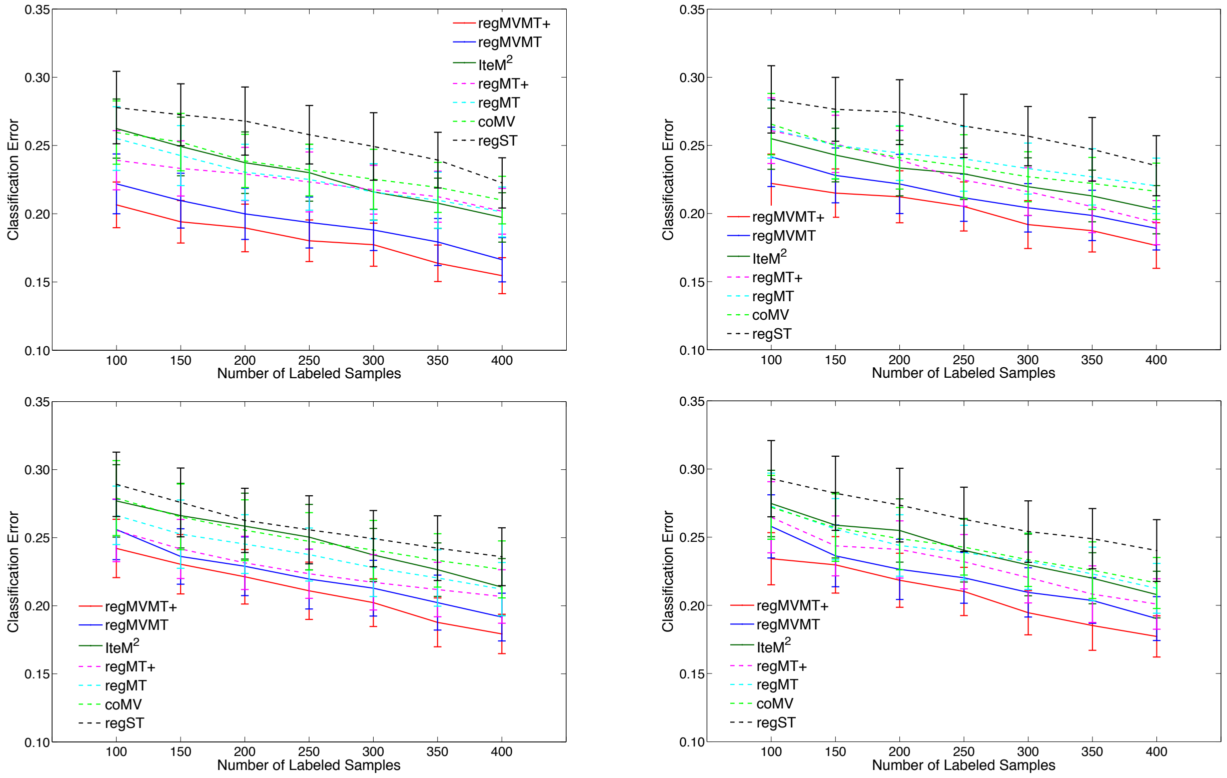

Figure 9.2: Mean classification errors using 100-400 labeled samples. Upper left: FGC + SIG + PRO view, upper right: FGC + SIG view, lower left: FGC + PRO view, and lower right: SIG + PRO view.

outperforms the *IteM*^{2} method, and χ^2-test shows that the improvement is significant ($P < 10^{-3}$).

Next we examine the three combinations of using two views and perform similar experiments as above, and the results are plotted in the other three panels in Figure 9.2. As we discussed earlier that the FGC and the SIG view are both extracted from chemical structures, and they are weakly correlated. The upper right panel shows the experimental results when using these two views, and the performance of all methods significantly decrease, and the advantage of the *regMVMT* method over baseline methods also shrinks. The results reveal that the two views are not completely independent and multi-view learning affords only limited performance gain. When we include the protein binding profile view in our models, the performance of the *regMVMT* method also increases and become comparable to the results when using all three views, as shown in the two lower panels in Figure 9.2. Furthermore, we conduct experiments on the same data set of 12 ADRs using the extended MVMT method (*regMVMT+*) and the extended baseline method (*regMT+*). The extended MVMT method slightly performs better than the counterpart method *regMVMT* for about 2% accuracy improvement. Though the improvement is marginal, it demonstrates the effectiveness of task relationship learning. These experimental results show that the PRO view is critical for ADR prediction. Nevertheless, the experimental results using all the three views are the best, which demonstrates the causation of ADRs is related to functional groups, important substructures, and protein binding profiles jointly.

9.4.5 Recovery of ADR Clusters

When we select the 12 ADRs, we intentionally collect them from three groups, and we aim to examine if our MVMT method can effectively recover the three groups via cluster analysis. We conduct experiments using 200 labeled 800 unlabeled samples, and repeat the experiments for 20 times. We calculate the mean decision function for each of the 12 ADRs, and perform K-means clustering on the 12 ADRs using the coefficients in the decision functions as features for each ADR. Results show that our MVMT method can successfully recover all three ADRs in the kidney injury cluster, four of the five ADRs in the CNS cluster, and two of the four ADRs in the liver injury cluster. Three ADRs (congestive heart failure, diabetes mellitus, and hepatomegaly) are incorrectly clustered into the kidney injury cluster, resulting in a 75% accuracy of clustering.

We perform literature survey to investigate the three seemly misplaced ADRs, and results show that all these three ADRs are also related to and/or cause kidney injury (cluster 3). Agrawal *et al.* [Agrawal & Swartz, 2000] showed that acute renal failure in patients with congestive heart failure occurs because of decreased renal blood flow caused by the latter, which confirmed that congestive heart failure also causes kidney damage. Clinical studies [Soläng et al., 1999] predicted that diabetes and congestive heart failure are highly correlated and there will be a considerable increase in diabetes-related cardiovascular disease in the near future [Clark & Perry, 1999]. In the kidney, damage resulting in the development of diabetic nephropathy (kidney damage secondary to diabetes) may lead to kidney failure in extreme cases [Newman, 2005], and minor kidney damage in patients with type 1 diabetes leads to increased mortality. In addition, Nachar *et al.* [Nacher et al., 2001] demonstrated that hepatomegaly is associated with jaundice with acute renal failure. All these literature results verify that the three misclassified ADRs (congestive heart failure, diabetes mellitus, and hepatomegaly) are not really classified in the wrong cluster (kidney injury) by using the results of our MVMT methods. Due to the fact that the three ADRs cause injury on multiple organs, clustering them into the kidney injury cluster using our MVMT method reveals the hidden mechanisms.

9.4.6 Significant Features

From the results of the 20 repeated experiments above, we calculate the mean and the variance of the decision functions for each ADRs. The Z-score of a feature is defined as the mean to variance ratio of its coefficient, and it can be used to evaluate the significance of this feature to the final models [Hastie et al., 2009]. A feature with the absolute value of the Z-score greater than 2.0 is generally considered significant. Specifically, a positive (negative) Z-score means that it is more statistically probable for a molecule with (without) the feature to have the ADR. We identify 19 such features that are significant to at least one ADR, including four functional groups and eight human binding proteins. The five positively significant features are listed in Table 9.3.

Table 9.3: Some features significant to the ADR models.

Feature	Z-score	Adverse Drug Reaction
Ar-OR(functional group)	2.020	congestive heart failure
Brain-derived neurotrophic factor precursor(BDNF)	2.194	hepatitis
DNA topoisomerase(TOP2B)	2.206	hepatic failure
Arginine vasopressin(AVP)	2.008	renal failure
Prostaglandin E(PTGES2)	2.236	myocardial infarction

Existing literature confirm the correlation of the identified significant features to their target ADRs. For instance, protein TOP2B is found a commonly upregulated gene in intrahepatic cholangiocarcinomas [Obama et al., 2005]. Schrier *et al.* [Schrier & Wang, 2004] postulate that arginine vasopressin (AVP) may have a protective effect on kidney function in patients with sepsis in their review acute renal failure and sepsis. Literature also show that congestive heart failure is highly related to prostaglandin E synthase 1 and 2 via unknown mechanisms [Wang & FitzGerald, 2010,Giannico et al., 2005]. By identifying significant features using our MVMT learning models, we can map adverse drug reactions into both chemical space (e.g. functional groups and molecular substructures) [Scheiber et al., 2009] and biological space (e.g. protein binding profiles), and achieve more insightful results.

9.5 Conclusions

Multi-view learning has been applied to many challenging machine learning and data mining applications, but almost all of them learn a single task at a time and are for binary classification. There is a gap between our current understanding and the need of solving the more challenging multi-view multi-task learning problem. Here we first formulate ADR prediction as a multi-view multi-task learning problem, and then apply the previously described inductive MVMT algorithm (*regMVMT*) and develop an effective algorithm to optimize the convex objective function. We conduct regularized MTL learning [Evgeniou & Pontil, 2004] by treating each ADR as a task, while within each task one decision function is learned on each view and the co-regularized MVL [Sindhwani & Niyogi, 2005] approach is applied to the task. We further extend the *regMVMT* method by learning task relationship from data, which allows the task relationships to positively or negatively correlated, or uncorrelated. Experimental results show that the *regMVMT+* and *regMVMT* method outperform all the state-of-the-art baseline methods, while *regMVMT+* is slightly better than *regMVMT* due to learning task relationships. Moreover, recovery of ADR clusters and significant feature identification reveals interesting hidden mechanisms underlying the causation of ADRs. Note that the *regMVMT+* and *regMVMT* algorithms become very inefficient and may use up all memory when the number of tasks is large. Handling a large number of tasks in the MVMT learning framework could be our future work plan.

Chapter 10

Conclusions and Future Work

Adverse drug reactions represent a huge burden to the national public health care system, a leading cause of deaths to patients, and a major obstacle in drug development. Hundreds of billions of dollars are spent on drug-related morbidity and mortality, over 100,000 deaths are caused by ADRs, and about one third of late stage drug failures are due to adverse effects. Although ADRs are not preventable, they can be anticipated and predictable. Effective prediction of ADRs is highly challenging due to the fact that there are so many factors related to the causation of ADRs, including drug chemical structures, functional groups, their interacting protein profiles, protein binding sites, ADME/PK properties, and etc. In this thesis, we investigated the prediction of ADRs using advanced machine learning techniques systematically from multiple perspectives.

We first analyzed a variety of feature extraction methods for chemical descriptors and compared their performance, and then proposed a set of protein features extracted from human protein-protein interaction networks. These network topological features are proven very effective for identifying potential drug targets and off-targets. A further analysis of the BioAssay network existing in PubChem data revealed some important observations for protein-chemical interaction data (chapter 4). Although the chemical structures of all approved drugs are known, the links between ADRs and drugs are, however, highly noisy and sparse with errors, biases, and uncertainty, as shown in online databases such as Arizona CERT (`http://www.azcert.org/`) and SIDER (http://sideeffects.embl.de/, which provide annotation links between approved drugs and ADRs. As one of the most important ADR causing factors, the interacting protein profiles of most drugs are incomplete. Hence we proposed two supervised learning methods for protein-chemical interaction prediction using efficient multi-task learning techniques, and their performance is significantly better than the state of the arts (chapter 5).

For effective ADRs prediction, we proposed a sparse multi-view learning method for predicting a single ADR at a time (chapter 6). The ADR picked is drug-induced QT prolongation, which is caused by the blockage of some important ion-channel proteins. We use features extracted from drug chemical structures as the first view, and the ion-channel binding profiles of drugs as the second view. The sparse MVL method outperformed many existing methods on predicting QT prolongation using the noisy data set. The general binding protein profiles of drugs were also used as the alternative second view, but the performance decreased. The work is the first attempt to apply multi-view learning for ADR prediction, and our method is novel by introducing sparsity in our models and hence improves the performance.

We further extend the MVL method to the multi-task learning framework so that multiple related ADRs can be predicted jointly (chapter 8). Three views were used: substructures, functional groups, and interacting protein profiles (human protein only). A set of ADRs were selected and each was treated as a task. The MVL method was applied to each ADR, and the decision functions of all ADRs are enforced to be similar by regularized multi-task learning. Using a small number of labeled samples and a large number of unlabeled samples, the MVMT learning method outperformed all existing state of the arts significantly. We then improved our method to a more generalized inductive MVMT learning algorithm that can handle missing views and more complex task relationships, applied the new method to other real-world application, and also achieved outstanding performance. This extension represents a significant contribution on methodology development and has been published in a top conference (KDD'12).

There are also some limitations in our methods and implementations. First, our proposed methods are all linear methods, which might be a reasonable approximation when the decision boundaries of the learning problems are not linear separable. Second, our MVMT methods can only handle a relatively small number of tasks (up to 20) in the multi-view learning framework, and learning a large number of tasks is challenging in this MVMT framework. Finally, there are still some ADR-causing factors that are not included in the current models due to data availability. All the limitations will be our future work plans.

In summary, this thesis represents the systematic efforts on effective ADRs prediction and novel method development of relevant multi-task and multi-view learning in the past five years of my Ph.D. study. The methods and results should be beneficial to the future research on ADR prediction, and the tough training will be helpful to my future career.

References

[UCM, 2005] (2005). E14 clinical evaluation of QT/QTc interval prolongation and proarrhythmic potential for non-antiarrhythmic drugs. Food and Drug Administration, `http://www.fda.gov/downloads/RegulatoryInformation/Guidances/UCM129357.pdf`.

[Day, 2007] (2007). *Daylight User Manual*. Daylight, Inc., `http://www.daylight.com`.

[Dra, 2008] (2008). *DRAGON User Manual*. Milano Chemometrics and QSAR Research Group, `http://www.talete.mi.it/products/products.htm`.

[Med, 2010] (2010). *MedWatch - What Is Serious Adverse Event?* The FDA Safety Information and Adverse Event Reporting System, `http://www.fda.gov/Safety/MedWatch/default.htm`.

[Pip, 2010] (2010). *PipelinePilot 5.1*. Scitegic, Inc., `http://www.scitegic.com/`.

[Abney, 2002] Abney, S. (2002). Bootstrapping. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (pp. 360–367).

[Adams & Brantner, 2006] Adams, C. P. & Brantner, V. V. (2006). Estimating the cost of new drug development: Is it really $802 million? *Health Aff*, 25(2), 420–428.

[Agrafiotis et al., 2002] Agrafiotis, D., Lobanov, V., & Salemme, F. (2002). Combinatorial informatics in the post-genomics era. *Nat. Rev. Drug Discovery*, 1, 337–346.

[Agrawal & Swartz, 2000] Agrawal, Kmalay, M. & Swartz, Richard, M. (2000). Acute renal failure. *Am. Fam. Physician.*, 61(7), 2077–2088.

[Ahram et al., 2006] Ahram, M., Litou, Z. I., Fang, R., & Al-Tawallbeh, G. (2006). Estimation of membrane proteins in the human proteome. *In Silico Biology*, 6, 0036.

[Amery, 1999] Amery, W. K. (1999). Why there is a need for pharmacovigilance? *Pharmacoepidemiol Drug Saf.*, 8(1), 61–64.

[Amini et al., 2009] Amini, M.-R., Usunier, N., & Goutte, C. (2009). Learning from multiple partially observed views - an application to multilingual text categorization. In *NIPS'09* (pp. 28–36).

[Ando & Zhang, 2005] Ando, R. & Zhang, T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, 6, 1817–1853.

[Argyriou et al., 2006] Argyriou, A., Evgeniou, T., & Pontil, M. (2006). Multi-task feature learning. In *Advances in Neural Information Processing Systems 19*.

[Austin et al., 2004] Austin, C., Brady, L., Insel, T., & Collins, F. (2004). NIH molecular libraries initiative. *Science*, 306(5699), 1138–1139.

[Bakheet & Doig, 2009] Bakheet, T. M. & Doig, A. J. (2009). Properties and identification of human protein drug targets. *Bioinformatics*, 25(4), 451–457.

[Balcan & Blum, 2005] Balcan, M.-f. & Blum, A. (2005). A PAC-style model for learning from labeled and unlabeled data. In *Proceedings of COLT'05* (pp. 111–126).

[Ball, 2000] Ball, P. (2000). Quinolone-induced QT interval prolongation: a not-so-unexpected class effect. *J. Antimicrob. Chemother.*, 45(5), 557–559.

[Barabasi & Oltvai, 2004] Barabasi, A. & Oltvai, Z. (2004). Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.*, 5, 101–113.

[Barabasi & Albert, 1999] Barabasi, A.-L. & Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439), 509–512.

[Bass et al., 2004] Bass, A., Kinter, L., & Williams, P. (2004). Origins, practices and future of safety pharmacology. *J. Pharmacol. Toxicol. Methods*, 49, 145–151.

[Bender et al., 2007] Bender, A., Scheiber, J., Glick, M., Davies, J. W., Azzaoui, K., Hamon, J., Urban, L., Whitebread, S., & Jenkins, J. L. (2007). Analysis of pharmacology data and the prediction of adverse drug reactions and off-target effects from chemical structure. *ChemMedChem*, 2, 861–873.

[Benson et al., 2006] Benson, J., Chen, Y. P., Cornell-Kennon, S., Dorsch, M., Kim, S., Leszczyniecka, M., Sellers, W., & Lengauer, C. (2006). Validating cancer drug targets. *Nature*, 441, 451–456.

[Bhavani et al., 2006] Bhavani, S., Nagargadde, A., Thawani, A., Sridhar, V., & Chandra, N. (2006). Substructure-based support vector machine classifiers for prediction of adverse effects in diverse classes of drugs. *J. Chem. Inf. Model.*, 46, 2478–86.

[Bickel et al., 2008] Bickel, S., Bogojeska, J., Lengauer, T., & Scheffer, T. (2008). Multi-task learning for hiv therapy screening. In *Proceedings of 25th International Conference on Machine learning* (pp. 56–63).

[Bleicher et al., 2003] Bleicher, K. H., Bohm, H.-J., Muller, K., & Alanine, A. I. (2003). Hit and lead generation: beyond high-throughput screening. *Nat. Rev. Drug Discov.*, 2, 369–378.

[Blum & Mitchell, 1998] Blum, A. & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *COLT'98* (pp. 92–100).

[Bogojeska et al., 2010] Bogojeska, J., Bickel, S., Altmann, A., & Lengauer, T. (2010). Dealing with sparse data in predicting outcomes of hiv combination therapies. *Bioinformatics*, 26(17), 2085–2092.

[Boutell et al., 2004] Boutell, M. R., Luo, J., Shen, X., & Brown, C. M. (2004). Learning multi-label scene classification. *Pattern Recognition*, 37(9), 1757–71.

[Burges, 1998] Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(4), 121–167.

[Butcher, 2003] Butcher, S. P. (2003). Target discovery and validation in the post-genomic era. *Neurochem. Res.*, 28(2), 367–371.

[C. & Hopkins, 2004] C., L. & Hopkins, A. (2004). Navigating chemical space for biology and medicine. *Nature*, 432, 855–861.

[Caruana, 1997] Caruana, R. (1997). Multitask learning: a knowledge-based source of inductive bias. *Machine Learning*, 28, 41–75.

[Cavallanti et al., 2010] Cavallanti, G., Cesa-Bianchi, N., & Gentile, C. (2010). Linear algorithms for online multitask and multiview classification. *J. Mach. Learn. Res.*, 11, 2901–2934.

[Champeroux et al., 2005] Champeroux, P., Viaud, K., El Amrani, A. I., Fowler, J. S. L., Martel, E., Le Guennec, J.-Y., & Richard, S. (2005). Prediction of the risk of torsade de pointes using the model of isolated canine purkinje fibres. *Br. J. Pharmacol.*, 144(3), 376–385.

[Chang & Lin, 2001] Chang, C.-C. & Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[Chaurasia et al., 2007] Chaurasia, G., Iqbal, Y., Hanig, C., Herzel, H., Wanker, E. E., & Futschik, M. E. (2007). UniHI: an entry gate to the human protein interactome. *Nucl. Acids Res.*, 35, D590–594.

[Chaurasia et al., 2009] Chaurasia, G., Malhotra, S., Russ, J., Schnoegl, S., Hanig, C., Wanker, E. E., & Futschik, M. E. (2009). UniHI 4: new tools for query, analysis and visualization of the human protein-protein interactome. *Nucl. Acids Res.*, 37, D657–660.

[Chen et al., 2010a] Chen, J., Liu, J., & Ye, J. (2010a). Learning incoherent sparse and low-rank patterns from multiple tasks. In *KDD'10* (pp. 1179–1188).

[Chen & Rosenfeld, 2003] Chen, S. & Rosenfeld, R. (2003). A gaussian prior for smoothing maximum entropy models. Technical Report (28 pages), Carnegie-Mellon University.

[Chen et al., 2010b] Chen, X., Kim, S., Lin, Q., Carbonell, J. G., & Xing, E. P. (2010b). Graph-structured multi-task regression and an efficient optimization method for general fused lasso. *Stat.*, 1050, 21.

[Chen et al., 2002a] Chen, X., Lin, Y., & Gilson, M. (2002a). The binding database: Overview and user's guide. *Biopoly. Nucl. Acid Sci.*, 61, 127–141.

[Chen et al., 2002b] Chen, X., Lin, Y., Liu, M., & Gilson, M. (2002b). The binding database: Data management and interface design. *Bioinformatics*, 18, 130–139.

[Chen et al., 2001] Chen, X., Liu, M., & Gilson, M. (2001). BindingDB: A web-accessible molecular recognition database. *J. Combi. Chem. High-Throughput Screen*, 4, 719–725.

[Christoudias et al., 2008] Christoudias, C. M., Urtasun, R., & Darrell, T. (2008). Multi-view learning in the presence of view disagreement. In *Proceedings of UAI'08*.

[Chua et al., 2009] Chua, T.-S., Tang, J., Hong, R., Li, H., Luo, Z., & Zheng, Y.-T. (2009). Nus-wide: A real-world web image database from National University of Singapore. In *CIVR'09*.

[Clark & Perry, 1999] Clark, C. J. & Perry, R. (1999). Type 2 diabetes and macrovascular disease: epidemiology and etiology. *Am. Heart. J.*, 138(5), S330–3.

[Classen et al., 1997] Classen, D., Pestotnik, S., Evans, R., Lloyd, J., & Burke, J. P. (1997). Adverse drug events in hospitalized patients: Excess length of stay, extra costs, and attributable mortality. *J. Am. Med. Assoc.*, 277(4), 301–306.

[Cokol et al., 2005] Cokol, M., Iossifov, I., Weinreb, C., & Rzhetsky, A. (2005). Emergent behavior of growing knowledge about molecular interactions. *Nat. Biotechnol.*, 23, 1243–1247.

[Culp & Michailidis, 2009] Culp, M. & Michailidis, G. (2009). A co-training algorithm for multi-view data with applications in data fusion. *J. Chemometr.*, 23(6), 294–303.

[Culp et al., 2009] Culp, M., Michailidis, G., & Johnson, K. (2009). On multi-view learning with additive models. *Ann. Applied Stat.*, 3(1), 292–318.

[Dasgupta et al., 2001] Dasgupta, S., Littman, M. L., & McAllester, D. A. (2001). PAC generalization bounds for co-training. In *NIPS'01* (pp. 375–382).

[de Abajo & Rodríguez, 1999] de Abajo, F. & Rodríguez, L. (1999). Risk of ventricular arrhythmias associated with nonsedating antihistamine drugs. *Br. J. Clin. Pharmacol.*, 47(3), 307–313.

[Deshpande et al., 2005] Deshpande, M., Kuramochi, M., Wale, N., & Karypis, G. (2005). Frequent substructure-based approaches for classifying chemical compounds. *IEEE TKDE*, 17(8), 1036–50.

[DiMasi, 2002] DiMasi, J. A. (2002). The value of improving the productivity of the drug development process: faster times and better decisions. *Pharmacoeconomics*, 20(Suppl. 3), 1–10.

[DiMasi et al., 2003] DiMasi, J. A., Hansen, R. W., & Grabowski, H. G. (2003). The price of innovation: New estimates of drug development costs. *J. Health Econ.*, 22, 151–185.

[Dobson, 2004] Dobson, C. M. (2004). Chemical space and biology. *Nature*, 432, 824–828.

[Drews, 2000] Drews, J. (2000). Drug discovery: a historical perspective. *Science*, 287(5460), 1960–4.

[Eichelbaum et al., 2006] Eichelbaum, M., Ingelman-Sundberg, M., & Evans, W. (2006). Pharmacogenomics and individualized drug therapy. *Annu. Rev. Med.*, 57, 119–137.

[Eppig et al., 2005] Eppig, J. T., Bult, C. J., Kadin, J. A., Richardson, J. E., Blake, J. A., & the Mouse Genome Database Group (2005). The Mouse Genome Database (MGD): from genes to mice–a community resource for mouse biology. *Nucl. Acids Res.*, 33, D471–475.

[Erhan & L'Heureux, 2006] Erhan, D. & L'Heureux, P. (2006). Collaborative filtering on a family of biological targets. *J. Chem. Inf. Model.*, 46, 626–635.

[Evgeniou & Pontil, 2004] Evgeniou, T. & Pontil, M. (2004). Regularized multi-task learning. In *KDD'04* (pp. 109–117).

[Faulon et al., 2004] Faulon, J., Collins, M., & Carr, R. (2004). The signature molecular descriptor. 4. canonizing molecules using extended valence sequences. *J. Chem. Inf. Comput. Sci.*, 44(2), 427–436.

[Faulon et al., 2008] Faulon, J. L., Misra, M., Martin, S., Sale, K., & Sapra, R. (2008). Genome scale enzyme-metabolite and drug-target interaction predictions using the signature molecular descriptor. *Bioinformatics*, 24(2), 225–233.

[Fontaine et al., 2005] Fontaine, F., Pastor, M., Zamora, I., & *et al.* (2005). Anchor-GRIND: Filling the gap between standard 3d QSAR and the grid-independent descriptors. *J. Med. Chem.*, 48(7), 2687–94.

[Freund & Schapire, 1995] Freund, Y. & Schapire, R. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of 2nd European Conference on Computational Learning Theory* (pp. 23–37).

[Gedeck et al., 2006] Gedeck, P., Rohde, B., & Bartels, C. (2006). Qsar - how good is it in practice? comparison of descriptor sets on an unbiased cross section of corporate data sets. *J. Chem. Info. Model.*, 46(5), 1924–36.

[Giannico et al., 2005] Giannico, G., Mendez, M., & LaPointe, M. C. (2005). Regulation of the membrane-localized prostaglandin E synthases mPGES-1 and mPGES-2 in cardiac myocytes and fibroblasts. *Am. J. Physiol. Heart Circ. Physiol.*, 288(1), H165–H174.

[Goh et al., 2007] Goh, K.-I., Cusick, M. E., Valle, D., Childs, B., Vidal, M., & Barabasi, A.-L. (2007). The human disease network. *Proc. Natl. Acad. Sci.*, 104(21), 8685–8690.

[Gurwitz et al., 2000] Gurwitz, J. H., Field, T. S., Avorn, J., McCormick, D., Jain, S., Eckler, M., & *et al.* (2000). Incidence and preventability of adverse drug events in nursing homes. *Am. J. Med.*, 109(2), 87–94.

[Guyon et al., 2002] Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3), 389–422.

[Hajduk et al., 2005] Hajduk, P. J., Huth, J. R., & Tse, C. (2005). Predicting protein druggability. *Drug Discov. Today*, 10(23-24), 1675 – 1682.

[Hammann et al., 2010] Hammann, F., Gutmann, H., Vogt, N., Helma, C., & Drewe, J. (2010). Prediction of adverse drug reactions using decision tree modeling. *Clin. Pharmacol. Ther.*, 88(1), 52–59.

[Hamosh et al., 2005] Hamosh, A., Scott, A., Amberger, J., Bocchini, C., & McKusick, V. (2005). Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders. *Nucl. Acids Res.*, 33, D514–7.

[Hastie et al., 2009] Hastie, T., Tibshirani, Robert, & Friedman, J. (2009). *Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed.* Springer.

[He & Lawrence, 2011] He, J. & Lawrence, R. (2011). A graph-based framework for multi-task multi-view learning. In *ICML'11*.

[Hert et al., 2004] Hert, J., Willett, P., Wilton, D., Acklin, P., Azzaoui, K., Jacoby, E., & Schuffenhauer, A. (2004). Comparison of topological descriptors for similarity-based virtual screening using multiple bioactive reference structures. *Org. Biomol. Chem.*, 2(22), 3256–66.

[Hopkins & Groom, 2002] Hopkins, A. L. & Groom, C. R. (2002). The druggable genome. *Nat. Rev. Drug Discov.*, 1(9), 727–730.

[Horvath et al., 2004] Horvath, T., Grtner, T., & Wrobel, S. (2004). Cyclic pattern kernels for predictive graph mining. *SIGKDD'04*, (pp. 158–167).

[Huan et al., 2003] Huan, J., Wang, W., & Prins, J. (2003). Efficient mining of frequent subgraph in the presence of isomorphism. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM'03)* (pp. 549–552).

[Jacob & Vert, 2008] Jacob, L. & Vert, J.-P. (2008). Protein-ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics*, 24(19), 2149–2156.

[Ji & Ye, 2009] Ji, S. & Ye, J. (2009). Linear dimensionality reduction for multi-label classification. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence (IJCAI'09)* (pp. 1077–1082).

[Ji et al., 2006] Ji, Z. L., Wang, Y., Yu, L., Han, L. Y., Zheng, C. J., & Chen, Y. Z. (2006). *In silico* search of putative adverse drug reaction related proteins as a potential tool for facilitating drug adverse effect prediction. *Toxic. Let.*, 164, 104–112.

[Johnson & Bootman, 1995] Johnson, J. A. & Bootman, J. L. (1995). Drug-related morbidity and mortality: A cost-of-illness model. *Arch. Intern. Med.*, 155(18), 1949–56.

[Kennedy, 1997] Kennedy, T. (1997). Managing the drug discovery/development interface. *Drug Discov. Today*, 2, 436–444.

[Kim & Xing, 2010] Kim, S. & Xing, E. P. (2010). Tree-guided group lasso for multi-task regression with structured sparsity. In *ICML'10* (pp. 543–550).

[Knox et al., 2011] Knox, C., Law, V., Jewison, T., Liu, P., & et al. (2011). DrugBank 3.0: a comprehensive resource for 'omics' research on drugs. *Nucl. Acids Res.*, (pp. D1035–41).

[Koh et al., 2007] Koh, K., Kim, S.-J., & Boyd, S. (2007). An interior-point method for large-scale l1-regularized logistic regression. *J. Mach. Lear. Res.*, 8, 1519–1555.

[Kola & Landis, 2004] Kola, I. & Landis, J. (2004). Can the pharmaceutical industry reduce attrition rates? *Nat. Rev. Drug Discov.*, 3(8), 711–716.

[Krishnapuram et al., 2004] Krishnapuram, B., Williams, D., Xue, Y., Hartemink, A., Carin, L., & Figueiredo, M. (2004). On semi-supervised classification. In *Proceedings of the 18th Annual Conference on Neural Information Processing Systems*, NIPS'04.

[Kuhn et al., 2009] Kuhn, M., Szklarczyk, D., Franceschini, A., Campillos, M., von Mering, C., Jensen, L. J., Beyer, A., & Bork, P. (2009). STITCH 2: an interaction network database for small molecules and proteins. *Nucl. Acids Res.*

[Kuhn et al., 2010] Kuhn, M., Szklarczyk, D., Franceschini, A., Campillos, M., von Mering, C., Jensen, L. J., Beyer, A., & Bork, P. (2010). STITCH 2: an interaction network database for small molecules and proteins. *Nucleic Acids Research*, 38(suppl 1), D552–D556.

[Kuhn et al., 2008] Kuhn, M., von Mering, C., Campillos, M., Jensen, L. J., & Bork, P. (2008). STITCH: interaction networks of chemicals and proteins. *Nucleic Acids Research*, 36(suppl 1), D684–D688.

[Kuramochi & Karypis, 2004] Kuramochi, M. & Karypis, G. (2004). An efficient algorithm for discovering frequent subgraphs. *IEEE Trans. Knowl. Data Eng.*, 16(9), 1038–51.

[Lasser et al., 2002] Lasser, K. E., Allen, P. D., Woolhandler, S. J., Himmelstein, D. U., Wolfe, S. M., & Bor, D. (2002). Timing of new black box warnings and withdrawals for prescription medications. *J. Am. Med. Assoc.*, 287, 2125–2200.

[Lazarou et al., 1998] Lazarou, J., Pomeranz, B., & Corey, P. (1998). Incidence of adverse drug reactions in hospitalized patients: a meta-analysis of prospective studies. *J. Am. Med. Asso.*, 279, 1200–5.

[Leach, 2001] Leach, A. R. (2001). Prentice Hall, Englewood Cliffs, NJ.

[Leape et al., 1991] Leape, L. L., Brennan, T. A., Laird, N., Lawthers, A. G., Localio, A. R., Barnes, B. A., & *et al.* (1991). The nature of adverse events in hospitalized patients. results of the harvard medical practice study II. *N. Eng. J. Med.*, 324(6), 377–384.

[Leone et al., 2008] Leone, R., Sottosanti, L., Luisa Iorio, M., Santuccio, C., Conforti, A., Sabatini, V., Moretti, U., & Venegoni, M. (2008). Drug-related deaths: an analysis of the italian spontaneous reporting database. *Drug Saf.*, 31(8), 703–713.

[Li et al., 2005] Li, H., Ung, C. Y., Yap, C. W., Xue, Y., Li, Z. R., Cao, Z. W., & Chen, Y. Z. (2005). Prediction of genotoxicity of chemical compounds by statistical learning methods. *Chem. Res. Toxicol.*, 18, 1071–80.

[Li & Lai, 2007] Li, Q. & Lai, L. (2007). Prediction of potential drug targets based on simple sequence properties. *BMC Bioinformatics*, 8(1), 353.

[Li et al., 2006] Li, Z. R., Lin, H. H., Han, L. Y., Jiang, L., Chen, X., & Chen, Y. Z. (2006). PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucl. Acids Res.*, 34, W32–37.

[Lin et al., 2008] Lin, S. F., Xiao, K. T., Huang, Y. T., & Soo, V. W. (2008). A tool for finding possible explanation for adverse drug reactions through drug and drug target interactions. In *The 2008 International Conference on Biomedical Engineering and Informatics,* (pp. 580–4).

[Liu et al., 2010] Liu, Q., Xu, Q., Zheng, V., Xue, H., Cao, Z., & Yang, Q. (2010). Multi-task learning for cross-platform sirna efficacy prediction: an in-silico study. *BMC Bioinformatics*, 11, 181–196.

[Liu et al., 2007] Liu, T., Lin, Y., Wen, X., Jorissen, R. N., & Gilson, M. K. (2007). BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities. *Nucl. Acids Res.*, 35, D198–D201.

[MacDonald et al., 2007] MacDonald, M. L., Lamerdin, J., Owens, S., Keon, B. H., Bilter, G. K., & *et al.* (2007). Identifying off-target effects and hidden phenotypes of drugs in human cells. *Nat. Chem. Biol.*, 2, 329–337.

[Mason et al., 2000] Mason, L., Baxter, J., Bartlett, P., & Frean, M. (2000). Boosting algorithms as gradient descent. In *Advances in Neural Information Processing Systems 12* (pp. 512–518).

[Matthies & Strang, 1979] Matthies, H. & Strang, G. (1979). The solution of non linear finite element equations. *Int. J. Numer. Meth. Eng.*, 14, 1613–1626.

[McCallum et al., 2000] McCallum, A. K., Nigam, K., Rennie, J., & Seymore, K. (2000). Automating the construction of internet portals with machine learning. *Information Retrieval*, 3, 127–163.

[Mirams et al., 2011] Mirams, G. R., Cui, Y., Sher, A., Fink, M., Cooper, J., Heath, B. M., McMahon, N. C., Gavaghan, D. J., & Noble, D. (2011). Simulation of multiple ion channel block provides improved early prediction of compounds' clinical torsadogenic risk. *Cardiovasc. Res.*

[Moore et al., 2007] Moore, T., Cohen, M., & Furberg, C. (2007). Serious adverse drug events reported to the Food and Drug Administration. *Arch. Intern. Med.*, 167, 1752–1759.

[Moss, 2006] Moss, A. J. (2006). Drug-induced QT prolongation: an update. *Ann. Noninvasive Electrocardiol.*, 11(1), 1–2.

[Mushiroda et al., 2005] Mushiroda, T., Saito, S., Tanaka, Y., Takasaki, J., Kamatani, N., Beck, Y., Tahara, H., Nakamura, Y., & Ohnishi, Y. (2005). A model of prediction system for adverse cardiovascular reactions by calcineurin inhibitors among patients with renal transplants using gene-based single-nucleotide polymorphisms. *J. Hum. Genet.*, 50, 442–447.

[Muslea et al., 2002] Muslea, I., Minton, S., & Knoblock, C. A. (2002). Adaptive view validation: A first step towards automatic view detection. In *Proceedings of ICML'02* (pp. 443–450).

[Nacher et al., 2001] Nacher, M., Treeprasertsuk, S., Singhasivanon, P., Silachamroon, U., & et al. (2001). Association of hepatomegaly and jaundice with acute renal failure but not with cerebral malaria in severe falciparum malaria in thailand. *Am. J. Trop. Med. Hyg.*, 65(6), 828–33.

[Need et al., 2005] Need, A., A.G., M., & Goldstein, D. (2005). Priorities and standards in pharmacogenetic research. *Nat. Genet.*, 37, 671–681.

[Needleman & Wunsch, 1970] Needleman, S. B. & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48(3), 443–453.

[Newman, 2005] Newman, D. G. (2005). Diabetes mellitus and its effects on pilot performance and flight safety: A review. *Aviation Research Investigation Report*, (pp. B2005–7).

[Nigam & Ghani, 2000] Nigam, K. & Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. In *CIKM'00* (pp. 86–93).

[Ning et al., 2009] Ning, X., Rangwala, H., & Karypis, G. (2009). Multi-assay-based structure-activity relationship models: improving structure-activity relationship models by incorporating activity information from related targets. *J. Chem. Inf. Model.*, 49, 2444–2456.

[Nisius & Goller, 2009] Nisius, B. & Goller, A. H. (2009). Similarity-based classifier using topomers to provide a knowledge base for hERG channel inhibition. *J. Chem. Inf. Model.*, 49(2), 247–256.

[Nocedal, 1980] Nocedal, J. (1980). Updating quasi-newton matrices with limited storage. *Math. Comput.*, 35, 773–782.

[Obama et al., 2005] Obama, K., Ura, K., Li, M., Katagiri, T., & et al. (2005). Genome-wide analysis of gene expression in human intrahepatic cholangiocarcinoma. *Hepatology*, 41(6), 1339–1348.

[Onder et al., 2002] Onder, G., Pedone, C., Landi, F., Cesari, M., Della Vedova, C., Bernabei, R., & *et al.* (2002). Adverse drug reactions as cause of hospital admissions: results from the italian group of pharmacoepidemiology in the elderly (GIFA). *J. Am. Geriatr. Soc.*, 50(12), 1962–68.

[Ouillé et al., 2011] Ouillé, A., Champéroux, P., Martel, E., Ferro, F., Bouard, D., Richard, S., Richard, S., & Guennec, J.-Y. L. (2011). Ion channel blocking profile of compounds with reported torsadogenic effects: what can be learned? *Br. J. Pharmacol.*

[Paolini et al., 2006] Paolini, G., Shapland, R., van Hoorn, W., Masonn, J., & Hopkins, A. (2006). Global mapping of pharmacological space. *Nat. Biotechnol.*, 24, 805–815.

[Puniyani et al., 2010] Puniyani, K., Kim, S., & Xing, E. P. (2010). Multi-population GWA mapping via multi-task regularized regression. *Bioinformatics*, 26(12), 208–216.

[Ravasz et al., 2002] Ravasz, E., Somera, A., Mongru, D., Oltvai, Z., & Barabasi, A.-L. (2002). Hierarchical organization of modularity in metabolic networks. *Science*, 297, 1551–1555.

[Redfern, 2003] Redfern, W. e. (2003). Relationships between preclinical cardiac electrophysiology, clinical qt interval prolongation and torsade de pointes for a broad range of drugs: evidence for a provisional safety margin in drug development. *Cardiovasc. Res.*, 58(1), 32–45.

[Robert & Goodnow, 2006] Robert, A. & Goodnow, J. (2006). Hit and lead identification: integrated technology-based approaches. *Drug Discov. Today: Technol.*, 3(4), 367–375.

[Rockey & Elcock, 2002] Rockey, W. M. & Elcock, A. H. (2002). Progress toward virtual screening for drug side effects. *Proteins Struct. Funct. Bioinf.*, 48, 664–671.

[Rogers et al., 2005] Rogers, D., Brown, R., & Hahn, M. (2005). Using extended-connectivity fingerprints with laplacian-modified bayesian analysis in high-throughput screening. *J. Biomolecular Screening*, 10(7), 682–686.

[Rual, 2005] Rual, J.-F. e. (2005). Towards a proteome-scale map of the human protein-protein interaction network. *Nature*, 437, 1173–1178.

[Sams-Dodd, 2005] Sams-Dodd, F. (2005). Target-based drug discovery: is something wrong? *Drug Discov. Today*, 10(2), 139–147.

[Scheiber et al., 2009] Scheiber, J., Chen, B., Milik, M., Sukuru, S. C., Bender, A., & *et al.* (2009). Gaining insight into off-target mediated effects of drug candidates with a comprehensive systems chemical biology analysis. *J. Chem. Inf. Model.*, 49, 308–317.

[Schneider et al., 2004] Schneider, G., Filimonov, D., & *et al.* (2004). *Chemogenomics in Drug Discovery: A Medicinal Chemistry Perspective*. Willey-VCH.

[Schneider et al., 2008] Schneider, G., Filimonov, D., & *et al.* (2008). *Chemoinformatics Approaches to Virtual Screening*. The Royal Society of Chemistry, London, UK.

[Schrier & Wang, 2004] Schrier, R. W. & Wang, W. (2004). Acute renal failure and sepsis. *New Eng. J. Med.*, 351(2), 159–169.

[Sindhwani & Niyogi, 2005] Sindhwani, V. & Niyogi, P. (2005). A co-regularized approach to semi-supervised learning with multiple views. In *Proceedings of the ICML Workshop on Learning with Multiple Views*.

[Sindhwani & Rosenberg, 2008] Sindhwani, V. & Rosenberg, D. S. (2008). An RKHS for multi-view learning and manifold co-regularization. In *Proceedings of the 25th international conference on Machine learning*, ICML'08 (pp. 976–983).

[Smalter et al., 2008] Smalter, A., Huan, J., & Lushington, G. (2008). Pattern diffusion graph kernel for chemical compound classificationfrom lasso regression to feature vector. In *Proceedings of IWDMB'08*.

[Sokolov & Ben-Hur, 2011] Sokolov, A. & Ben-Hur, A. (2011). Multi-view prediction of protein function. In *ACM BCB'11*.

[Soläng et al., 1999] Soläng, L., Malmberg, K., & Rydén, L. (1999). Diabetes mellitus and congestive heart failure. *Euro. Heart J.*, 20, 789–795.

[Stelzl & *et al.*, 2005] Stelzl, U. & *et al.* (2005). A human protein-protein interaction network: A resource for annotating the proteome. *Cell*, 122(6), 957 – 968.

[Stephenson et al., 2005] Stephenson, V., Heyding, R., & Weaver, D. (2005). The “promiscuous drug concept” with applications to alzheimer’s disease. *FEBS Letters*, 579, 1338–1342.

[Sutherland et al., 2004] Sutherland, J., O’Brien, L., & Weaver, D. (2004). A comparison of methods for modeling quantitative structure-activity relationships. *J. Med. Chem.*, 47(22), 5541–54.

[Tatonetti et al., 2009] Tatonetti, N. P., Liu, T., & Altman, R. B. (2009). Predicting drug side-effects by chemical systems biology. *Genome Biol.*, 10(9), 238.

[Tibshirani et al., 2005] Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., & Knight, K. (2005). Sparsity and smoothness via the fused lasso. *J. Royal Stat. Soc. - Ser. B: Stat. Method.*, 67(1), 91–108.

[Tropsha, 2010] Tropsha, A. (2010). Best practices for qsar model development, validation, and exploitation. *Mol. Inf.*, 29, 476–488.

[Tsoumakas & Katakis, 2007] Tsoumakas, G. & Katakis, I. (2007). Multi-label classification: an overview. *Int. J. Data Warehousing and Mining*, 579, 1–13.

[Ursem et al., 2009] Ursem, C. J., Kruhlak, N. L., Contrera, J. F., MacLaughlin, P. M., & et al. (2009). Identification of structure-activity relationships for adverse effects of pharmaceuticals in humans. Part A. *Regul. Toxicol. Pharm.*, 54(1), 1–22.

[van der Hooft et al., 2006] van der Hooft, C. S., Sturkenboom, M. C., van Grootheest, K., Kingma, H. J., & Stricker, B. H. (2006). Adverse drug reaction-related hospitalizations: a nationwide study in the netherlands. *Drug Saf.*, 29(2), 161–168.

[Viswanadhan et al., 1989] Viswanadhan, V., Ghose, A., Revankar, G., & Robins, R. (1989). Atomic physicochemical parameters for three-dimensional structure directed quantitative structure-activity relationships. *J. Chem. Inf. Comput. Sci.*, 29, 163–172.

[Wale et al., 2007] Wale, N., Watson, I., & Karypis, G. (2007). Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information System*, 1, 1–29.

[Wang & FitzGerald, 2010] Wang, M. & FitzGerald, G. A. (2010). Cardiovascular biology of microsomal prostaglandin e synthase-1. *Trends Cardiovas. Med.*, 20(6), 189–195.

[Wang & Zhou, 2007] Wang, W. & Zhou, Z. H. (2007). Analyzing co-training style algorithms. In *Proceedings of ECML'07* (pp. 454–465).

[Wang et al., 2009] Wang, X., Zhang, C., & Zhang, Z. (2009). Boosted multi-task learning for face verification with applications to web image and video search. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 142–149).

[Whitebread et al., 2005] Whitebread, S., Hamon, J., Bojanic, D., & Urban, L. (2005). Keynote review: in vitro safety pharmacology profiling: an essential tool for successful drug development. *Drug Discov. Today*, 10, 1421–33.

[Wishart et al., 2008] Wishart, D. S., Knox, C., Guo, A. C., Cheng, D., Shrivastava, S., Tzur, D., Gautam, B., & Hassanali, M. (2008). DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucl. Acids Res.*, 36, D901–906.

[Wishart et al., 2006] Wishart, D. S., Knox, C., Guo, A. C., Shrivastava, S., Hassanali, M., Stothard, P., Chang, Z., & Woolsey, J. (2006). DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucl. Acids Res.*, 34, D668–672.

[Woosley, 2003] Woosley, R. L. (2003). Drugs that prolong the QT interval and/or induce torsades de pointes ventricular arrhythmia. `http://www.azcert.org/medical-pros/drug-lists/printable-drug-list.cfm`.

[Wu et al., 2010] Wu, F., Han, Y., Tian, Q., & Zhuang, Y. (2010). Multi-label boosting for image annotation by structural grouping sparsity. In *Proceedings of the International Conference on Multimedia (MM'10)* (pp. 15–24).

[Xie et al., 2007] Xie, L., Wang, J., & Bourne, P. E. (2007). *In silico* elucidation of the molecular mechanism defining the adverse effect of selective estrogen receptor modulators. *PLoS Comput. Biol.*, 3(11), e217.

[Xu et al., 2010] Xu, Q., Pan, S., Xue, H., & Yang, Q. (2010). Multitask learning for protein subcellular location prediction. *IEEE/ACM Trans. Comp. Biol. and Bioinfo.*, 99, 748–759.

[Yao, 2008] Yao, X. e. (2008). Predicting QT prolongation in humans during early drug development using herg inhibition and an anaesthetized guinea-pig model. *Br. J. Pharmacol.*, 154(7), 1446–1456.

[Yap et al., 2004] Yap, C. W., Cai, C. Z., Xue, Y., & Chen, Y. Z. (2004). Prediction of torsade-causing potential of drugs by support vector machine approach. *Toxicological Sciences*, 79, 170–177.

[Yildirim et al., 2007] Yildirim, M. A., Goh, K.-I., Cusick, M. E., Barabasi, A.-L., & Vidal, M. (2007). Drug-target network. *Nat. Biotech.*, 25(10), 1119–1126.

[Yu et al., 2007] Yu, S., Krishnapuram, B., Rosales, R., Steck, H., & Rao, R. B. (2007). Bayesian co-training. In *Proceedings of the 21th Annual Conference on Neural Information Processing Systems*, NIPS'07.

[Zerhouni, 2003] Zerhouni, E. (2003). The NIH roadmap. *Science*, 302(5642), 63–72.

[Zhang & Huan, 2010] Zhang, J. & Huan, J. (2010). Comparison of chemical descriptors for protein-chemical interaction prediction. *Int. J. Comput. Biosci.*, 1(1), 13–21.

[Zhang et al., 2007] Zhang, J.-X., Huang, W.-J., Zeng, J.-H., Huang, W.-H., & et al. (2007). DITOP: drug-induced toxicity related protein database. *Bioinformatics*, 23(13), 1710–1712.

[Zhang et al., 2010] Zhang, K., Gray, J. W., & Parvin, B. (2010). Sparse multitask regression for identifying common mechanism of response to therapeutic targets. In *Proceedings of 18th Annual International Conference on Intelligent Systems for Molecular Biology* (pp. 97–105).

[Zheng et al., 2006] Zheng, C. J., Han, L. Y., Yap, C. W., Ji, Z. L., Cao, Z. W., & Chen, Y. Z. (2006). Therapeutic Targets: Progress of Their Exploration and Investigation of Their Characteristics. *Pharmacological Reviews*, 58(2), 259–279.

[Zhu et al., 2009] Zhu, M., Gao, L., Li, X., & Liu, Z. (2009). Identifying drug-target proteins based on network features. *Sci. in China Ser. C: Life Sci.*, 52(4), 398–404.