Learning with Low-Quality Data: Multi-View Semi-Supervised Learning with Missing Views

By

Brian Quanz

Submitted to the Department of Electrical Engineering and Computer Science 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

Luke Huan, Chairperson

Xue-wen Chen

Committee members

Victor Frost

Bo Luo

Brian Potetz

Zsolt Talata

Date defended: 7/24/2012

The Dissertation Committee for Brian Quanz certifies
that this is the approved version of the following dissertation :

Learning with Low-Quality Data: Multi-View Semi-Supervised Learning with Missing Views

Luke Huan, Chairperson

Date approved: 7/24/2012

Abstract

The focus of this thesis is on learning approaches for what we call “low-quality data” and in particular data in which only small amounts of labeled target data is available. The first part provides background discussion on low-quality data issues, followed by preliminary study in this area. The remainder of the thesis focuses on a particular scenario: multi-view semi-supervised learning.

Multi-view learning generally refers to the case of learning with data that has multiple natural views, or sets of features, associated with it. Multi-view semi-supervised learning methods try to exploit the combination of multiple views along with large amounts of unlabeled data in order to learn better predictive functions when limited labeled data is available.

However, lack of complete view data limits the applicability of multi-view semi-supervised learning to real world data. Commonly, one data view is readily and cheaply available, but additionally views may be costly or only available in some cases. This thesis work aims to make multi-view semi-supervised learning approaches more applicable to real world data specifically by addressing the issue of missing views through both feature generation and active learning, and addressing the issue of model selection for semi-supervised learning with limited labeled data.

This thesis introduces a unified approach for handling missing view data in multi-view semi-supervised learning tasks, which applies to both data with completely missing additional views and data only missing views in some instances. The idea is to learn a feature generation function mapping one view to another with the mapping biased to encourage the features generated to be useful for multi-view semi-supervised learning algorithms. The mapping is then used to fill in views as pre-processing. Unlike previously proposed single-view multi-view learning approaches, the proposed approach is able to take advantage of additional view data when available, and for the case of partial view presence is the first feature-generation approach specifically designed to take into account the multi-view semi-supervised learning aspect.

The next component of this thesis is the analysis of an active view completion scenario. In some tasks, it is possible to obtain missing view data for a particular instance, but with some associated cost. Recent work has shown an active selection strategy can be more effective than a random one. In this thesis, a better understanding of active approaches is sought, and it is demonstrated that the effectiveness of an active selection strategy over a random one can depend on the relationship between the views.

Finally, an important component of making multi-view semi-supervised learning applicable to real world data is the task of model selection, an open problem which is often avoided entirely in previous work. For cases of very limited labeled training data the commonly used cross-validation approach can become ineffective. This thesis introduces a re-training alternative to the method-dependent approaches similar in motivation to cross-validation, that involves generating new training and test data by sampling from the large amount of unlabeled data and estimated conditional probabilities for the labels.

The proposed approaches are evaluated on a variety of multi-view semi-supervised learning data sets, and the experimental results demonstrate their efficacy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In data mining or machine learning, a fundamental goal is to be able to predict some quantity of interest about some data based on computational representations of the data with measurable features for each instance of the data. For instance we might want to predict the categories present in an image such as "car" or "fish" based on features of the image such as texture or shape descriptors or whether or not a certain chemical has a toxic (carcinogenic) effect in humans based on its chemical structure and in-vitro lab tests. Data mining and machine learning methods try to look at collected sets of data called training data, e.g., images or chemicals, that are annotated with ground truth, or “label”, information about some property of interest for each data instance, e.g., image category or toxicity, in order to estimate, or *learn*, the relationship between the representations of the data and the labels. In an ideal scenario, collected data is *high-quality*. That is an abundant amount of labeled data is fully available, all from the target data source of interest. In the ideal high-quality data case, labeled data is abundant - so that predictors can be estimated with high confidence, the labeled data is all from the same fixed source as the data for the target task, all the features of the data are available in all instances, data instances are independent, and there are no erroneous data or outliers - extreme values not representative of the data which can mislead learning algorithms. Unfortunately, such ideal high-quality data scenarios are rarely encountered in real-world applications due to error, difficulty, and cost associated with collecting and annotating data. Typically data have one or more of the following low-quality aspects.

- Only a small sample of labeled data is available from the target data.
- The data is only partially observed - i.e., there are missing values.
- There are errors, outliers, or noise present in the data and annotations.
- The distribution of the target data is not the same as the distribution of the training data, so that the relationships learned in the collected data may not be accurate for the target data. This includes such issues as concept drift where the target data distribution changes over time, and sample selection bias where the collected data sample is not representative of the target data sample.

The focus of this thesis is on the first case, of limited labeled data. It is often the case that only a limited amount of labeled data can be collected for new tasks, due to such factors as time and cost. When labeled data is limited, it becomes more important to make use of any additional sources of information available - which can be in the form of different but related sets of data that are fully labeled, different representations of the data (sets of data features), information about the relationships between features of the data, and unlabeled data from the target data source. In general, the type of low-quality issues along with the specific form of auxiliary information available, whether data or some type of prior knowledge, determines the specific learning problem. For instance when little or no labeled data is available from the target data distribution, but a different set of high-quality labeled data is available from a related distribution, it may be desirable to make use of this data in learning a predictive model for the target data, in some sense transferring knowledge from one task to a related one. This corresponds to both issues of limited labeled data and differing data distributions. The same issue arises if the data are unavoidably different, as is the case with concept drift. Both of these cases correspond to the problem of *transfer learning* [142], and the auxiliary information available comes in the form of the related high-quality data. My previous work in this area focused on how to learn a predictive model using related but different training data along with unlabeled target data that could then be applied to the target data [148], and also how to find an embedding for training and target data that would align the data distributions and ideally remove the low-quality aspects from the data as a type of pre-processing [149, 150]. Another line of my previous work with limited labeled data is on utilizing auxiliary information in the form of a known relationship between features of the data [147, 66]. These works are discussed chronologically in the first part (the next three chapters) of this thesis, comprising preliminary study on learning with low-quality data, and learning with limited labeled data in particular.

The main focus of this thesis, multi-view semi-supervised learning, corresponds to a different learning problem for the case of small amounts of labeled training data. There are two key types of auxiliary information associated with multi-view semi-supervised learning. The first corresponds to prior knowledge about the features of the data - in the form of a natural partition of the features, such that each partitioned set is sufficient for learning (as explained in Section 1.2) and also such that the views are not entirely dependent on each other so that some different information is potentially available. The second corresponds to the semi-supervised learning aspect, learning when an additional, usually large, set of unlabeled data is available. This thesis can be seen as also addressing an additional low-quality data aspect often associated with multi-view semi-supervised learning in real world applications - that of structured missing values in the form of missing views. That is, some data instances may be completely missing additional views of the data.

The remainder of this chapter proceeds as follows. First, more detail and background are provided in Sections 1.1 and 1.2. Next, the motivation behind the main focus of this thesis is described in Section 1.3. In Section 1.4, the contributions of this thesis are described. In the last section, Section 1.5, the organization for the remainder of the thesis is given.

1.1 Supervised and Semi-Supervised Learning

The general goal of machine learning is to learn a predictive function $f : \mathscr{X} \to \mathscr{Y}$ mapping an input data space $\mathscr{X}$ to an output label space $\mathscr{Y}$ using a set of training data examples. The characteristics of the label space for a learning problem determine the corresponding machine learning task, for instance if $\mathscr{Y}$ is fixed and finite the task corresponds to classification and if $\mathscr{Y} \equiv \mathbb{R}$ the task corresponds to regression. Supervised learning addresses the case where a training data set consists of a set of data and label pairs, $(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n) \in \mathscr{X} \times \mathscr{Y}$. In order to employ supervised learning, data must be collected and annotated with labels, usually by a human. In many scenarios, unlabeled data examples are abundant but obtaining labeled data for a target learning task can be error-prone, time-consuming, expensive, or even impossible. Semi-supervised learning approaches aim to make use of the available unlabeled data to improve the predictive performance of the learned function, particularly in cases where the amount of labeled training data is small. Specifically, in addition to the training examples, a set of unlabeled training instances, $x_{n+1},x_{n+2},\ldots,x_{n+m} \in \mathscr{X}$, is available. While the unlabeled data alone do not provide any information about the predictive function mapping, the combination of the unlabeled data, specific assumptions about the data, and the limited labeled data can make it possible to learn a function with improved predictive performance compared to a function learned using only the limited labeled training data [224]. Typically this improvement is possible through a reduction in some sense of the size of the hypothesis space for the predictive function [224]. A main category of semi-supervised learning methods, and the focus of this thesis, is multi-view semi-supervised learning.

1.2 Multi-View Learning and Multi-View Semi-Supervised Learning

Multi-view learning generally addresses the case of learning with data that has multiple natural views, generally corresponding to distinct sets of features, associated with it. Specifically $x \in \mathscr{X}$ can be naturally represented as $x = (x^1,x^2,\ldots,x^k) \in \mathscr{X}^1 \times \mathscr{X}^2 \times \ldots \times \mathscr{X}^k$, corresponding to k different views of the data. For example, when classifying webpages, two natural views for a given webpage could be considered: the set of text features for any text on the webpage, and the set of link text features for any links to the webpage. Another example is chemical data. The set of chemical structure features could correspond to one view and chemical-protein interaction profiles could correspond to a second view. Multi-view semi-supervised learning methods try to exploit the combination of multiple views with associated assumptions along with large amounts of unlabeled data in order to learn better predictive functions when limited labeled data is available. The fundamental idea exploited for multi-view semi-supervised learning is the idea of predictive function agreement (consensus) of view-specific functions' predictions on the unlabeled data. If for each view a function from an associated hypothesis class exists that can achieve zero prediction error, restricted to that view, then all of these functions from different views must agree exactly in their predictions on all data instances, in particular the unlabeled data instances. Therefore, when learning the predictive functions for the views, any combination of functions that disagree in their predictions on the unlabeled data can be eliminated from consideration. In this way, the size of the set of hypothesis functions that explain the labeled data well in each view can potentially be reduced. In the more realistic case that the best performing functions in each view have some base error, as long as the error is not too great there will still necessarily be overlap between these functions' predictions even if they do not universally agree on all instances [58]. In this case the solutions can still be biased toward predictors that mostly agree on the unlabeled data instances. The condition that for each individual view there exists an associated function from a given hypothesis class that is able to achieve the best possible error rate is referred to as *view sufficiency*.

1.3 Motivation

Multi-view data arises naturally in many applications. However, lack of complete view data limits the applicability of multi-view semi-supervised learning to real world data. A common scenario is that one data view is readily and cheaply available, but additional views may only be available in some cases and may be costly to obtain.

This proposed work aims to make multi-view semi-supervised learning approaches more applicable to real world data specifically by addressing the issue of missing views.

1.3.1 Some Motivating Examples

The following are some detailed examples of potential applications that fit the multi-view semi-supervised learning scenario, with missing views being an issue.

1.3.1.1 Medical Diagnostics

In terms of medical diagnosis, in particular cancer diagnosis, prognosis prediction both before and after treatments can be cast as a multi-view semi-supervised learning problem. For instance, if the goal is survival prediction, since the data is censored ground truth labels are not obtainable for many patients. If the goal is to determine pathologic complete response, potentially invasive surgical procedures are required which furthermore are not entirely accurate, making ground truth labels difficult to obtain. Additional views for patients can be obtained but these can be both costly and inconvenient for the patients. For disease diagnosis in general, in many cases there is no definitive test for a disease, or the disease can only be determined with more certainty after many expensive tests such as ultrasound, MRI, and biopsy or after analyzing the results of different treatments. For instance, a common test for elevated thyroid stimulating hormone levels could indicate hypothyroidism, a pituitary adenoma, or a number of auto-immune diseases, with no reliable single test to determine the underlying cause. Obtaining all sets of views for all patients is prohibitively costly and in some case impossible, as is the case with obtaining label information. Ideally, a diagnostic system could aid doctors by considering all partial view information available and including undiagnosed patient information. This problem also motivates an active solution where expensive and invasive procedures are only carried out if necessary. On the other hand, there are some common sets of easily obtainable clinical features which would correspond to a view present for all patients related to a particular disease. For instance, for lung cancer, common clinical factors include forced expiratory volume, performance status, and gender.

Recently an active multi-view semi-supervised learning approach was applied to data for long cancer survival prediction and pathologic complete response prediction for chemo-radiotherapy treatment, with promising results [209]. In these experiments, additional views were provided for individual patients by imaging techniques like PET/CT scanning.

1.3.1.2 Cheminformatics

For prediction tasks involving chemicals, molecular structure features based on chemical graphs can be readily obtained, but obtaining chemical-protein interaction profiles for a set of proteins can be costly and time-consuming. Other expensive or difficult to obtain views include general in-vitro tests and bio-assay screening, and various more complete characterizations of structure, such as the results of nuclear magnetic resonance and x-ray crystallography. Additionally, labels are also difficult to obtain, particularly when the goal is to evaluate new chemical compounds, for the purpose of drug discovery and evaluation. If the final goal is to predict whether or not a chemical would make an effective and safe drug, the amount of labeled data is limited. Another goal is to determine side effects for a chemical compound, since so few drugs make it to the clinical trial phase there is only a limited amount of data available about the side effects of drugs. Another example is with chemical toxicity prediction, an earlier step in the drug discovery process. In this case, reliable end-points are usually determined using animal studies which are both expensive and time-consuming, and also not entirely accurate.

A small set of complete data has been used with multi-view semi-supervised learning for adverse drug effect predictions [54], but for new chemicals or chemical groups additional views will generally not be readily available.

1.3.1.3 Webpage Data

Webpage data potentially contain many views, which may or may not be present in a given instance, including images, sounds, and information about incoming links. A standard view that is always present is the text on the webpage itself. Additionally, classifying webpages manually would involve hiring human annotators; the process would be time consuming and expensive, and error-prone due both to human error and the ambiguity of assigning a class to a webpage in some cases. Furthermore, new classification tasks are constantly arising as the result of user-specific preferences and search. For instance, a user's particular preferences about what kind of webpages he or she likes and also what webpages are relevant to a particular semantic search correspond to prediction tasks with little to no labeled instances. More generally, this idea applies to personalized prediction of other kinds as well, for instance such as for personalized product recommendation.

Considering in particular the additional view associated with the text of links on other pages linking to a given webpage, the availability of this view is also limited. As an example, the WebKB data presented in the first work on co-training [25] and used in subsequent work [220, 210] uses text features for text on a webpage as one view, and text features from the incoming link text as a second view. This second view is actually incomplete even in the WebKB data set, but the incomplete view instances are just removed for the purposes of the experiments. For instance, for the faculty vs. student classification task, about half of the webpages in each category do not have any incoming links. However it is likely other pages do link to these, just that the crawler used to collect the webpages did not find them in its finite search. Additionally, as new webpages are created initially no incoming link information will be available, and existing webpages being updated also changes this information; this may lead to misleading representation in the link view if the same procedure is used for generating this view.

1.3.1.4 Multimedia Data

Another category of examples is with multi-media data, for example, tagged and annotated multimedia data such as tagged images. In this case the annotation or tagging can be sporadic and noisy, in the sense that tags may not necessarily correspond to categories present in media or desired categories. Taking tagged images as an example, when available, tags may provide highly relevant information as to the categories of objects or concepts captured in an image, but as annotators cannot be obtained to annotate every image or new images, ideally it would be preferable to be able to use tag information when available to improve a classifier for the single image view. Additionally new classification tasks are likely to arise, limiting the amount of labeled data available in such cases, for instance, as with webpage classification for each user there may be multiple new classification tasks defined, characterizing a particular type of image he or she is looking for based on high-level concepts.

1.3.2 Motivation from Theoretical Work

In order to determine what kind of bias to assert when trying to estimate missing views, a key motivation for this thesis comes from theoretical study of multi-view semi-supervised learning. As mentioned in Section 1.2, if each view is sufficient then multi-view semi-supervised learning may offer some benefit, but another condition is necessary to determine whether or not it will offer a benefit. Theoretical work characterizing what conditions are sufficient for multi-view semi-supervised learning to succeed in improving predictive performance is a key motivation for the proposed approach of this thesis for handling missing view data, and discussed in more detail in Chapter 6. In short, conditions of expansion [9], and differences in empirical kernel maps using the unlabeled data [179] are connected in characterizing how the labeled and unlabeled data are related to each other in different views. These works motivate the idea of this thesis of using the difference between the distance profiles with respect to the unlabeled data in each view for determining if pairs of views provide sufficiently complementary information when evaluating candidate values for filling in missing views, and for estimating the utility of completing an instance for active view completion. This motivates the feature generation (Chapter 6) and active view completion (Chapter 7) approaches of this thesis work.

1.4 Contributions

This analysis of the commonality of theoretical results on multi-view semi-supervised learning leads to the first proposed contribution of this thesis: a novel way of biasing the values selected for missing views so that the filled in values will be useful for multi-view semi-supervised learning algorithms. A unified approach for handling missing view data in multi-view semi-supervised learning tasks is introduced, which applies to the complete range of missing view data. The idea is to use the criteria for the success of multi-view semi-supervised learning algorithms to bias a feature generation function mapping one view to another. This is carried out using additional terms in the objective function of a feature generation network model that encourages the data instances in distinct views to be nearby different unlabeled instances, and also takes into account classification performance for the generated data. The proposed approach can be seen as a pre-processing step that fills in missing views, and so allows a user's choice of multi-view semi-supervised learning algorithms to be applied to the completed multi-view data. Unlike previously proposed single-view multi-view learning approaches, the proposed approach is able to take advantage of additional view data when available, and for the case of partial view presence is the first feature-generation approach specifically designed to take into account the multi-view semi-supervised learning aspect.

The second contribution of my thesis is the analysis of the active view completion scenario, which can be an alternative approach for semi-supervised learning depending on the application. In some tasks, it is possible to obtain missing view data for a particular instance, but with some associated cost, for example, an annotator could be hired to label an image, or a PET/CT scan could be ordered for a patient. Recent work has shown for some data that an active selection strategy can result in faster predictive performance improvement than when instances are randomly selected for view completion [209]. However this work does not consider at all when an active strategy may or may not be useful, and additionally the methods proposed for active selection are not directly applicable to multi-view semi-supervised learning methods in general, as they require, for example, estimates of predictive variance. In this thesis, different selection strategies are analyzed and it will be demonstrated that the effectiveness of an active selection strategy over a random one can depend greatly on the relationship between the views. Additionally a simple active selection approach is proposed for which improved performance is demonstrated in the experimental study.

The final contribution of this thesis is on model selection for semi-supervised learning algorithms with limited labeled data. An important component of making multi-view semi-supervised learning applicable to real world data is the task of model selection, which is often avoided entirely in previous work and excluded from consideration. For cases of very limited labeled training data such as those commonly encountered with multi-view semi-supervised learning scenarios, model selection is a significant challenge, and listed as a key open problem in a recent survey [78]. With missing views this task potentially becomes even more difficult since additional hyper-parameters may need to be selected for the pre-processing step. Experimental results have demonstrated the benefit of multi-view semi-supervised learning in cases of very limited labeled training data (e.g., [220, 25, 179]), but in order for such results to be achievable in practice, some practical method of selecting the hyper-parameters for these methods is necessary. The widely used cross-validation approach can become ineffective with too few labeled training instances [176], and the majority of other proposed model selection methods are specific to the corresponding proposed algorithms and frameworks. For instance one such approach is a marginal likelihood approach, in which hyper-parameter estimation is achieved by numerical procedures attempting to approximately integrate out the model parameters from a particular Bayesian probabilistic model for multi-view semi-supervised learning, and maximizing this marginal likelihood with respect to the hyper-parameters [209] (also called type II maximum likelihood or evidence-based approach). However this requires assuming a particular probabilistic model for the different components of the model and the data, so there is no straight-forward way to apply this approach to, for instance, the iterative co-training algorithm (described in Chapter 5) that may, for example, use a decision tree classifier for one view and a support-vector machine for the other, and whose final output is the result of iterative pseudo-labeling and re-training. Furthermore an approach such as cross-validation allows performance results to be estimated from actual observed performance of implemented algorithms as opposed to analytic approximations. Therefore my thesis introduces an alternative, a sampling approach similar in motivation to cross-validation in order to estimate model performance. The proposed approach involves generating new training and test data by sampling from the large amount of unlabeled data and estimated conditional probabilities for the labels, and like cross-validation evaluates performance by re-training models and computing average predicted test errors.

Each component of the thesis is evaluated on several synthetic and real world data sets and the experimental results demonstrate the efficacy of the proposed methods.

1.5 Thesis Organization

The chapters of this thesis together form a cohesive body of work/study on learning with low-quality data and in particular learning with limited labeled data, and multi-view semi-supervised learning with missing views. However the chapters are intended to be independent. While the chapters are related, they were written, and the associated work was carried out, so that each chapter could stand by itself.

The outline of the remainder of this thesis is as follows. First, preliminary study on learning with low-quality data is given in the following three chapters. The first part, Chapter 2, is on work on incorporating the structured relationship between features in learning for limited labeled data problems [147], the second part, Chapter 3, is on adapting a large margin learning algorithm for transductive transfer learning [148], and the final part of the preliminary study, Chapter 4, is on feature extraction for knowledge transfer [150].

Afterwards, a general overview is given of the related work in multi-view semi-supervised learning in Chapter 5. Then Chapters 6, 7, and 8 provide additional background information, details on the proposed methods, and detailed experimental study for the proposed methods of view completion via feature generation, active view completion, and model selection, respectively. Finally, conclusions and key areas of future work drawn from the results of this thesis work are discussed in the final chapter, Chapter 9.

Chapter 2

Preliminary Study I: Laplacian Regularization for Structured Input

2.1 Introduction

Consider a p-dimensional multivariate random variable $X = (x_1, x_2, \ldots, x_p) \in \mathbb{R}^p$ where there are some known relationships for the features in X. We investigate the problem of performing effective supervised learning to build accurate classification models for mapping such random variables to class labels, based on observed samples and the relation of the features.

Data with intrinsic feature relationships are becoming abundant in many application domains such as bioinformatics, sensor networks, and social networks among others. For instance, in pathway-based microarray classification, a biological network contains a set of genes, taking values based on their expression levels, and there is a known binary relation of genes: the pathway topology [119, 144]. In this case the goal of the data analysis is to use the expression data to predict a measurable outcome, such as the presence or absence of a disease. In sensor networks, there has been a burgeoning interest in incorporating sensors in everyday life to monitor the environment, supply information, and ensure security. At a given time point regarding the state of the full sensor network, the features are the readings of the sensors, and we usually know the topology or the physical location of the sensors in relation to each other. The goal of the analysis is to detect events of interest based on the collective values of the sensors in the network.

Exploring the relationship between features is not new. Recently in structured feature selection, supervised learning algorithms have been explored for data sets where features have some natural “structure” relationships [198, 211, 215, 219, 223]. For example, Yuan and Lin explored the situation where features may be naturally partitioned into groups and studied the regression problem of grouped features using a technique called grouped Lasso [211]. Another possible type of structure relationship of features is a hierarchical relation (i.e., a directed acyclic graph defined on features) and that has been explored in [198, 219]. In [215], both group structure and hierarchical relation have been studied in a unified framework. Recently Kim and Xing assumed that all the features fit into a linear chain (e.g., genes in a chromosome) and have studied regression problems for such data sets [109]. All these studies, however, do not consider the general case where a general undirected graph is defined to capture the structure relationship of features for classification and regression.

Here we extend previous work on structured feature selection and investigate the new classification problem where features of a data set have a natural graph relationship. We assume such relationships are known and fixed among all instances of the data set. We call such a problem an *aligned graph classification problem* where we may use a graph to model a datum, vertices represent features, edges represent binary relation between features, and vertex and edge set remains the same across a set of samples. Specifically we formalize our classification problem below.

Problem Statement: the Aligned Graph Classification Problem. Given a random variable $X = (x_1, x_2, \ldots, x_p) \in \mathbb{R}^p$, a graph G is a *feature relationship graph* of X if the vertex set of G is the p features. Given a set of n observations $\{(X_i, y_i)\}$, $X_i \in \mathscr{X} \subset \mathbb{R}^p$, $y_i \in \mathscr{Y} = \{1, 2, \ldots, K\}$, $K \in \mathbb{N}$, $i \in [1, n]$, and a feature relationship graph, the *aligned graph classification problem* is to build a classification model $f : \mathscr{X} \to \mathscr{Y}$ to assign class labels to unseen random variables in $\mathscr{X}$ to minimize expected loss. To simplify discussion, from here on, we restrict $\mathscr{Y} = \{1, 2\}$ to the binary class case, 0-1 loss function (i.e., 1 if $y = f(x)$ and 0 otherwise), and undirected feature relationship graphs. Furthermore, we restrict the feature relationship graph structure to be fixed across the set of observations. In other words, the relationship between features is fixed and thus the edges defined between features are fixed for the aligned graphs, each graph will have the same set of edges but possibly different, but aligned, vertex labels, given by the value the random variable takes for that observation.

One way to perform aligned graph classification is to simply use traditional supervised classification algorithms that do not consider the fixed graph structured represented by the feature relationships. By incorporating the graph structure information along with the vertex labels (feature values) in the classification model construction the aim is to improve predictive performance over methods that only consider the feature values for a given observation. Another approach for aligned graph classification that might be considered is to use graph kernel functions for classification [86]. Graph kernels map a set of data to a high dimensional Hilbert space without explicitly computing the coordinates of the data. Coupled with kernel machines such as support vector machines, graph kernel methods can be used for tasks include classification [189], regression [51] and feature extraction through principle component analysis [166]. The adoption of existing graph kernels for aligned graphs, however, is not straightforward for two major reasons: (i) most current graph kernels assume discrete node labels and aligned graphs have numeric node labels and (ii) most current graph kernels measure the difference of graph structures while the graph structures do not change in the aligned graph data.

Here instead of exploring graph kernel methods, we adopt the framework of logistic regression and extend the work from numeric data to data with an intrinsic graph structure using regularization. Logistic regression is a popular statistical method for classification that works by modeling conditional probability distributions using a log-linear model and identifying parameters that maximize the log likelihood of the data, and has been successfully applied to many problems [84, 120]. Comparing to other classification algorithms, logistic regression has the benefits of probabilistic outputs - the probability of a label is returned as opposed to only a discrete class label - and a straight-forward generalization from the binary classification case to the multi-class case. In addition, logistic regression tolerates missing values in data [121]. Many improvements have been proposed and the two most significant ones are (i) adding regularization to the objective function and (ii) applying logistic regression in a kernel space. Incorporating a regularization term that penalizes the square of the L_2 norm of the parameters has been seen to improve the predictive performance of the method particularly for high-dimensional and highly-correlated data [34], following the same idea as ridge regression [91] in which, by penalizing the L_2 norm of the parameters, reduced generalization error can be achieved by shrinking the prediction variance at the cost of increasing bias.

Here, we extend the L_2 regularized logistic regression with a straight-forward modification of the objective function that allows the model learning to be regularized with respect to the graph structure. The basic idea is to force the parameters to vary smoothly over the graph, the idea being quite similar to recent work in semi-supervised learning. The structure of a similarity graph is incorporated in the learning framework in the form of the Laplacian of the graph; the Laplacian of the graph is used in unsupervised (e.g., [174]) and transductive and semi-supervised learning (e.g., [3, 227] when such a similarity structure exists between the data samples. We pursue a similar idea; to improve prediction we incorporate additional information in the form of the graph structure relating the variables and enforce a smooth parameter variation over the graph structure for the variables by means of regularization. The idea should be of particular interest when less labeled information is available, i.e., for small sample data sets or data sets where the ratio of the number of samples to the dimensionality of the data is small.

In summary, our contributions are

- We formalized the aligned graph classification problem for data set where features have a natural structure relationship.
- We extended the logistic regression to include the normalized graph Laplacian, incorporating the Laplacian in the regularization term. We showed that this results in a simple modification to the original logistic regression solution and update using the efficient newton-raphson

approach for finding the zeros of the gradient.

- We developed an approach to incorporate the graph Laplacian regularization in *kernel logistic regression*, which uses a basis expansion to allow non-linear functions of the variables, similar to support vector machines.

- We performed a comprehensive experimental evaluation, showed that Laplacian regularized logistic regression is an effective method for incorporating the graph structure in the prediction problem, evaluated these methods on synthetic and real world data sets and compared the performance of the methods to competing methods including support vector machines and unregularized logistic regression.

The rest of this chapter is organized in the following way. Section 2.2 discusses related work. Section 2.3 presents background information and detailed discussion of our algorithms. Section 2.4 presents the experimental study of our algorithms as compared to competing methods. Finally we give a short conclusion and a discussion of the future work.

2.2 Related Work

We use logistic regression as our framework for building classification models for aligned graph classification; logistic regression has also been used extensively for scientific data analysis. For example, sparse logistic regression was proposed to perform gene selection in [173], a partial least squares with penalized logistic regression algorithm was proposed for high-dimensional, small-sample problems in [67], and in [120] logistic regression is used for feature selection. The approach of [173] has been recently improved in [33] using Bayesian regularization, and applied to the problem of cancer classification, and an L_2 penalized logistic regression method for classification was proposed in [223].

In bioinformatics research there has recently been much interest in using computational methods to associate groups of genes such as groups defined by biological pathways (graphs) with a clinical outcome such as a disease. For example, a statistical method for determining if a group of genes is significantly related to a clinical outcome by calculating a p-value for the group was proposed in [72]. Another statistical test, the Multi-dimensional Cluster Misclassification test (MCM-test), was proposed in [119] for associating pathways with disease outcomes by modeling expression values for a group of genes as fuzzy sets for each outcome and using the membership of the genes in the fuzzy sets to determine significance. For the similar problem of selecting significant pathways and performing classification, a random forest approach was proposed in [143]. For the problem of detecting gene-gene interaction, an L_2 regularized logistic regression method was proposed in [144].

Our work is different from existing work in that we use a general graph to capture relationship between features. In our method we consider a graph as a manifold and we factor in the graph topology using graph Laplacian as a regularization factor. Hence the key insight is that the conditional probability distribution, as evaluated in the logistic regression, varies smoothly along the manifold representing a graph.

2.3 Methodology

2.3.1 Background and Notations.

A *graph* G is described by a finite set of nodes V and a finite set of edges $E \subset V \times V$. In most applications, a graph is labeled, where labels are drawn from a label set λ. A labeling function $\lambda : V \cup E \to \Sigma$ assigns labels to nodes and edges. In *node-labeled graphs*, labels are assigned to nodes only and in *fully-labeled graphs*, labels are assigned to nodes and edges. Here we consider node labeled graphs only since nodes represent features for a sample.

Following convention, we denote a graph as a quadruple $G = (V, E, \Sigma, \lambda)$ where V, E, Σ, λ are explained before. We represent a graph with n nodes using its adjacency matrix $\xi = (\xi_{i,j})_{i,j=1}^{n}$ where $\xi_{i,j} = 1$ if there exists an edge incident on nodes i and j in G, and zero otherwise. We use capital letters, such as G, for a single graph, $V[G]$ for the node set of G and $E[G]$ for the edge set of G, and upper case calligraphic letters, such as $\mathscr{G} = G_1, G_2, \ldots, G_n$, for a set of n graphs.

Two graphs G, G' are *aligned* if there exists a 1-1 mapping $\varphi : V[G] \to V[G']$ such that $(u,v) \in E[G]$ if and only if $(\varphi(u), \varphi(v)) \in E[G']$. Clearly the aligned relation is (i) reflective, (ii) symmetric, and (iii) transitive and hence an equivalence relation. A group of graphs is *aligned* if the graphs in the group are pair-wise aligned.

Example 2.3.1. *In Figure 2.1 we show three graphs defined on 4 features* $\{x_1, x_2, x_3, x_4\}$ *with a star topology. Clearly the three graphs are aligned since they have the same topology. We view each graph as an instance of a 4-dimensional variable* $X_i = (x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4}) \in \mathbb{R}^4$*,* $i \in [1,3]$ *with a binary relation defined on the 4 features.*

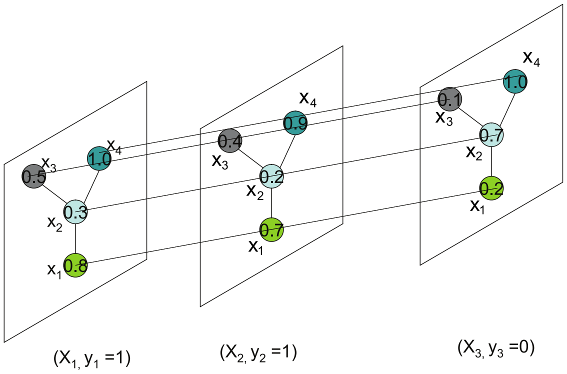


Figure 2.1: Three aligned graphs

2.3.2 Logistic Regression.

Before we introduce regularized logistic regression, we briefly overview basic logistic regression [84]. Logistic regression fits a sigmoid function, $P(Y = 1|\vec{X} = \vec{x}; \vec{\beta}) = \frac{1}{1+e^{-\vec{\beta}^T \vec{x}}} = \frac{e^{\vec{\beta}^T \vec{x}}}{e^{\vec{\beta}^T \vec{x}}+1}$, representing the probability the class label takes value 1 given the data sample has values $\vec{x}$ and the parameters are $\vec{\beta}$, to the training data, here we use $\vec{x}$ to denote a data vector with an additional feature value of 1 concatenated to the beginning for convenience (to incorporate the intercept). Using the training data we find the parameters $\vec{\beta}$ that best fit the data, and can then use the sigmoid function to map any future data vector to a value in $[0,1]$. The fitting is achieved by maximizing the log-likelihood of the data (which we will denote as $\ell(\vec{\beta})$, as it is a function of the parameters $\vec{\beta}$), $\sum_{i=1}^{N}\{y_i \log(P(Y=1|\vec{X}=\vec{x}_i;\vec{\beta})) + (1-y_i)\log(1-P(Y=1|\vec{X}=\vec{x}_i;\vec{\beta}))\}$, which can be expressed as:

$$\ell(\vec{\beta}) = \sum_{i=1}^{N}\{y_i\vec{\beta}^T\vec{x}_i - \log(1+e^{\vec{\beta}^T\vec{x}_i})\} \tag{2.1}$$

, by setting the gradient, $\frac{\partial\ell(\vec{\beta})}{\partial\vec{\beta}} = \sum_{i=1}^{n}\{\vec{x}_i(y_i - P(Y=1|\vec{X}=\vec{x}_i;\vec{\beta}))\}$, equal to $\vec{0}$. We then find the zeros using an iterative process, the Newton-Raphson algorithm, which requires taking the second derivative of the log-likelihood. We express the derivative and second derivative of the log-likelihood in matrix form so that the update becomes:

$$\vec{\beta}^{new} = \vec{\beta}^{old} - (\frac{\partial^2\ell(\vec{\beta}^{old})}{\partial\vec{\beta}\partial\vec{\beta}^T})^{-1}\frac{\partial\ell(\vec{\beta}^{old})}{\partial\vec{\beta}} \tag{2.2}$$

which is:

$$\vec{\beta}^{new} = \vec{\beta}^{old} - (X^TWX)^{-1}X^T(\vec{y}-\vec{p}) \tag{2.3}$$

where $\vec{p}$ is a column vector with $p_i = P(Y=1|\vec{X}=\vec{x}_i;\vec{\beta}^{old})$, and $W = diag(p) * diag(\vec{1}-p)$, where $diag(p)$ signifies a diagonal matrix with diagonal entries $W_{ii} = p_i$ and all other entries set to 0, and $\vec{1}$ is a column vector of ones, with dimension N. With the new beta calculated with equation 2.3, the probabilities are recalculated (p and W updated), and the process repeats until convergence, measured by the entries of W becoming close to 0 or by the change in $\vec{\beta}$ becoming close to 0, using some small threshold value.

Thus for each data vector, we learn a set of parameters $\vec{\beta}$, and can then map each data vector to a probability of class label. We can threshold the output from the logistic regression at 0.5 to obtain the predicted class.

2.3.3 Laplacian-Norm Regularized Logistic Regression.

Here we incorporate graph Laplacian as a regularization term in the logistic regression. Before we talk about regularized logistic regression, we define graph Laplacian and normalized graph

Laplacian.

For an undirected graph G with the adjacency matrix ξ, the *Laplacian* L of G is:

$$L = D - \xi; \tag{2.4}$$

Where D is the density matrix of ξ, defined as $D = (d_{i,j})_{i,j=1}^{n}$ where

$$d_{i,j} = \begin{cases} \sum_{k=1}^{n} \xi_{i,k} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

The *normalized Laplacian* is $\mathscr{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$.

Incorporating the normalized graph Laplacian norm as a regularization term in the logistic regression actually results in a simple modification to the original logistic regression solution. Furthermore, substituting the identity matrix for the normalized Laplacian $\mathscr{L}$ results in logistic regression with the ridge penalty (the square of the L_2 norm of β), since $\vec{\beta}^T I \vec{\beta} = \vec{\beta}^T \vec{\beta}$.

The new objective function becomes:

$$g(\vec{\beta}) = \sum_{i=1}^{N} \{ y_i \vec{\beta}^T \vec{x}_i - \log(1 + e^{\vec{\beta}^T \vec{x}_i}) \} - \frac{1}{2} \lambda \vec{\beta}^T \mathscr{L} \vec{\beta} \tag{2.5}$$

The new gradient is given by:

$$\frac{\partial g(\vec{\beta})}{\partial \vec{\beta}} = X^T (\vec{y} - \vec{p}) - \lambda \mathscr{L} \vec{\beta} \tag{2.6}$$

The new hessian is given by:

$$\frac{\partial^2 g(\vec{\beta})}{\partial \vec{\beta} \partial \vec{\beta}^T} = -X^T W X - \lambda \mathscr{L} \tag{2.7}$$

And the new newton-raphson update is given by:

$$\vec{\beta}^{new} = \vec{\beta}^{old} - (X^T W X + \lambda \mathscr{L})^{-1}(X^T(\vec{y} - \vec{p}) - \lambda \mathscr{L} \vec{\beta}^{old}) \tag{2.8}$$

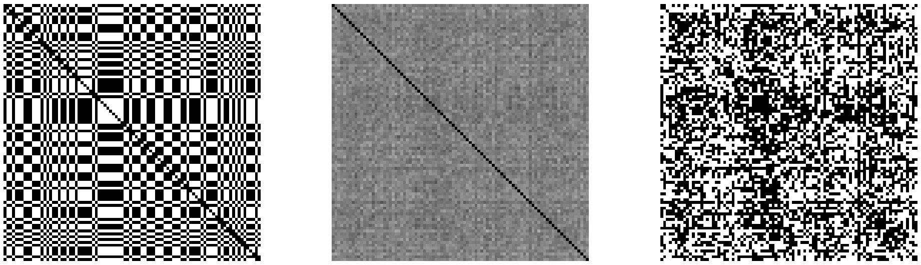
2.3.4 Graph Regularized Kernel Logistic Regression.

Kernel logistic regression works by introducing a basis expansion so that $f(\vec{x})$ in $P(Y = 1|\vec{X} = \vec{x}; \vec{\beta}) = \frac{1}{1+e^{-f(\vec{x})}}$, previously equal to $\vec{\beta}^T \vec{x}$ is now equal to $\alpha_0 + \sum_{i=1}^{N} \alpha_i K(\vec{x}, \vec{x}_i)$ where $K(.,.)$ is a kernel function implicitly defining a Hilbert space and a feature mapping. In order to keep our Laplacian-regularization framework intact, we define a second method. Since the parameters are translated to the feature space, i.e., from $\vec{\beta}$ varying over the p features (vertices) in the input feature space to $\vec{\alpha}$ varying over the n features in the kernel space, the original constraints on the graph structure are lost for the parameters α. Thus, in order to include the Laplacian regularization in the kernel space it is necessary to translate the graph structure from the input feature space to the kernel feature space. Essentially we want to define a new weighted graph structure between the n-samples such that the similarity function between two samples is regularized by the original graph structure (the original graph Laplacian in our framework). This is a similar idea to semi-supervised learning where we define an underlying similarity graph from the data. Here we want the graph created to impose similarity based on the closeness for matching vertices and the smoothness over the vertices.

In order to derive a similarity graph to regularize the alpha parameters, we estimate a sample similarity function that itself is regularized by the Laplacian of the original graph. We start with an edge of weight 1 between each training sample with the same label, of weight 0 (no edge) otherwise, a rough graph with connections between all samples of the same class. To incorporate the original graph structure, we train a logistic regression model to predict probabilities of link connections that is regularized by the original graph Laplacian. To do this we use a similarity measure (in the form of a Gaussian kernel function) between each pair of aligned vertices in the original graph, and fit a set of logistic regression parameters, using the Laplacian regularization. This translates the binary edge existence function to a weight that is regularized by the original graph structure, in effect smoothing the similarity function over the original graph structure.

To select the vertex-wise similarity parameter (width of the Gaussian) and the regularization parameter, λ, one option is to perform a cross-validation grid search with the training data, enforcing only that the thresholded output correctly predicts the link. In this way, the values can still vary smoothly. However, the number of samples in this case becomes $(n^2 - n)/2$ (for n training samples), since each pair of training samples becomes a new training sample for the edge prediction function, so performing the multiple iterations with this higher sample size set can be time consuming. As an alternative, we only perform the logistic regression once by setting σ equal to the standard deviation for each feature and using a high λ value to strongly enforce the regularization term (two times the number of new training samples), avoiding the lengthy grid search process.

In this way we can achieve our goal of creating a new graph structure in the kernel feature space that is still regularized by the original graph structure in the input feature space. Figure 2.2 shows a comparison of the rough, original similarity matrix to the derived similarity matrix for 90 training samples from a synthetic data set. The original structure can still be seen in the regressed similarity matrix (e.g., the cross shape) but this structure is softened (regularized).



(a) Similarity matrix determined by class membership

(b) Similarity matrix derived from regularized regression

(c) Thresholded regression similarity matrix (at 0.5)

Figure 2.2: Regularized similarity graph for 90 samples of synthetic data

2.3.5 Regularized Local Logistic Regression.

Since the regularized kernel logistic regression method described in the previous section is time-consuming to perform in full, we explore another kernel logistic regression method for learning nonlinear class boundaries as an alternative, local logistic regression. The motivation is that often we may desire a model that does not find a global fit to the data, but rather a local fit, similar to the nearest neighbor method and local linear regression method. In this case local logistic regression can be used. Local logistic regression results from a simple modification to the original logistic regression formulation; each sample is weighted by how close it is to the input test sample using some smoothed distance function such as the Gaussian kernel, when the model is fitted. This is described by the following weighting of the likelihood (L) equation: $L = \prod_{i=1}^{N} P(Y = y_i|\vec{X} = \vec{x}_i; \vec{\beta})^{\gamma_i}$, with $\gamma_i = e^{-\frac{||\vec{x}_i - \vec{x}_t||^2}{2\sigma^2}}$ for test input $\vec{x}_t$, which translates into multiplying each term in the log-likelihood by its sample weight. The Laplacian regularized version is the same as for regular logistic regression, except for weighting samples in the likelihood term of the objective function. The new update equations result by modifying equations 2.3 and 2.8 so that $W_{ii} = p_i\gamma_i$ and $\vec{y} - \vec{p}$ is scaled by the weights ($diag(\vec{\gamma})(\vec{y} - \vec{p})$). Here increasing the kernel width σ results in moving closer to the global solution.

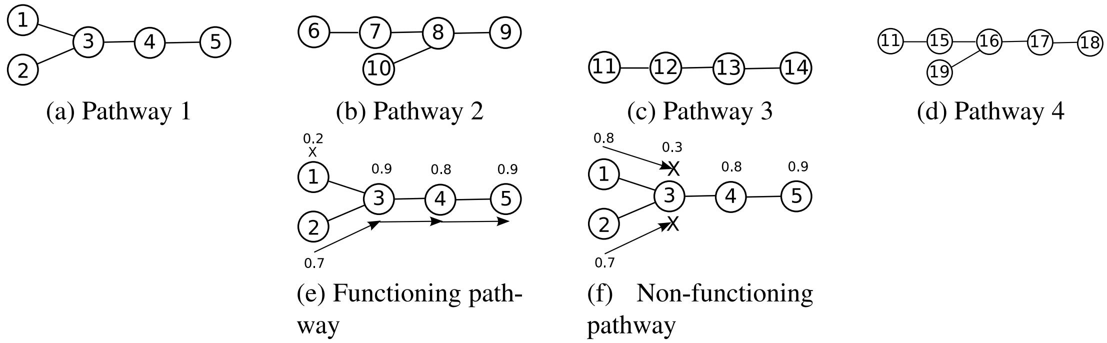
In the subsequent discussion for simplicity, we refer to the logistic regression method as “LR”, the Laplacian-regularized logistic regression method as “LREG”, the L_2 norm regularized logistic regression method (with $\mathscr{L}$ equal to the identity matrix) as “L2”, the kernel logistic regression as “KLR”. Similarly, we refer to the unregularized local logistic regression method as “LOC_LR”, the the L_2 norm regularized local logistic regression method as “LOC_L2” and the Laplacian-regularized local logistic regression method as “LOC_LREG”.

2.4 Experimental Evaluation

2.4.1 Data

2.4.1.1 Synthetic Data.

We generated synthetic test data for an undirected graph with 19 vertices described by the 4 arbitrary created pathways shown in figures 2.3a - 2.3d, which specify the binary relationships between the given variables. For our tests we assume all we know is the existence of a relationship between the variables and form the corresponding undirected graph and 19x19 adjacency matrix. To generate data, the graph class is labeled 1 if at least 2 pathways “produce” (take value) 1, otherwise it is 0. A pathway “produces” 1 if all the node values along any path from a start node (at the left) to an end node of the path are greater than 0.5, otherwise it produces 0. Examples are given in figures 2.3e and 2.3f. We indicate a path with all values greater than 0.5 in Figure 2.3e by small arrows. In Figure 2.3f we show a broken path since node (3) has value 0.3 which is less than 0.5. Thus the pathway in Figure 2.3e “produces” a label 1 and the pathway in Figure 2.3f “produces” a label 0. To generate data we randomly generate values for all the nodes in the range $[0, 1]$ and test the graph outcome. We generate 100 samples, and continue replacing samples with label 0 until half have label 1.



(a) Pathway 1 (b) Pathway 2 (c) Pathway 3 (d) Pathway 4

(e) Functioning pathway (f) Non-functioning pathway

Figure 2.3: Artificial pathways used to generate test data

2.4.1.2 Real World Data.

Next, we consider microarray gene expression data classification: given a set of samples of gene expression values and the associated class labels (e.g., disease or no disease), learn a classification model to predict the label of a test sample using its gene expression values as features. We can view the microarray classification task as an aligned graph classification task by considering the biological pathway structures associated with the genes. Here each pathway related to the outcome of interest is represented by an undirected graph with vertices as genes and edges representing the existence of relations between the genes such as protein-protein interactions resulting in activation or phosphorylation. To obtain the aligned graph structures for our experiments, we extract pathway graphs from a standard source of biological pathway information, the internet-accessible KEGG pathway database [107].

Since incorporating pathway structure in the learning process for pathways that are not related to the outcome of interest would not be expected to improve performance, and to avoid testing every pathway, we first perform external pathway selection. Determining which pathways are related to a particular outcome could be performed separately by any number of methods, e.g., searching through scientific literature for known related pathways, or using a computational statistical test tool; we use a readily-available method provided as a pre-built statistical package, the *global test* [72] method which tests if a group of variables are significantly related to an outcome of interest (the idea of incorporating grouped variable selection into our Laplacian regularized framework is an area of future work). We use global test with the pathway gene expression data paired with the outcome labels to obtain a top candidate list of pathways from the KEGG database; the pathway structures of the selected pathways form the aligned graphs used for evaluating our algorithms.

We used the following three data sets for our experimental study:

- **Diabetes Data:** The first microarray data set we include is a microarray data set related to diabetes, obtained from [128] (available online at `http://www.broad.mit.edu/mpg/oxphos/`). The data set contains the gene expression values of $22,280$ genes for 44 different subjects, 17 with type 2 diabetes (DM2), 17 with normal glucose tolerance (NGT) and 10 with impaired glucose tolerance (IGT). As in [119], we use only the samples of subjects with type 2 diabetes and those with normal glucose tolerance, resulting in a total of 34 samples. We use the global test method to estimate related pathways; we select all pathways found to be related to the diabetes outcome by the global test method with a significance p-value of less than 0.1 and keep those that have an associated graph structure, resulting in the 14 pathways shown in table 2.1. In evaluating the aligned graph classification methods, their performance on the Insulin Signaling Pathway is of particular interest, since aside from the global test results, we would expect this pathway to be related to the diabetes disease, and as such can be more confident that the pathway is related to the outcome in this case.

- **Breast Cancer Data:** The next data set we use is a microarray gene expression data set for human breast cancer samples [45]; in this case there are 118 breast tumor samples and we select the "alive at endpoint" factor as the class label, resulting in 41 positive samples and 77 negative samples. We once again use global test to select related pathways, however since only 3 pathways were found with *p*-value less than 0.13, we select the pathways with graphs from the top 20, resulting in 14 pathways.

- **Yeast Data:** The final data set is a microarray data set for yeast [127, 154]; here the gene expression values are measured across 18 independent samples of (*Saccharomyces cerevisiae*) yeast cultures, and the goal is to classify whether or not a sample was grown with irradiation (6 samples are labeled as Irradiated, I, and 12 as Not Irradiated, NI). Since the data set was much smaller (around $6,000$ genes), we obtained results for all pathways we were able to make graphs for, a total of 94 pathways. In addition we applied pre-processing to handle missing values by replacing feature values with the average value for that feature if at least 80% were not missing, otherwise we removed the feature.

2.4.2 Evaluation Criteria.

We use several approaches to evaluate the performance of the graph classification methods. For the synthetic data we perform 100 trial iterations using a hold-out approach, generating a new sample set from the given graph and using a fixed fraction of the 100 samples for the training data and the remainder for testing, taking the average and standard deviation of the performance criteria across the trials. For the diabetes data set, we average the performance across 30 iterations of ten-fold cross-validation [110], and for the breast cancer data set, 30 iterations of five-fold cross-validation, since there are more samples. For the yeast data, we estimated performance using two approaches, due to the small data set size and imbalance of labels. For the first approach, we generate 50

Table 2.1: Estimated related pathways found with global test (p-value < 0.1) for the Diabetes data set

Index	Pathway	Genes	P-value
1	Insulin signaling pathway	250	0.0673
2	mTOR signaling pathway	90	0.0229
3	Biosynthesis of steroids	42	0.0577
4	Oxidative phosphorylation	153	0.0384
5	Alanine and aspartate metabolism	44	0.0264
6	Phenylalanine, tyrosine and tryptophan biosynthesis	14	0.0497
7	Glycosphingolipid bio-synthesis - lactoseries	15	0.0931
8	Glycosphingolipid bio-synthesis - globoseries	23	0.0839
9	Lipoic acid metabolism	2	0.0379
10	Terpenoid biosynthesis	12	0.0337
11	Nitrogen metabolism	39	0.0969
12	Alkaloid biosynthesis I	7	0.0500
13	PPAR signaling pathway	118	0.0755
14	SNARE interactions in vesicular transport	65	0.0263

training and test sets by generating all 50 unique partitions of the positive class such that at least 2 samples from the positive class (I) are in each set, and randomly partition the data from negative class (NI) so that the training set always has 10 samples. The other approach we used was bootstrap sampling, the ".632+" bootstrap estimator (see [84] for more details), using 100 bootstrap data sets.

For all the experiments, we estimate the accuracy and performance for our new Laplacian regularized logistic regression method (LREG) and compare it to five other methods, which only use the feature values of the graphs: previous logistic regression methods, including unregularized logistic regression (LR), L_2 norm regularized logistic regression (L2), and kernel logistic regression (KLR), and support vector machine methods which include a linear kernel support vector machine (SVM_LIN) and a Gaussian radial-basis function (RBF) kernel support vector machine (SVM_RBF) (see, e.g., [84] for more information about these common classifiers). In addition, for our synthetic experiments and for the key diabetes pathway, we include results for the Laplacian regularized local logistic regression (LOC_LREG) along with the unregularized local logistic regression (LOC_LR) and an L_2 norm regularized local logistic regression (LOC_L2). We implemented the logistic regression methods in Matlab and used a Matlab toolbox implementation for the support-vector methods. To select parameters for all aligned graph classification models where needed (specifically λ for the various regularized logistic regression methods, σ for the kernel logistic regression methods and RBF SVM method, and C for the SVM methods), we perform a cross-validation grid search with the training data using a course-to-fine grid approach as in LibSVM [35].

In addition to accuracy, we include three other common performance criteria as described in the following list:

1. Accuracy (ACC): $\frac{TP+TN}{TP+TN+FP+FN}$
2. Matthews Correlation Coefficient (MCC): $\frac{TP*TN-FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+RFN)}}$
3. Sensitivity(SEN): $\frac{TP}{TP+FN}$
4. Specificity (SPE): $\frac{TN}{TN+FP}$

In this description, FP denotes “false positive,” a negative instance that was classified as positive, TP denotes “true positive,” a positive instance that was classified as positive, TN denotes “true negative,” a negative instance that was classified as negative, and FN denotes “false negative,” a positive instance that was classified as negative.

Additionally, since the average accuracy of one method may be better than another, but the standard deviation could be too high to distinguish if the method performed better consistently across test iterations, we perform a paired t-test at the five percent level between the 100 test accuracies for each method, to determine if a method’s higher accuracy can be considered statistically significant. For the real-world data sets with the cross-validation, the t-test is across the number of iterations, 30.

2.4.3 Synthetic Data Classification Results.

The first set of results shows the performance criteria averaged over 100 iterations of a 60% holdout, so that for each iteration, 100 samples were generated from which 40 samples were randomly selected for training, 60 for testing (the samples were selected so that at least one-third of each class was present). These results are shown in table 2.2, with the best method for each criteria shown in bold (results for the local logistic regression methods are not included in this table to save space, but are shown in figure 2.4).

Table 2.2: Results on synthetic test data for aligned graph classification methods

	LREG	**L2**	**SVM_ LIN**	**LR**	**SVM_ RBF**	**KLR**
ACC	**0.767**	0.732	0.722	0.666	0.726	0.716
std	0.060	0.062	0.060	0.066	0.067	0.061
MCC	**0.541**	0.469	0.448	0.338	0.460	0.470
std	0.119	0.125	0.121	0.133	0.135	0.110
SEN	0.789	0.747	0.739	0.677	0.716	**0.890**
std	0.094	0.100	0.095	0.107	0.117	0.086
SPE	**0.746**	0.716	0.704	0.656	0.737	0.542
std	0.092	0.096	0.099	0.111	0.104	0.147

We performed a paired t-test at the five percent level on the accuracies obtained from the 100 runs, and found that the LREG method is performs significantly better in terms of accuracy (the null hypothesis of same mean of distribution is rejected) than all of the other methods. Similarly, all the regularized methods are found to perform significantly better than the unregularized logistic regression (LR). These results are shown in table 2.3, in which a significance was found using the paired t-test between the method in each row and column, a 1 indicating a significant difference with a positive 1 indicating the method in the row had a higher average accuracy than the method in the column and a negative 1 lower, and a 0 representing that the null hypothesis could not be rejected.

Table 2.3: Paired t-test results on synthetic test data across 100 iterations, between each pair of methods. A positive 1 indicates the method in the row performed significantly better on average than the method in the column, a negative 1, worse, and a 0 that the difference in performance of the two methods was not statistically significant according to the t-test at the 5% level.

	LREG	**L2**	**SVM_ LIN**	**LR**	**SVM_ RBF**	**KLR**
LREG	0	+1	+1	+1	+1	+1
L2	-1	0	+1	+1	0	+1
SVM_ LIN	-1	-1	0	+1	0	0
LR	-1	-1	-1	0	-1	-1
SVM_ RBF	-1	0	0	+1	0	0
KLR	-1	-1	0	+1	0	0

The next set of results, figure 2.4, shows the relationship between accuracy and the size of the training set used, obtained by running the experiments with each hold-out percentage (100 iterations as before). As can be seen the Laplacian regularized method (LREG) outperforms the others consistently, but the performance gain is greatest with smaller training sample size. While the other methods converge to a lower value at the smallest training sample size tested (10 training samples), the Laplacian regularized method maintains a 5 percent higher accuracy. We also included results for the local logistic regression methods, for the first 4 training set sizes. Here we see that the L_2 regularized local logistic regression (LOC_L2) is a significant improvement over the unregularized local logistic regression (LOC_LR), and that the Laplacian regularized local logistic regression (LOC_LREG) significantly outperforms both. For small samples, regular Laplacian regularized logistic regression (LREG) outperforms LOC_LREG, which in turn outperforms the other methods, but with increasing sample size the LOC_LREG method achieves comparable performance. While in general, the results obtained for the local logistic regression method using a nonlinear similarity function were worse than the methods with linear models, the results were not far off. We included these results to show the plausibility of using the Laplacian regularized local logistic regression to incorporate aligned graph structure for those cases where a nonlinear boundary is desired or known to exist.

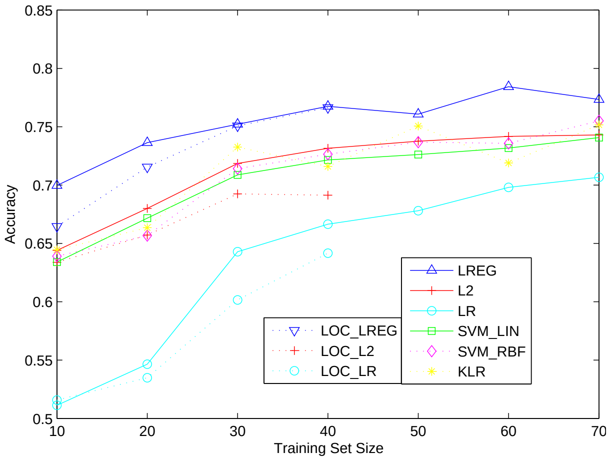


Figure 2.4: Average Accuracy vs. Training Set Size for Synthetic Data

Figure 2.5 shows the variation of the accuracy of the Laplacian regularized logistic regression method (LREG) with respect to the regularization weight, λ, obtained by averaging over 100 iterations as before with a training set size of 40. We also include the results for the L_2 regularized logistic regression (L2) for comparison as well as the constant result for unregularized logistic regression as a baseline. From the results we see that the LREG method's performance varies in a similar way to the L2 method's performance with respect to the regularization parameter for this experiment, and additionally that in this case it is safer to overestimate the value of the regularization parameter than underestimate, since accuracy increases steadily until about $\lambda = 2^4$ at which point it remains close to the highest value reached.

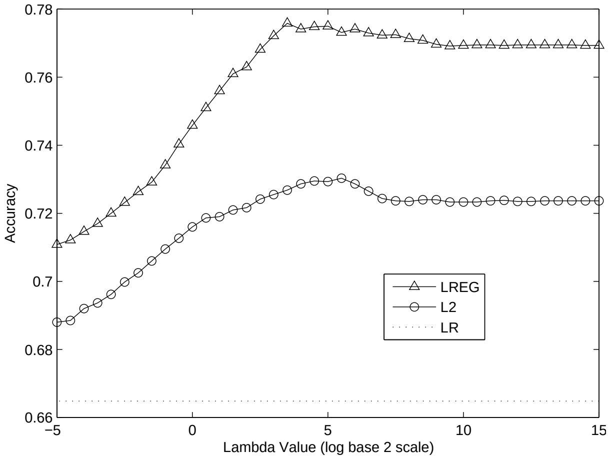


Figure 2.5: Average Accuracy vs. Regularization Parameter for Synthetic Data

2.4.4 Real-World Data Classification Results.

For the real-world data classification results, we show the results for each pathway, i.e., by treating the set of data for each pathway as an aligned graph classification problem. Thus, for example, for data with 14 pathways we in effect have 14 data sets. For the diabetes data, we performed 30 iterations of ten-fold cross-validation to estimate the performance of each method for each pathway. The results of each method for each pathway are shown in figure 2.6, in which each point on the x-axis represents a pathway, and each point on the y-axis the average accuracy.

For the 14 pathways, the Laplacian regularized method (LREG) performed significantly better than the rest for 2 of the pathways, as did the linear SVM (SVM_LIN); the other methods did not perform significantly better than the rest of the methods for any of the pathways, except for the kernel logistic regression method (KLR) for 1 pathway. Furthermore, the only pathways for which the LREG method performed the worst were those for which all the methods had 50 percent accuracy or worse.

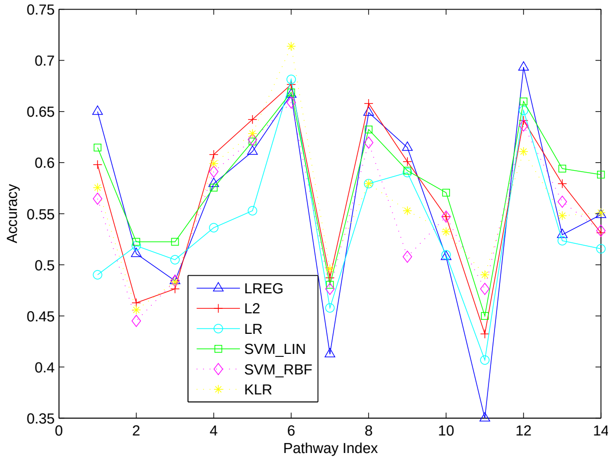


Figure 2.6: Average Accuracy vs. Pathway Index for Diabetes Data

We suspect one reason the Laplacian regularized method did not perform significantly better on all pathways is that many pathways are likely unrelated to the disease outcome, or some of the genes in a given pathway are related, but as a part of a different pathway instead of the given pathway, in which case the Laplacian regularized method would not be expected to improve the performance. Thus we take a closer look at the Insulin Signaling Pathway which we reason is one pathway that is more likely to be related to the diabetes disease outcome. For this pathway we also include results from the local logistic regression methods. The results for the Insulin Signaling Pathway are shown in table 2.4, the best score for each criteria is shown in bold. For this pathway, the Laplacian regularized logistic regression (LREG) performed the best for all criteria. We also see that for this pathway the Laplacian regularized local logistic regression outperformed the other kernel methods, and for each method adding regularization improved the performance. By performing paired t-tests as with the synthetic data, we see that the improvement from the LREG method was statistically significant (table 2.5).

In general in our experiments, the linear logistic regression methods, LR, LREG, and L2

Table 2.4: Results on diabetes data for aligned graph classification methods for the Insulin Signaling Pathway

	LREG	**L2**	**SVM_LIN**	**LR**	**SVM_RBF**	**KLR**
ACC	**0.650**	0.598	0.615	0.490	0.565	0.575
std	0.056	0.052	0.035	0.068	0.062	0.053
MCC	**0.301**	0.197	0.230	-0.019	0.130	0.151
std	0.112	0.104	0.070	0.140	0.124	0.106
SEN	**0.633**	0.588	0.584	0.463	0.565	0.584
std	0.081	0.056	0.046	0.087	0.068	0.064
SPE	**0.667**	0.608	0.645	0.518	0.565	0.567
std	0.052	0.071	0.036	0.113	0.080	0.065

	LOC_LREG	**LOC_L2**	**LOC_LR**
ACC	0.590	0.540	0.509
std	0.046	0.056	0.075
MCC	0.180	0.081	0.017
std	0.093	0.113	0.156
SENS	0.588	0.551	0.483
std	0.073	0.084	0.131
SPEC	0.591	0.529	0.536
std	0.060	0.076	0.121

Table 2.5: Paired t-test results on diabetes test data across 30 iterations, between each pair of methods. A positive 1 indicates the method in the row performed significantly better on average than the method in the column, a negative 1, worse, and a 0 that the difference in performance of the two methods was not statistically significant according to the t-test at the 5% level.

	LREG	**L2**	**SVM_LIN**	**LR**	**SVM_RBF**	**KLR**
LREG	0	+1	+1	+1	+1	+1
L2	-1	0	-1	+1	+1	+1
SVM_LIN	-1	+1	0	+1	+1	+1
LR	-1	-1	-1	0	-1	-1
SVM_RBF	-1	-1	-1	+1	0	0
KLR	-1	-1	-1	+1	0	0

had comparable training time to the support-vector machine methods, and were in many cases faster. However the kernel-based logistic regression methods, KLR and LOC_LR, LOC_L2, and LOC_LREG usually took longer to train, KLR due to calculating the basis expansions and a slower convergence of Newton's method, and the local logistic regression took longer since the regression process had to be repeated for each test point, since the weights γ_i assigned in the optimization were based on the kernel similarity of the tests point to the training points. Thus due to time constraints, we do not include results for these kernel-based methods for all data sets.

Next, we show the results for the breast cancer data in the same graph form as the diabetes data in figure 2.7. In general the less regularized logistic regression such as L_2 regularized logistic regression performs as well as unregularized logistic regression; the Laplacian regularized logistic regression did not outperform all of the other classifiers for any pathway. We suspect that, since the pathways themselves are not known for certain, the relation to the known pathways to the disease may not be strong and hence regularization does not help too much. To test the hypothesis, we checked the global test matches and identified that none of the pathways have *p*-value less than 0.05 and only the first three had *p*-value less than 0.10.

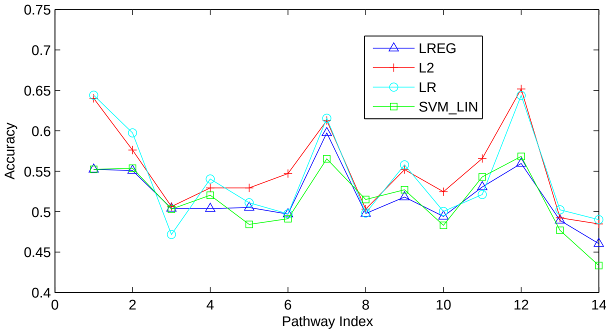


Figure 2.7: Average Accuracy vs. Pathway Index for Breast Cancer Data

Finally we show the results for the 94 pathways of the yeast data for the 50 partition estimate (training set size 10) in figure 2.8 and the ".632+" bootstrap estimate (training sets of size 18) in figure 2.9, with the pathway number on the x-axis and the estimated accuracy on the y-axis. The results are similar to the diabetes results, the best performing method varies for each pathway. The Laplacian regularized logistic regression only obtains significantly improved performance for a few of the pathways. However, we might expect this since it is likely only a few of the pathways are directly related to the outcome of interest. In this case, however, we have no ground truth available for which pathways are truly related, and the methods performed similarly on the top pathways selected by global test, though even this test we would expect to be less accurate with such few samples.

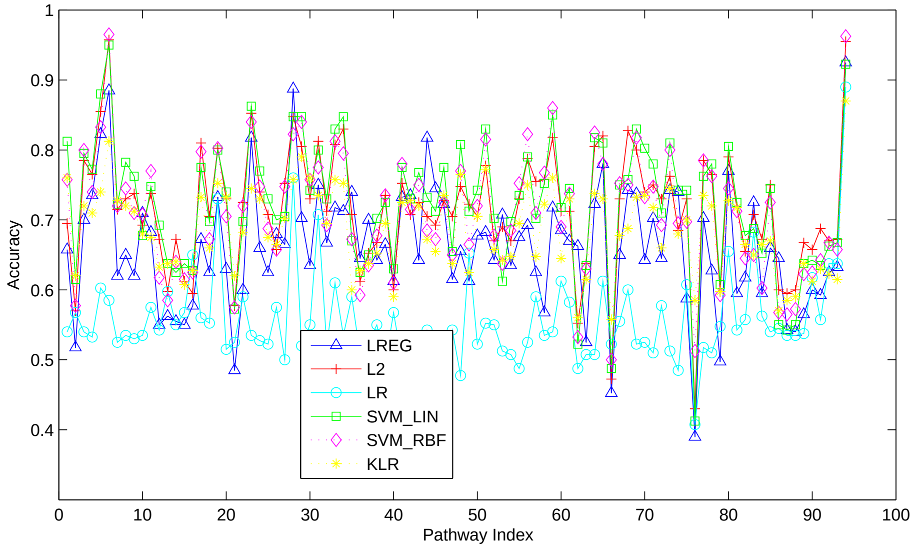


Figure 2.8: Average Accuracy vs. Pathway Index for Yeast Data: Partitioning Estimate

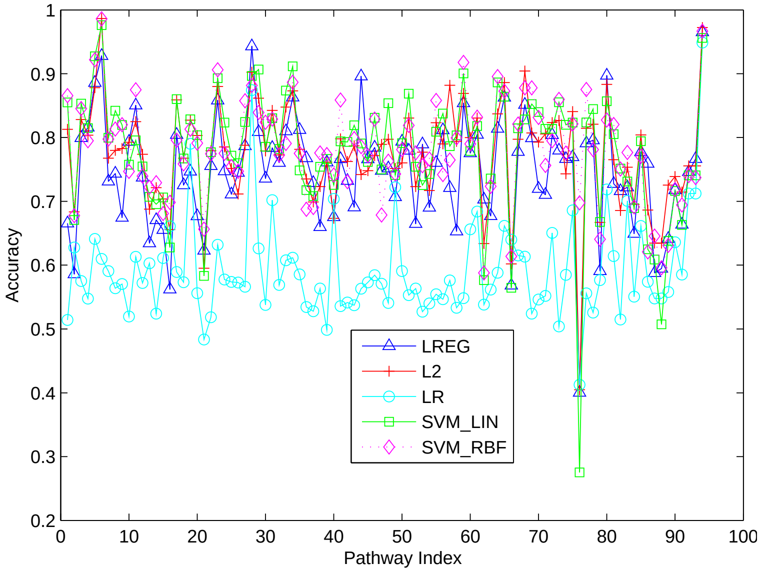


Figure 2.9: Average Accuracy vs. Pathway Index for Yeast Data: Bootstrap Estimate

2.5 Conclusion

Data with intrinsic graph topology are becoming abundant in many applications including bioinformatics and sensor network analysis. We call such data aligned graphs and in this chapter we investigated a new problem of classification on aligned graphs. We have extended the L_2 regularized logistic regression to aligned graph classification. Our experimental study demonstrates the utility of the methods in synthetic and real data sets. In the future, we will investigate dynamic graph structure, where we allow small amount of graph topology changes, in the Laplacian based logistic regression framework.

Chapter 3

Preliminary Study II: Large Margin Transfer Learning

3.1 Introduction

Constructing mining and learning algorithms for data that may not be identically and independently distributed (*i.i.d.*) is one of the emergent research topics in data mining and machine learning [6, 18, 69, 96, 152, 165, 185, 196, 203]. Non-i.i.d. data occur naturally in applications such as cross-language text mining, bioinformatics, distributed sensor networks and sensor-based security [151], social network studies, low quality data mining [228], and ones found in multi-task learning [114]. The key challenge of these applications is that accurately-labeled task-specific data are scarce while task-relevant data are abundant. Learning with non-i.i.d. data in such scenarios helps build accurate models by leveraging relevant data to perform new learning tasks, identifying the true connections among samples and their labels, and expediting the knowledge discovery process by simplifying the expensive data collection process.

Transfer learning aims to learn classification models with training and testing data sampled from possibly different distributions. The common assumption in transfer learning is that the training and testing data sets share a certain level of commonality and identifying such common structures is of key importance. For data that have well-separated structures, exploring the common cluster structure of training and testing sets is a widely used technique [69, 196]. Instance based methods assume a common relationship between the class label and samples and use weighting or sampling strategies to correct differences between training and testing distributions [18, 96, 185]. In feature based methods, shared feature structure is learned in order to transfer knowledge in training data to testing data [152, 165]. In addition, Xue *et al.* used a hierarchical Bayesian model and developed a matrix stick-breaking process to learn shared prior information across a group of related tasks [203]. From a multi-task learning framework, if we assume that the testing data is coming from a new task and that the new task belongs to a parameterized task family, we can learn the structure of such a parameterized task family and use that information for transfer learning, as demonstrated in the zero-data learning algorithm [114].

In this chapter we explore a research direction motivated by manifold regularization which assumes that data distribute on a low dimensional manifold embedded in a high dimensional space [13]. The learning task is to find a low complexity decision function that well separates the data and that varies smoothly on the manifold. Following the same intuition, we approach the non-i.i.d. data learning problem by learning a decision function with low empirical error, regularized by the complexity of the function and the difference between training and testing data distributions, evaluated against the decision function. The idea is to in effect find a manifold for which the training and testing data distributions are brought together so that the labeled training data can be used to learn a model for the testing data. In particular, we aim to obtain a linear classifier, in a reproducing kernel Hilbert space, such that it achieves a trade-off between the large margin class separation and the minimization of training and testing distribution discrepancy, as projected along the linear classifier. Our hypothesis is that unlabeled testing data reveal information about testing data distribution and help build accurate classification models. Though large margin classifiers have been investigated in similar contexts including semi-supervised learning and transductive learning [13, 100, 190], applying large margin classifiers to transfer learning by incorporating a regularization component measuring the distances between training and testing data is new and

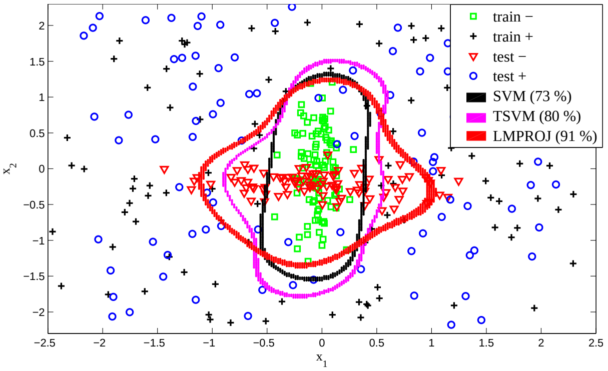


Figure 3.1: Decision boundaries for the standard support vector classifier (black) and our method (red) on a simple generated 2-D transfer learning problem. This example is discussed in detail in Section 3.5.

worth a careful investigation.

We illustrate our hypothesis in Figure 3.1 where we show an artificial data set in a 2D space where training and testing data sets have different distributions. As shown in the figure, the support vector machine builds a decision boundary that fits the training data well. Clearly the decision boundary is not the optimal one as evaluated on the testing data set. Clustering based methods are widely used in designing transfer learning algorithms. In this example, there is no obvious clustering structure for the positive and negative samples and clustering based techniques will not be very helpful. Yet another class of widely used methods is ones that are based on feature extraction and feature selection. These methods will not be very useful since in this case we only have two features and both of them are important. The key observation, as illustrated in this example, is that we need to integrate feature weighting (in order to handle distribution mismatches between training and testing samples) and model selection in a unified framework.

The major advantage of adopting the regularized empirical error minimization paradigm such as the SVM is the potential to exploit many algorithms designed specifically for SVMs with only slight modifications, if any. For example, there have been fast algorithms designed for handling large data sets [94, 101], anomaly detection with one-class SVM, and multi-class SVM for multi-category classification. Other advantages are the rigorous mathematical foundation such as the Representer Theorem, global optimization with polynomial running time using convex optimization, and geometric interpretations through generalized singular value decomposition. We discuss these properties of SVM based transfer learning in detail in the Algorithmic study section.

3.1.1 Notations and Problem Statement

In supervised learning, we aim to derive ("learn") a mapping for a sample $\vec{x} \in \mathscr{X}$ to an output $y \in \mathscr{Y}$. Towards that end we collect a set of n training samples $\mathscr{D}_s = \{\{\vec{x}_1, y_1\}, \ldots, \{\vec{x}_n, y_n\}\}$ sampled from $\mathscr{X} \times \mathscr{Y}$ following a (unknown) probability distribution $Pr(\vec{x}, y)$. We also have a set of m testing samples $\mathscr{D}_t = \{\vec{z}_1, \ldots, \vec{z}_m\}$ sampled from $\mathscr{X}$ following a (unknown) probability distribution $Pr'(\vec{x}, y)$, where the corresponding outputs from $\mathscr{Y}$ are unavailable, or *hidden*, and must be predicted. We assume that $\mathscr{D}_s$ are i.i.d. sampled according to the distribution $Pr(\vec{x}, y)$ and $\mathscr{D}_t$ are i.i.d. sampled according to the distribution $Pr'(\vec{x}, y)$. In standard supervised learning, we assume that $Pr(\vec{x}, y) = Pr'(\vec{x}, y)$. The problem of *large margin transductive transfer learning* is to learn a classifier that accurately predicts the outputs (class labels) for the unlabeled testing data set when $Pr(\vec{x}, y)$ and $Pr'(\vec{x}, y)$ are different.

3.2 Related work

There are two main approaches to transfer learning that have been considered, inductive transfer learning, where a small number of labeled test data are used along with labeled training data [4], and transductive transfer learning, where a significant number of unlabeled testing samples are used along with the labeled training data. Here we focus on transductive transfer learning.

A common approach to transfer learning is a model-based approach in which the different distributions are incorporated in a model, e.g., through domain specific priors [41] or through a model with general and domain-specific components [59]. Several approaches have also been developed for transductive transfer learning which consider the local structure of the unlabeled data, utilizing some unsupervised learning methods, such as clustering [69] or co-clustering [196]. Our approach is most similar to feature-based approaches to transfer learning, which include such approaches as weighting features to find feature subsets [165] or feature subspaces [122, 140] that generalize well across distributions. The difference is that we do so in a regularization framework, which aims to avoid over fitting and minimize the generalization error. Another approach that is similar to ours is that of Bickel *et al.* [20]. They address the problem of covariate shift through a likelihood model approach that takes into account the discrepancy between train and test distributions. However their method results in a logistic regression based classifier from a non-convex problem, whereas our approach results in an SVM classifier from a convex problem.

At the heart of our approach is the goal of finding a feature transform such that the distance between the testing and training data distributions, based on some distribution distance measure, is minimized, while at the same time maximizing a class distance or classification performance criterion for the training data. There has also been work describing how to measure the distance between distributions. A key idea is that the distance between two distributions can be measured with respect to how well they can be separated, given some function class. For instance, Ben-David *et al.* [15] used as an example the class of hyperplane classifiers and showed that the performance of the hyperplane classifier that could best separate the data could provide a good method for measuring distribution distance for different data representations. Along these same lines, Gretton *et al.* [76] showed that for a specific function class, the measure simplifies to a form that can be easily computed, the distance between the two means of the distributions, resulting in the maximum mean discrepancy (MMD) measure, which we use here. The particular form of this measurement makes it easier to incorporate into optimization problems, and so we chose this formulation to estimate distribution distances.

All the methods cited previously, including transfer learning, are closely related to multi-task learning and may be viewed as a special case of semi-supervised learning where unlabeled data is used to enhance the learning of a decision function. The difference is that in transfer learning, there is an assumed bias between training and testing samples. A recent review of semi-supervised learning may be found in [38, 225]. A discussion of possible sample bias, in a multi-task learning framework, may be found in [96, 175].

3.3 Background

3.3.1 Large Margin Classifier

Here we briefly discuss the formulation of the standard support vector machine (SVM), since it forms the basis for our transductive transfer support vector machine. Given $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n) \in \mathscr{X} \times \{\pm 1\}$ the supervised binary classification learning task is to learn a function $f^*(\vec{x})$ for any $\vec{x} \in \mathscr{X}$ that correctly predicts its corresponding class label y; of particular interest is *generalization accuracy* the accuracy of the function on predicting unseen future data. For hyperplane classifiers such as the SVM, the decision function is given by the function $f^*(\vec{x}) = \text{sign}(f(\vec{x}) + b)$, where $f(\vec{x}) = \vec{w}^T \vec{x}$, and $\vec{w}$ controls the orientation of the hyperplane, and b the offset. For the separable case, in which the two classes of data can be separated by a hyperplane, the SVM method tries to find the hyperplane with the maximum margin of separation, where the margin is the distance to the hyperplane of a point closest to the hyperplane. For the non-separable case, the SVM method tries to identify the hyperplane with the maximal margin with slack variables called the soft-margin. It can be shown that selecting the hyperplane with the largest margin minimizes a bound on expected generalization error [190].

The binary soft-margin SVM formulation aims to learn a decision function f specified below:

$$f = \underset{f \in \mathscr{H}_K}{\arg\min} \quad C \sum_{i=1}^{n} V(\vec{x}_i, y_i, f) + \frac{1}{2} ||f||_K^2 \tag{3.1}$$

where $K(\vec{x}, \vec{x}') : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ is a kernel function which defines an inner product (dot product) between samples in $\mathscr{X}$, $\mathscr{H}_K$ is the set of functions in the kernel space, $||f||_K^2$ is the L_2 norm of the function f, and C is a regularization coefficient. V measures the fitness of the function in terms of predicting the class labels for training samples and is called a risk function. The hinge loss function is a commonly used risk function in the form of $V = (1 - y_i f(\vec{x}_i))_+$ and $x_+ = x$ if $x \geq 0$ and zero otherwise.

If the decision function f is a linear function represented by a vector $\vec{w}$, equation 3.1 can be represented as:

$$
\begin{aligned}
\text{min.} \quad & \tfrac{1}{2}||\vec{w}||^2 + C\sum_{i=1}^{n} \varepsilon_i \\
\text{s.t.} \quad & \varepsilon_i \geq 0 \\
& y_i(\vec{w}^T \phi(\vec{x}_i) + b) \geq 1 - \varepsilon_i \quad \forall i = 1, ..., n
\end{aligned}
\tag{3.2}
$$

Where an unregularized bias term b is included and $\phi(\vec{x}_i)$ is the kernel feature vector of $\vec{x}_i$. Following common terminology (e.g., [172]) we refer to this as the 1-norm soft margin SVM, and if squared slack variables are penalized instead, i.e., $C\sum_{i=1}^{n} \varepsilon_i^2$, the 2-norm soft margin SVM.

3.3.2 Distribution Distance and MMD

For our formulation, it is necessary to choose a convenient distribution distance measure. One popular distribution "distance" measure is the Kullback-Leibler divergence, based on entropy calculations. However for our approach we need a nonparametric method suitable for a reproducing kernel Hilbert space (RKHS) that is both efficient to compute and relatively easy to incorporate into optimization problems while still allowing accurate distance measurement. One method that has recently been shown to be both efficient and effective for estimating the distance between two distributions in a reproducing kernel Hilbert space is the maximum mean discrepancy (MMD) measure [76]. The measure derives from computing the distribution distance by finding the function from a given class of functions that can best separate the two distributions, with the function class restricted to a unit ball in the RKHS. Additionally the particular form of this measure fits quite well into our support vector formulation, as shown in Section 3.4. Here we briefly overview the MMD measure for estimating the distance between two distributions. Given a set of n training samples $\mathscr{D}_s = \{\{\vec{x}_1, y_1\}, \ldots, \{\vec{x}_n, y_n\}\}$ and a set of m testing samples $\mathscr{D}_t = \{\vec{z}_1, \ldots, \vec{z}_m\}$. The (squared) maximum mean discrepancy distance of the training and testing distributions is given by the following formula:

$$\begin{aligned}
&\text{MMD}^2 = ||\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\phi(\vec{x}_i) - \tfrac{1}{m}\sum_{i=1}^{m}\phi(\vec{z}_i)||^2 \\
&= \tfrac{1}{n^2}\textstyle\sum_{i,j=1}^{n}K(\vec{x}_i,\vec{x}_j) + \tfrac{1}{m^2}\sum_{i,j=1}^{m}K(\vec{z}_i,\vec{z}_j) \\
&-2\tfrac{1}{nm}\textstyle\sum_{i,j=1}^{n,m}K(\vec{x}_i,\vec{z}_j)
\end{aligned} \tag{3.3}$$

The MMD measure has also recently been used in the context of transfer learning, e.g., for kernel learning [140].

3.4 Algorithm

Our general approach is as follows. We want to find a feature transform that minimizes the between-distribution distance, but at the same time maximizes the performance of a classifier on data from the training distribution. The latter criterion could also be considered a distribution distance measure (along the lines of [15]) in this case the distance between the distributions of the classes of the training data distribution. Thus in essence our general transfer learning approach is described with Equation 3.4.

$$f = \underset{f\in\mathscr{H}_K}{\arg\min}\quad C\sum_{i=1}^{n}V(\vec{x}_i,y_i,f) + \frac{1}{2}||f||_K^2 + \lambda d_{f,K}(Pr,Pr') \tag{3.4}$$

where Pr is the distribution of the training samples, Pr' the distribution of the testing samples, $d_{f,K}(Pr,Pr')$ is a distance measure of the two distributions, as evaluated against the decision function f and the kernel function K. λ controls the trade-off between the three components in the objective function. Other symbols such as $C,V,\mathscr{H}_K$ are the same as explained in Equation 3.1.

Following convention, we only consider linear decision functions f in the format $f(\vec{x}) = \vec{w}^T\phi(\vec{x})$ where $\vec{w}$ is the direction vector of f. Also following convention, we introduce an unregularized bias term, b, so that the final function is given by $f(\vec{x})+b$ and the label is assigned as $\text{sign}(f(\vec{x})+b)$.

3.4.1 Projected Distribution Distance

One approach we take to measure the distance between two distributions is to estimate how well the two distributions are separated as explored in the maximum mean discrepancy distance [76], mentioned previously. We define the *projected maximum mean discrepancy distance measure*, using a set of training samples $\mathscr{D}_s = \{\{\vec{x}_1, y_1\}, \ldots, \{\vec{x}_n, y_n\}\}$ and a set of m testing samples $\mathscr{D}_t = \{\vec{z}_1, \ldots, \vec{z}_m\}$ below. Here we take the squared projected maximum mean discrepancy measure for our distribution distance measure, to estimate the distribution distance under a given projection $\vec{w}$:

$$\begin{aligned} & d_{f,K}(Pr,Pr')^2 = ||\tfrac{1}{n}\textstyle\sum_{i=1}^{n} f(\vec{x}_i) - \tfrac{1}{m}\textstyle\sum_{j=1}^{m} f(\vec{z}_j)||^2 \\ & = \tfrac{1}{n^2}(\textstyle\sum_{i=1}^{n} \vec{w}^T \phi(\vec{x}_j))^2 + \tfrac{1}{m^2}(\textstyle\sum_{j=1}^{m} \vec{w}^T \phi(\vec{z}_j))^2 \\ & -2\tfrac{1}{nm}\textstyle\sum_{i,j=1}^{n,m} \vec{w}^T \phi(\vec{x}_i)\vec{w}^T \phi(\vec{z}_j) \end{aligned} \tag{3.5}$$

With the given decision and distance functions, we can rewrite Equation 3.4 in vector format below:

$$\begin{aligned} \text{min.} \quad & \tfrac{1}{2}||\vec{w}||^2 + C\sum_{i=1}^{n} \varepsilon_i + \lambda d_{f,K}(Pr,Pr')^2 \\ \text{s.t.} \quad & \varepsilon_i \geq 0, \quad y_i(\vec{w}^T \phi(\vec{x}_i) + b) \geq 1 - \varepsilon_i \quad \forall i = 1, ..., n \end{aligned} \tag{3.6}$$

where $d_{f,K}(Pr,Pr')^2$ is estimated using Equation 3.5.

The major difficulty in solving Equation 3.6 is that $\vec{w}$ is a vector in the Hilbert space defined by the kernel function K and hence may have infinite dimensionality. The Representer Theorem, which states that any vector $\vec{w}$ that minimizes Equation 3.6 should be a linear combination of the kernel feature vectors of the training and testing samples, provides a useful remedy.

$$\vec{w} \quad = \quad \textstyle\sum_{i=1}^{n} \beta_i \phi(\vec{x}_i) + \textstyle\sum_{j=1}^{m} \beta'_j \phi(\vec{z}_j) \tag{3.7}$$

where β_i and β'_j are coefficients and $\vec{w}$ is the vector that optimizes Equation 3.6. For simplicity, we denote

$$\phi(S) = (\phi(\vec{s}_1), \ldots, \phi(\vec{s}_{n+m})) = (\phi(\vec{x}_1), \ldots, \phi(\vec{x}_n), \phi(\vec{z}_1), \ldots, \phi(\vec{z}_m))$$

is a list of kernel feature vectors for training and testing samples and

$$\vec{\beta} = (\beta_1, \ldots, \beta_n, \beta'_1, \ldots, \beta'_m)^T$$

is a $(n+m)$ column vector. Hence we have $\vec{w} = \phi(S)\vec{\beta}$.

The key observation of the Representer Theorem is that if $\vec{w}$ has a component that is not in the span of column vectors in $\phi(S)$, that component must be orthogonal to the linear space spanned by the training and testing samples. In that case, the value of f, evaluated on training and testing samples will remain unchanged but the L_2 norm of f will increase [13]. The details of the formal proof in this case can be found in the appendix. With the Representer Theorem, we state our algorithm for large margin transductive transfer learning below.

3.4.2 Large Margin Transductive Transfer Learning Algorithm

With the Representer Theorem, we learn the decision boundary without explicitly learning the vector $\vec{w}$. We have the following observations.

$$||\vec{w}||^2 \;=\; \vec{\beta}^T \phi(S)^T \phi(S) \vec{\beta} \;=\; \vec{\beta}^T \Lambda \vec{\beta} \tag{3.8}$$

where Λ is a $(n+m)$ by $(n+m)$ positive semi-definite matrix and $\Lambda^{i,j} = K(\phi(\vec{s}_i), \phi(\vec{s}_j))$. Our projected distribution distance measure can then be expressed as:

$$\begin{aligned}
& d_{f,K}(Pr,Pr')^2 \\
= \;& \tfrac{1}{n^2}(\textstyle\sum_{i=1}^{n}\vec{w}^T\phi(\vec{x}_i))^2+\tfrac{1}{m^2}(\textstyle\sum_{j=1}^{m}\vec{w}^T\phi(\vec{z}_j))^2 \\
& -\tfrac{2}{nm}\textstyle\sum_{i,j=1}^{n,m}\vec{w}^T\phi(\vec{x}_i)\vec{w}^T\phi(\vec{z}_j) \\
= \;& \tfrac{1}{n^2}\textstyle\sum_{i,j=1}^{n}\vec{\beta}^T\phi(S)^T\phi(\vec{x}_i)\vec{\beta}^T\phi(S)^T\phi(\vec{x}_j)+ \\
& \tfrac{1}{m^2}\textstyle\sum_{i,j=1}^{m}\vec{\beta}^T\phi(S)^T\phi(\vec{z}_i)\vec{\beta}^T\phi(S)^T\phi(\vec{z}_j)- \\
& \tfrac{2}{nm}\textstyle\sum_{i,j=1}^{n,m}\vec{\beta}^T\phi(S)^T\phi(\vec{x}_i)\vec{\beta}^T\phi(S)^T\phi(\vec{z}_j) \\
= \;& \tfrac{1}{n^2}\vec{\beta}^T[\textstyle\sum_{i,j=1}^{n}(\phi(S)^T\phi(\vec{x}_i)\phi(\vec{x}_j)^T\phi(S))]\vec{\beta}+ \\
& \tfrac{1}{m^2}\vec{\beta}^T[\textstyle\sum_{i,j=1}^{m}\phi(S)^T\phi(\vec{z}_i)\phi(\vec{z}_j)^T\phi(S)]\vec{\beta}- \\
& \tfrac{2}{nm}\vec{\beta}^T[\textstyle\sum_{i,j=1}^{n,m}\phi(S)^T\phi(\vec{x}_i)\phi(\vec{z}_j)^T\phi(S)]\vec{\beta} \\
= \;& \tfrac{1}{n^2}\vec{\beta}^T K_{\text{Train}}[1]^{n\times n}K_{\text{Train}}^T\vec{\beta}+\tfrac{1}{m^2}\vec{\beta}^T K_{\text{Test}}[1]^{m\times m}K_{\text{Test}}^T\vec{\beta} \\
& -\tfrac{1}{nm}\vec{\beta}^T(K_{\text{Train}}[1]^{n\times m}K_{\text{Test}}^T+K_{\text{Test}}[1]^{m\times n}K_{\text{Train}}^T)\vec{\beta} \\
= \;& \vec{\beta}^T\Omega\vec{\beta}
\end{aligned} \tag{3.9}$$

where Ω is a $(n+m)\times(n+m)$ symmetric positive semi-definite matrix. K_{Train} is the $(n+m)\times n$ kernel matrix for the training data, K_{Test} the $(n+m)\times m$ kernel matrix for the testing data, and $[1]^{k\times l}$ is a $k\times l$ matrix of all ones.

With these two equations, Equation 3.6 is expressed using $\vec{\beta}$ in the following way:

$$\begin{aligned}
\text{min.} \quad & \vec{\beta}^T(\tfrac{1}{2}\Lambda+\lambda\Omega)\vec{\beta}+C\sum_{i=1}^{n}\varepsilon_i \\
\text{s.t.} \quad & \varepsilon_i \geq 0 \\
& y_i(\vec{\beta}^T K_i+b)\geq 1-\varepsilon_i \quad \forall i=1,\ldots,n
\end{aligned} \tag{3.10}$$

where $K_i=\phi(S)^T\phi(\vec{x}_i)$ is an $(n+m)$ column vector.

It is easy to show that the optimization problem of Equation 3.10 has an objective with a quadratic form of $\vec{\beta}$ and is a standard convex quadratic program, and hence can be solved using quadratic program solvers.

3.4.2.1 Regularization of the Hilbert space basis coefficients

We can view the problem of Equation 3.10 as performing regression in the Hilbert space with a hinge loss function and parameters $\vec{\beta}$. Thus we propose adding an L_2 penalty to the $\vec{\beta}$ parameters to shrink the selection of the data points used for the classifier and to add numerical stability to the algorithm in practical implementations - particularly with large matrices this can correct for slight negative eigenvalues from calculating Ω. Thus our final objective to minimize is:

$$\vec{\beta}^T(\frac{1}{2}\Lambda+\lambda\Omega+\lambda_2 I)\vec{\beta}+C\sum_{i=1}^{n}\varepsilon_i, \tag{3.11}$$

where I is the $(n+m)\times(n+m)$ identity matrix. In our experiments we found that generally a moderate amount of such L_2 regularization improved performance.

3.4.3 Simplification with Linear Kernel, Linear Feature Weighting

Below we show a special case with linear kernels and a feature weighting as opposed to a projection for measuring the distribution distance and demonstrate that in this case our algorithm can be viewed as a processing technique, following by a regular SVM model construction. We arrive at this simplification if we consider the target projection $\vec{w}$ as representing a linear feature weighting transform $W = diag(\vec{w})$ that does not project a data point but re-weights it, and measure the MMD with respect to the feature weighting introduced for a given $\vec{w}$ and the resulting W. With linear kernels, $\vec{w}$ is a vector in the original feature space, rather than in the kernel feature space, and the MMD measure under this linear transform is given by equation 3.12.

$$\text{MMD}^2 = (\tfrac{1}{n}\textstyle\sum_{i=1}^{n} W\vec{x}_i - \tfrac{1}{m}\sum_{j=1}^{m} W\vec{z}_j)^2 \tag{3.12}$$

We can rearrange the MMD measure to sum across each feature:

$$\begin{aligned}\text{MMD}^2 &= \textstyle\sum_{k=1}^{p} w_k^2(\tfrac{1}{n}\sum_{i=1}^{n} x_{ik} - \tfrac{1}{m}\sum_{j=1}^{m} z_{jk})^2 \\ &= \vec{w}^T Q\vec{w}\end{aligned} \tag{3.13}$$

where p is the dimensionality of $\vec{x}$ and Q is a $p \times p$ diagonal matrix with $Q_{k,k} = (\frac{1}{n}\sum_{i=1}^{n} x_{ik} - \frac{1}{m}\sum_{j=1}^{m} z_{jk})^2$ for $k \in [1, p]$.

Plugging this back into our 1-norm soft-margin SVM formulation, we can combine the MMD^2 term with the maximum margin term, resulting in the objective:

$$\text{min.} \quad \tfrac{1}{2}\vec{w}^T Q' \vec{w} + C\textstyle\sum_{i=1}^{n} \varepsilon_i \tag{3.14}$$

where I is a $p \times p$ identity matrix and $Q' = \lambda Q + \frac{1}{2}I$.

We could derive a similar quadratic programming for computing $\vec{w}$ but it is unnecessary. The problem presented in Equation 3.14 can be solved using a pre-processing step, followed by any off-the-shelf SVM solver. To see this, notice that since Q' is diagonal it can be expressed as $U^T U$ with $U = Q'^{\frac{1}{2}}$ so that $\vec{w}^T Q' \vec{w}$ becomes $\vec{w}^T U^T U \vec{w} = (U\vec{w})^T (U\vec{w})$. Thus by defining $\vec{w}'$ = $U\vec{w}$ and re-scaling the data by $1/U$ (i.e., $\vec{x}_i' = \vec{x}_i(1/U)$), we obtain the standard SVM problem. To obtain $\vec{w}$ from the solution $\vec{w}_*'$ we simply divided by U. Note that we can incorporate nonlinearity in this case through basis expansion; we simply define the feature f_j for a given $\vec{x}$ as the output of the kernel function between $\vec{x}$ and the data instance (from the training and testing sets) $\vec{s}_j$, $j \in \{1, \ldots, n+m\}$.

3.4.4 2-Norm Soft Margin Transductive Transfer Learning with Generalized Singular Value Decomposition

In the previous sections, we discussed the SVM with 1-norm soft margin for transductive transfer learning. In this section, we introduce a similar formalization for 2-norm soft margin transductive transfer learning that is equivalent for the case of the standard SVM, in which we fix the hyperplane norm $||\vec{w}||$ and find the hyperplane direction that gives maximum separation, measured by γ. This formalization reveals a geometric interpretation for the regularization. We discuss the geometric interpretation using a technique known as generalized singular value decomposition (GSVD).

The 2-norm transductive transfer learning is an optimization problem specified below:

$$\begin{aligned} \text{min.} \quad & -\gamma + \lambda \, \text{MMD}^2 + C\textstyle\sum_{i=1}^{n} \varepsilon_i^2 \\ \text{s.t.} \quad & y_i(\vec{w}^T \vec{x}_i + b) \geq \gamma - \varepsilon_i \quad \forall i = 1, ..., n \\ & ||\vec{w}|| = 1 \end{aligned} \tag{3.15}$$

With the Representer Theorem we have $\vec{w} = \vec{\beta}^T \phi(S)$ where $\phi(S) = (\phi(\vec{x}_1), \ldots, \phi(\vec{x}_n), \phi(\vec{z}_1), \ldots, \phi(\vec{z}_m))$.

Using the expression of MMD from Equation 3.9 and the L_2 norm of $\vec{w}$ in Equation 3.8, we have the following optimization problem:

$$\begin{aligned} \text{min.} \quad & -\gamma + \lambda \vec{\beta}^T \Omega \vec{\beta} + C\textstyle\sum_{i=1}^{n} \varepsilon_i^2 \\ \text{s.t.} \quad & y_i(\vec{\beta}^T K_i + b) \geq \gamma - \varepsilon_i \quad \forall i = 1, ..., n \\ & \vec{\beta}^T \Lambda \vec{\beta} = 1 \end{aligned} \tag{3.16}$$

The Lagrangian of Equation 3.16 is $L(w, b, \gamma, \alpha, \lambda, \lambda_0) = -\frac{1}{4C} \sum_{i=1}^{n} \alpha_i^2 - \frac{1}{4} \alpha_i y_i K_i^T M^{-1} K_i y_i \alpha_i - \lambda_0$ where $M = \lambda \Omega + \lambda_0 \Lambda$.

Clearly, if the value of λ_0 is known, the Lagrangian is a quadratic programming problem for α. The difficulty here is that we have to optimize two variables λ_0 and α. In regular SVM with 2-norm soft margin, the optimal value of λ_0 can be determined analytically once we know α and the optimization problem adopts the quadratic programming format. In transductive transfer learning, we do not have this convenience anymore. However, we may use a technique called generalized singular value decomposition to show the effect of the distribution distance measure Ω in the optimization.

For the kernel matrix Λ we obtain a matrix Γ_c such that $K = \Gamma_c^T \Gamma_c$. Similarly for the kernel matrix Ω we obtain a matrix Γ_d such that $K = \Gamma_d^T \Gamma_d$. Given two square-matrix Γ_c and Γ_d with the same size, if we apply the generalized singular value decomposition we have $\Gamma_c = U \Sigma_1 R Q^T$ and $\Gamma_d = V \Sigma_2 R Q^T$ where U, Q are orthogonal matrices and R is an upper-triangular matrix. Then we have the following formula:

$$\begin{aligned} M &= \lambda_0\Lambda + \lambda\Omega \quad = \quad \lambda_0\Gamma_c^T\Gamma_c + \lambda\Gamma_d^T\Gamma_d \\ &= \lambda_0 QR^T\Sigma_1^2 RQ^T + \lambda QR^T\Sigma_2^2 RQ^T \\ &= QR^T(\lambda_0\Sigma_1^2 + \lambda\Sigma_2^2)RQ^T \end{aligned} \tag{3.17}$$

We have $M^{-1} = QR^{(-1)}\frac{1}{(\lambda_0\Sigma_1^2 + \lambda\Sigma_2^2)}R^{(-1)T}Q^T$. Hence M^{-1} is a shrinkage operator, penalizing smaller generalized singular values and the penalization is controlled by the two parameters λ_0 and λ.

3.5 Synthetic Data Experiments

Here we give a synthetic 2D example to illustrate our approach. The training data distribution is shown as the green dots or squares (for the negative class) and the black plus symbols (as the positive class), generated by sampling from Gaussian distributions for each feature with $\sigma^2 = 1$, centered at $(0,-2)$ and $(2,0)$ respectively. The testing distribution is generated in a similar fashion, designed to be similar to the training distribution particularly along one dimension, with the negative class, depicted with upside-down red triangles generated from a Gaussian distribution centered at $(0,2)$ and the positive class, depicted as blue circles, generated with a Gaussian centered at $(2,0)$.

The transductive support vector machine is a widely used method that handled to some extent the possible difference between training and testing data sets. The transductive SVM tries to minimize the decision function norm and the errors on both the training and testing data, taking the unknown labels as variables of the optimization problem, so that these labels must be solved for along with the decision function. One of the key disadvantages of the transductive SVM is that the underlying optimization problem is an NP-hard problem and hence an iterative approximation has been used to solve it, which can take a very long time to finish. Our formalization of the transductive transfer SVM utilizes a quadratic programming optimization which is guaranteed to identify the global minimum in worst-case polynomial time.

The results for three versions of the support vector classifier are shown in Figure 3.2. The first is the standard support vector machine (green line), which performs the worst, obtaining an accuracy of .60, the second is the transductive SVM [100] (magenta line). The accuracy here improves to .72. Finally, the results of our transductive transfer SVM with a 1-norm soft margin are shown and the linear feature-weighting simplification (LMFW - red line), which tries to take into account the distance between the testing and training distributions. In this case it achieves the best accuracy, .84, and comes closest to finding the underlying ideal separation for a linear transform, a vertical line between the two classes.

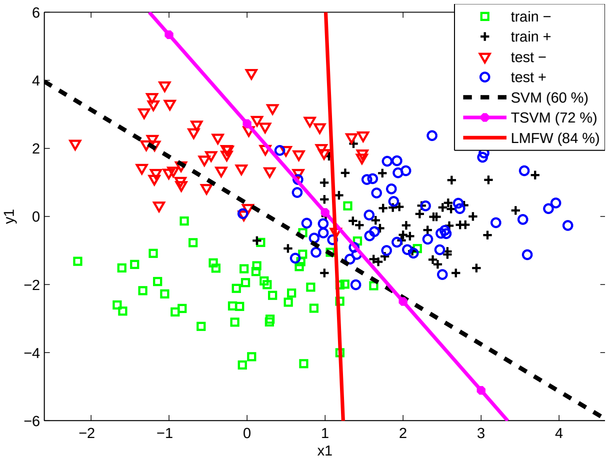


Figure 3.2: Performance of different support vector classifiers on a simple generated 2-D transfer learning problem.

The next example we give is for a nonlinear classification task. Here data of the negative class are generated around the origin by sampling 100 points from a Gaussian distribution that is stretched in one dimension and shrunken in the other, for the training data it is stretched along the x_2 axis, and for the test data along the x_1 axis. The positive class is then generated in each case by randomly sampling points from a uniform distribution in the box region around the negative class distributions. Points that are less than a fixed threshold when evaluated in the Gaussian function for the negative data distribution are discarded, and points are sampled until 100 are obtained. For all three methods we use default parameters of $\sigma = 0.5$ for the RBF kernel width and regularization parameter $C = 1$. The resulting classification boundaries learned by each of the three methods are shown in Figure 3.1, this time for our large-margin projection algorithm (LMPROJ). Our algorithm again achieves superior performance.

3.6 Real-World Data Experiments

Here we evaluate our methods using collections of real-world data. We use data from four different classification tasks, forming a combined total of 24 transfer learning data sets. Three of these tasks are commonly used in the literature and are related to text classification (work that used all or some of these data sets include [196, 69, 122, 140]). We include a fourth data set for transfer learning, related to protein-chemical interaction prediction.

Besides baseline methods of the standard support vector machine (SVM) and the transductive support vector machine (TSVM), we choose for comparison two recent state-of-the-art algorithms from KDD'08 that showed impressive results, out-performing baseline methods and some previous transfer learning methods in their experiments. The first comparison method is the Cross Domain Spectral Classifier (CDSC) [122] (out-performing the methods of [196] and [175] in their experiments). We implemented their method in Matlab, directly following the algorithm as presented in the paper. The second is the Locally-Weighted Ensemble (LWE) classifier of [69]. We used the same three methods that they used in their experiments for the ensemble, namely the Winnow algorithm from the SNoW Learning Architecture [Carlson et al.], a logistic regression algorithm from the BBR package [Genkin et al.] and the LIBSVM implementation of a support vector machine classifier [36]. We obtained parts of the code for their algorithm from an author's website `http://ews.uiuc.edu/~jinggao3/kdd08transfer.htm` and implemented the rest following the algorithm in their paper.

We obtained three pre-processed text classification data sets from the paper [69] for our experimental study: the Reuters data sets, 20 newsgroups text classification data sets, and the spam filtering data sets. We follow the sampling strategy in [122] to sample 500 instances each from the testing and training distribution to form our training and testing data sets.

We confirmed the correctness of our implementation by obtaining similar results to the performance reported in the respective papers (in some cases slightly more and in some cases slightly less accuracy). The methods we compared to did not list the type of normalization used, so we tried three different ways to normalize the non-binary features, no normalization, $[0,1]$ normalization using both the training and testing data, and $[0,1]$ normalization separately on the training and testing data. Interestingly, the performance of all the methods except LWE improved slightly using normalization, since normalization may interrupt the clustering structure in a data set. The difference between the second and the third normalization methods is negligible and hence we only report results on $[0,1]$ normalization separately on the training and testing data.

From our methods, we tested both the large-margin projection approach as described in Section 3.4.2 and Equation 3.10 and the large margin feature-weighting approach as described in Section 3.4.3. We denote the two approaches as LMPROJ and LMFW, respectively. We tested these two approaches as well as the basic SVM using a linear kernel and a cosine similarity measure, $K(\vec{x},\vec{y}) = (\vec{x}^T\vec{y})/(||\vec{x}||||\vec{y}||)$ the same similarity measure used by the CDSC method and commonly used in text mining. We only show results using the cosine similarity since they were slightly better than with the linear kernel. We used Matlab and a convex solver, `CVX` [74, 75], to solve the quadratic programs of the LMPROJ methods. For transductive transfer learning no labeled testing data can be used in the training, and since the testing and training distributions are different there is no easy way to use typical model selection approaches such as cross-validation to select appropriate parameters [69]. Thus we give the best performance for each method over a range of parameters, for the LWE and CDSC methods we center this range around the best performing parameters reported in their respective papers. Because of this, the base line SVM method and the transductive SVM method have higher accuracy as compared to those reported in the literature when default parameter values are used. We also perform detailed parameter sensitivity analysis to show how the performance is affected by each of the parameters in our method.

3.6.1 Evaluation Criteria

To compare the performance of the different methods, the first evaluation criterion we use is the F1 score, which is commonly used in information retrieval tasks such as document classification. The F1 score is the harmonic mean of the precision (P) and recall (R): $\frac{2PR}{P+R}$, where P is given by $\frac{tp}{tp+fp}$ and R by $\frac{tp}{tp+fn}$. tp denotes the number of true positive predictions, fp the number of false positives, fn false negatives, and tn true negatives. The F1 score is particularly appropriate for the spam filtering and chemical-protein interaction prediction data sets where predicting the positive class, the existence of spam and chemical-protein interaction respectively, is of particular interest. The second criterion we present results for is accuracy, commonly used to evaluate classification performance in general. Accuracy is given by $\frac{tp+tn}{tp+tn+fp+fn}$.

3.6.2 Data Sets

A brief description of each data set and its set-up is given here. Table 3.3 in the Appendix summarizes the data sets and gives the indexes by which we will refer to each in our results. For example, data set 10 is an email spam filtering data set where the training data set is a set of public messages and the testing data set is the set of emails collected from a specific user.

3.6.2.1 Reuters and 20 Newsgroups (Data sets 1 - 9)

These data sets both represent text categorization tasks, Reuters is made up of news articles with 5 top-level categories, among which, *Orgs*, *Places*, and *People* are the largest, and the 20 Newsgroups data set contains 20 newsgroup categories each with approximately 1000 documents. For these text categorization data, in each case the goal is to correctly discriminate between articles at the top level, e.g., "sci" articles vs. "talk" articles, using different sets of sub-categories within each top-category for training and testing, e.g., sci.electronics and sci.med vs. talk.politics.misc and talk.religion.misc for training and sci.crypt and sci.space vs. talk.politics.guns and talk.politics.mideast for testing. For more details about the sub-categories, see [196]. Each set of sub-categories represents a different domain in which different words will be more common. Features are given by converting the documents into bag-of-word representations which are then transformed into feature vectors using term frequency, details to this procedure can also be found in [196].

3.6.2.2 Spam Filtering (Data sets 10 - 12)

For this task, there is a large quantity of public email messages available, but an individual's emails are generally kept private, and these messages will have different word distributions. The goal is to use the publicly available email messages to learn to detect spam messages, and transfer this learning to individual users' email messages. There are three different users with associated email messages. The features for this data set are also made using term frequency from bag-of-word representations for the messages, details can be found in [19].

3.6.2.3 Protein-Chemical Interaction (Data sets 13 - 24)

For this data set, we test the ability of the algorithms to transfer learning across protein families for protein-chemical interaction prediction. The goal is to be able to use the known protein-chemical interactions for a given protein family to help predict which chemicals the proteins of another protein family will interact with, for which no interaction information is known. We obtained a data set from Jacob *et al.* [98] which includes all chemicals and their G protein-coupled receptor (GPCR) targets, built from an exhaustive search of the GPCR ligand database GLIDA [138]. The data set contains 80 GPCR proteins across 5 protein families, 2687 compounds, and a total of 4051 protein-chemical interactions. One family we discard since it has too few proteins and interactions. For the proteins we extracted features using the signature molecular descriptors [99], for the chemicals we used a frequent subgraph feature representation approach [95, 180], and we used a threshold on the feature frequencies to obtain about 100 features each. We then built the feature vector for a given protein-chemical pair by taking the tensor product between the protein and chemical feature vectors.

For each protein family we then built a data set by sampling 500 pairs of proteins from the

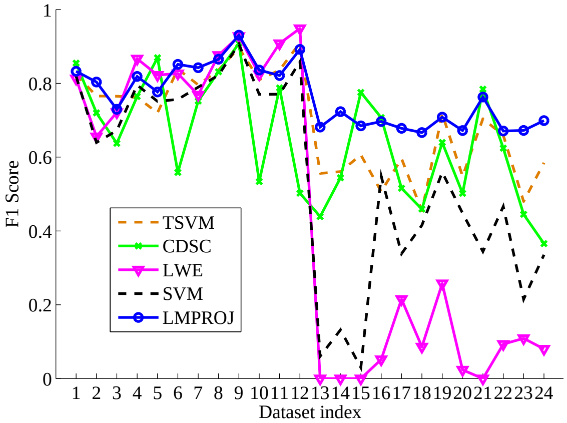


Figure 3.3: Prediction F1 score on all 24 data sets

family and chemicals that are known to interact (or took all available interactions for a given family if there were less than 500). Since we had no "negative interaction" data we randomly sampled the same number of protein-chemical pairs among the proteins of the given family and the chemicals for which there was no known interaction, the assumption being that the positive interactions are scarce. We then constructed 12 transfer learning tasks by using each protein family in turn as the training domain and each other protein family for the testing domain. The break-down of the protein families is shown in Table 3.3 in the Appendix.

3.6.3 Experimental Results

First, we show an overall comparison of our method with the two state-of-the-art methods we compared with as well as the baseline of a SVM classifier with a cosine similarity kernel and the off-the-shelf transductive SVM. For easy visualization we show a plot of the F1 scores in Figure 3.3 with the data set index on the x-axis and the F1 score on the y-axis for the different methods, only showing here our method LMPROJ with the cosine similarity kernel (though the LMFW method was comparable, as seen in Tables 3.1 and 3.2) marked by blue circles, the LWE method marked by upside-down purple triangles, the CDSC method marked by green crosses, transductive SVM (TSVM) by a dashed orange line, and traditional SVM by the dotted black line. The results for accuracy are reported in Tables 3.1 and 3.2.

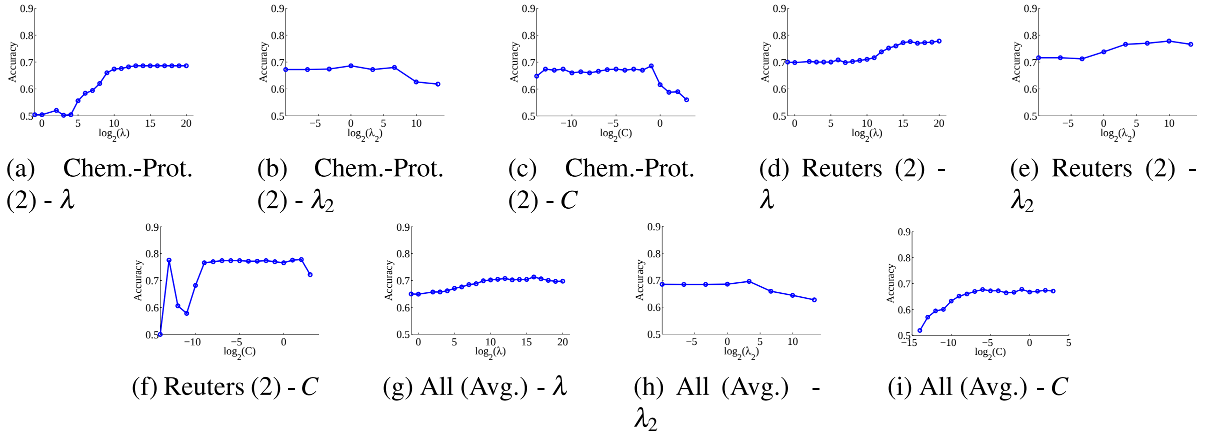
In Figure 3.3, we observe that there is a general agreement of all 5 different methods that we compared in the first 12 data sets. The chemical-protein interaction data sets are harder and there is a large performance gap between different methods. Specifically comparing different methods, the base-line SVM works almost always the worst. This is not surprising since we know there are differences between training and testing samples and ignoring such differences usually does not lead to optimal modeling.

The cross-domain spectral classifier method (CDSC) has competitive performance, as compared to other methods. For some reasons that we do not fully understand, we observe a large performance variation of the CDSC method across different data sets. The locally weighted ensemble method (LWE) and the transductive SVM (TSVM) method have competitive performance in the first 12 data sets but they do not perform very well in the chemical-protein data sets. The results may suggest that the chemical-protein interaction data do not follow the clustering assumption well.

We observe that the LMPROJ method delivers stable results across the 24 data sets. For both accuracy and F1 score LMPROJ achieves the best score in 11 out of 24 data sets and is competitive with the best methods for the majority of the other data sets. It obtains the best score more times than any of the other methods.

We also note that we obtained somewhat better results for the SVM and TSVM methods than typically reported in the literature (e.g., [69, 122]) on the same data sets that we use. This is because in our study instead of selecting a default parameter or allowing an internal cross-validation on the training data to be performed, to allow a fair comparison with the transfer learning approaches we reported the best results over a set of parameters for the baseline methods.

Next we give parameter sensitivity results in Figure 3.4, for the accuracy criterion and the three parameters λ, λ_2, and C. For each plot, two parameters are fixed at the best values while the third parameter is varied to generate the plots. Here we show representative results for a couple of data sets, the 2nd Reuters data set - a text data set, and the second chemical-protein interaction data set. In the last three subfigures we also show the sensitivity results for the three parameters averaged over all 24 data sets. While the base accuracy was different for different data



(a) Chem.-Prot. (2) - λ (b) Chem.-Prot. (2) - λ_2 (c) Chem.-Prot. (2) - C (d) Reuters (2) - λ (e) Reuters (2) - λ_2

(f) Reuters (2) - C (g) All (Avg.) - λ (h) All (Avg.) - λ_2 (i) All (Avg.) - C

Figure 3.4: Parameter Sensitivity

Table 3.1: Accuracies for All Methods on Text Classification Datasets

Methods	Reuters			20 Newsgroup						Spam Filtering		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
SVM	0.80	0.70	0.68	0.79	0.76	0.78	0.76	0.84	0.91	0.77	0.77	0.85
TSVM	0.82	**0.78**	**0.73**	0.76	0.73	0.84	0.80	0.84	0.90	0.81	0.84	0.91
CDSC	**0.86**	0.75	0.67	0.71	**0.87**	0.66	0.73	0.83	0.90	0.68	0.82	0.56
LWE	0.81	0.71	0.66	**0.87**	0.79	0.84	0.70	**0.87**	0.92	**0.84**	**0.91**	**0.95**
LMFW	0.81	0.75	0.70	0.79	0.76	0.82	0.78	0.85	0.92	0.77	0.78	0.87
LMPROJ	0.83	**0.78**	0.71	0.81	0.77	**0.85**	**0.84**	0.87	**0.93**	**0.84**	0.82	0.90

Table 3.2: Accuracies for All Methods on Protein-Chemical Datasets

Methods	Protein-Chemical Interaction											
	13	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
SVM	0.50	0.53	0.51	0.55	0.49	0.46	0.66	0.50	0.54	0.61	0.49	0.52
TSVM	0.56	0.56	0.61	0.51	**0.60**	0.45	**0.72**	0.55	0.72	0.66	0.48	0.57
CDSC	0.54	0.60	**0.78**	**0.72**	0.54	0.50	0.70	0.53	**0.80**	**0.70**	0.49	0.52
LWE	0.50	0.50	0.50	0.51	0.52	0.50	0.56	0.50	0.50	0.52	0.51	0.50
LMFW	0.56	0.63	0.74	0.60	0.54	0.56	0.66	0.54	0.75	0.57	0.49	0.63
LMPROJ	**0.58**	**0.69**	0.69	0.66	0.58	**0.61**	0.69	**0.56**	0.69	0.64	**0.53**	**0.63**

sets, the general trends are captured by averaging the results together. In general we see that as we suspected larger values of λ tend to improve performance; as λ is increased, the performance increases from the base standard SVM performance, and levels off to a maximum for a wide range of parameters. The results for λ_2 show that in general the L_2 regularization slightly improves performance up to moderate amounts, but past a certain point, i.e., too much regularization, the performance deteriorates. Also the performance is relatively insensitive to C for a wide range of values.

Finally the full results including a comparison of all the methods tested in terms of accuracy are given in Table 3.1 and Table 3.2.

3.7 Discussion and Future Work

We have addressed the problem of transductive transfer learning using regularization with the goal of maximizing a classification margin while at the same time minimizing a distance between training and testing distributions. With extensive experimental study we demonstrated the effectiveness of our approach, comparing it with some recent state-of-the-art methods. Our results demonstrate the effectiveness of this viewpoint of using regularization to find a decision function that brings the training and testing distributions together so that the training data can be effectively utilized.

One key idea for future work is incorporate an L_1 penalty on $\vec{\beta}$ of the projection method to encourage a sparse solution. Also, an open problem for transductive transfer learning in general is how to perform parameter selection, since no labeled testing data is available. Another area of future work is to experiment with different loss functions for our large-margin classifier, in particular, a truncated hinge-loss function (e.g., [200]), to avoid situations where errors on the training data effectively prevent the transfer to the test domain. Finally, from our results we have seen that two schools of thought for considering transfer learning problems, one which tries to match the structure of the testing data and the other which tries to find some type of transform/embedding that brings the testing and training data together, seem to some extent to provide complementary results. Forming a hybrid method could potentially result in a more powerful classifier.

3.8 Appendix

3.8.1 Characteristics of Data Sets

Details for the transfer learning tasks are provided in Table 3.3.

3.8.2 Representer Theorem

The major difficulty in solving Equation 3.6 is that $\vec{w}$ is a vector in the Hilbert space defined by the kernel function K and hence may have infinite dimensionality. Fortunately we have the following

Table 3.3: Break down of data sets

Set Ind.	**Task**	**Training**	**Test**
1	Orgs v. People	(Reuters) Documents from sub-categories	Documents from different sub-categories
2	Orgs v. Place		
3	People v. Place		
4	Comp v. Sci	(20 Newsgroups) Documents from sub-categories	Documents from different sub-categories
5	Rec v. Talk		
6	Rec v. Sci		
7	Sci v. Talk		
8	Comp v. Rec		
9	Comp v. Talk		
10	Email Spam Filtering	Public messages	User1's emails
11			User2's emails
12			User3's emails
13	Cross-family protein-chemical interaction prediction	Rhodopsin peptide receptors	Rhodopsin amine receptors
14		Rhodopsin peptide receptors	Rhodopsin other receptors
15		Rhodopsin peptide receptors	Metabotropic glutamate family
16		Rhodopsin amine receptors	Rhodopsin peptide receptors
17		Rhodopsin amine receptors	Rhodopsin other receptors
18		Rhodopsin amine receptors	Metabotropic glutamate family
19		Rhodopsin other receptors	Rhodopsin peptide receptors
20		Rhodopsin other receptors	Rhodopsin amine receptors
21		Rhodopsin other receptors	Metabotropic glutamate family
22		Metabotropic glutamate family	Rhodopsin peptide receptors
23		Metabotropic glutamate family	Rhodopsin amine receptors
24		Metabotropic glutamate family	Rhodopsin other receptors

theorem, known as the Representer Theorem, which states that $\vec{w}$ is always a linear combination of $\phi(x_i)$ and $\phi(z_j)$ where x_i in $\mathscr{D}_s$ and z_j in $\mathscr{D}_t$. Below we prove that the Representer Theorem is correct in our case.

Theorem 3.8.1. *The vector $\vec{w}$ that minimizes the Equation 3.6 can be represented as*

$$\vec{w} \quad = \quad \sum_{i=1}^{n} \beta_i \phi(\vec{x}_i) + \beta'_j \sum_{j=1}^{m} \phi(\vec{z}_j) \tag{3.18}$$

where β_i and β'_j are coefficients.

Proof. We prove the theorem by showing contradiction. Let $\vec{w}_1 = \sum_{i=1}^{n} \beta_i \phi(\vec{x}_i) + \beta'_j \sum_{j=1}^{m} \phi(\vec{z}_j) + \vec{w}_\perp$ be a vector optimize the Equation 3.6 where $\vec{w}_\perp \notin span(\phi(\vec{x}_i), \phi(\vec{z}_j))$. And let $\vec{w}_0 = \vec{w}_1 - \vec{w}_\perp$ be the projection of $\vec{w}_1$ in the linear space of $span(\phi(\vec{x}_i), \phi(\vec{z}_j))$. Then we have

$$\begin{aligned} f_{w1}(x_i) &= \vec{w}1^T \phi(x_i) \\ &= \vec{w}_0^T \phi(x_i) + \vec{w}_\perp^T \phi(x_i) \\ &= \vec{w}_0^T \phi(x_i) \end{aligned} \tag{3.19}$$

And $||\vec{w}_1||^2$ = $||\vec{w}_0||^2 + ||\vec{w}_\perp||^2 \geq ||\vec{w}_0||^2$. If we compare $\vec{w}_1$ and $\vec{w}_0$, we claim that the hinge loss function values are exactly the same and the MMD regularizer values are exactly the same. The only difference is that the norm of $\vec{w}_1$ is larger than $\vec{w}_0$. This claim contradicts the original assumption that $\vec{w}_1$ optimizes Equation 3.6. Hence $\vec{w}_\perp = 0$.

□

Chapter 4

Preliminary Study III: Feature Extraction for Knowledge Transfer with Low-Quality Data

4.1 Introduction

Knowledge transfer, modeling data that are from related but not identically distributed sources, is a problem of fundamental importance in knowledge discovery and data engineering. It has been extensively demonstrated through experimental study that traditional modeling methods typically perform drastically worse when the identically distributed assumption no longer holds (e.g., [57, 55, 69, 140]). A recurring knowledge transfer scenario that arises naturally in many application domains is the task of using a set of often high-quality, labeled auxiliary data that is expensive to obtain, to help predict the labels of a set of new data believed to come from a different but similar distribution and having little or no label information.

Knowledge transfer (e.g., transfer learning, domain adaption, learning with out-of-domain data) has attracted significant research interest from the machine learning and data mining community [18, 69, 96, 152, 165, 185, 56]. Many learning and mining algorithms have been developed, including those based on exploring the clustering structure of data [69, 56], sampling strategies which select samples that are more likely coming from the same distribution [18, 96, 185], shared feature structure between the training data to testing data [152, 165], and latent variables for related tasks [114, 201, 203].

In this chapter we investigate the problem of knowledge transfer in a totally different direction and focus on preprocessing techniques that are widely used in data engineering research. In particular, we notice that effective representation of the original data is a critical but yet not fully explored research area for knowledge transfer. Feature extraction methods have been widely utilized in data engineering for creating a suitable representation for subsequent modeling practices. One of the most commonly used feature extraction methods is Principle Component Analysis (PCA) [85], in which an ordered orthogonal basis is found for a set of data with the first vectors in the basis capturing most of the variance in the data, and the projection of the data instances on some top number of basis vectors is taken as the extracted feature representation. PCA based methods have also been applied to perform feature extraction for knowledge transfer tasks (e.g., directly in [201], in a kernel space in [140], and for comparison in [152, 27]). The direct application of PCA based methods for knowledge transfer, however, usually does not lead to optimal results due to various reasons. First different distributions of source and target data may mislead the direction of the principle components. Second, for high dimensional data where data are often clustered in subspaces rather than the full space, PCA may not reveal the best representation of the data.

Towards the end goal of effective data representation, we develop a general approach to feature extraction and data representation based on a technique called sparse coding. Sparse coding is widely used in high-dimensional data preprocessing for identifying a (small) group of higher-order features of data from the raw representations [139, 152]. Such higher-order features are suitable for subsequent analysis including subspace clustering [63] and missing value imputation [31]. The limitations of sparse coding are that sparse coding still does not explicitly consider distribution distance and can result in poor embeddings for knowledge transfer.

To address the limitations and enable effective feature extraction for data that may come from different distributions, we extend sparse coding to incorporate a regularization term that can in effect be used to control how identical the distributions for different data sets are under the learned embedding. In this way we hope to obtain an underlying structure that allows easy knowledge transfer. We evaluate the proposed method with synthetic and real data experiments, including an application to drug toxicity prediction.

4.2 Related Work

4.2.1 Feature Extraction with Sparse Coding

Sparse coding itself has been used for transfer learning [152] the idea being that it is able to capture higher level features of the data which can then be used to allow knowledge transfer (see discussion in Section 4.3.3 for details).

Recently Xie *et al.* considered the related problem of transfer learning for data sets having differing but overlapping feature sets [201]. This is closely related work to the problem we consider here, and is a special case of transfer learning with missing values. They proposed to use the shared features to build regression models for predicting the missing values, then perform singular value decomposition to find a lower dimensional structure explaining the data and allowing the knowledge transfer. The approach has two key shortcomings. First, imputation and learning the embedding are performed separately, but the underlying structure is what explains the missing values so that the latent structure and imputation should be learned in tandem; from matrix completion theory we know finding the lowest rank matrix that matches the non-missing values allows perfect matrix completion under certain conditions [31]. Secondly, traditional embedding techniques like SVD used in the previous approach can actually find poor embeddings for transfer learning since they are designed to approximate the data well and do not explicitly consider trying to make the data IID, in fact as we demonstrate with simple synthetic examples in a later section, the embeddings found can actually hinder transfer learning. We also describe how our algorithm can handle missing value imputation in tandem with the embedding process, and test this sparse coding approach under a standard classification setting.

4.2.2 Transfer Learning and Domain Adaption

Many learning algorithms have been developed for knowledge transfer [142]. A common approach is a model-based approach in which the different distributions are incorporated in a model, e.g., through domain specific priors [41] or through a model with general and domain-specific components [59]. Several approaches have also been developed for transductive transfer learning which consider the local structure of the unlabeled data, utilizing some unsupervised learning methods, such as clustering [69] or co-clustering [56]. There are methods based on model selection, selecting features that generalize well across distributions [122, 140, 165]. The difference between feature selection and feature generation is that we want to "discover" new features, based on the existing features, for knowledge transfer and we do so in a regularization framework, which aims to avoid over-fitting and minimize the generalization error.

Aside from the sparse-coding approaches and those embedding approaches mentioned previously, there has been additional work on embedding, specifically using eigendecomposition, for knowledge transfer. Zhong *et al.* [217] proposed an approach consisting of choosing a kernel, decomposing, and then selected instances to include by considering distribution distance; however distribution distance is not incorporated in the embedding and useful instances could be thrown away - potentially only reinforcing a poor concept. Pan *et al.* [140, 141] proposed learning a kernel matrix with constraints on nearest neighbor distances and a distribution distance based regularization using maximum mean discrepancy (MMD) [76], followed by eigendecomposition. However they do not incorporate any class-based distribution distance, and we show that embedding by only incorporating distribution distance can actually mislead the embedding and result in worse performance than not incorporating distribution distance. A key reason for this is the embedding changes the conditional distributions for the different data sources, so even if they were the same before (often considered a requirement for domain adaptation approaches), after embedding they may no longer agree. Additionally, depending on the kernel, the MMD can fail to capture differences in distributions, for instance if the kernel matrix learned happens to correspond to a linear kernel two very different distributions can be considered similar if they have close means.

4.3 Methodology

4.3.1 Notation

We use the following notations throughout the rest of the chapter. We use lowercase letters to represent scalar values, lower-case letters with an arrow to represent vectors (e.g., $\vec{\beta}$), uppercase letters to represent matrices, and uppercase calligraphic letters to represent sets. Unless stated otherwise, all vectors are column vectors. We use $||A||_F$ to denote the Frobenius norm of a matrix A, $\sqrt{\mathrm{Tr}(A^TA)}$, where Tr denotes the trace; $||\vec{a}||_1$ denotes the L_1 norm of the k-dimensional vector $\vec{a}$, $\sum_{i=1}^{k}|a_i|$. Note, for convenience we use: $A_{:i}$ to denote the i^{th} column vector of the matrix A, $A_{i:}$ to denote the i^{th} row vector of the matrix A, and A_{ij} to denote the $(i,j)^{th}$ entry of A, and similarly a_i to denote the i^{th} entry, or coefficient, of the vector $\vec{a}$. Additionally matrix powers are taken as entry-wise powers, for example, A^2 denotes the matrix obtained by squaring each entry in A.

4.3.2 Preliminary Background on Sparse Coding

Given a set of n p-dimensional data points, $\{\vec{x}_1, \vec{x}_2, \ldots \vec{x}_n\} \subset \mathbb{R}^p$, we form the $p \times n$ data matrix X by taking $\vec{x}_i$ as column i, $i = 1, \ldots, n$. The goal of sparse coding is to learn a set of r p-dimensional basis vectors, $\{\vec{b}_1, \ldots \vec{b}_r\} \subset \mathbb{R}^p$ forming $p \times r$ basis matrix B with column $i = \vec{b}_i, i = 1, \ldots, r$, and a set of n r-dimensional sparse (having few non-zero values) weight vectors, $\{\vec{w}_1, \ldots \vec{w}_r\} \subset \mathbb{R}^p$ forming weight matrix W with column $i = \vec{w}_i, i = 1, \ldots, n$, that approximate the original patterns well, that is, $BW \approx X$. Assuming the reconstruction error for a data pattern $\vec{x} - B\vec{w}$ follows a zero-mean Gaussian distribution with covariance $\sigma^2 I$, and taking a Laplace prior for the weight coefficients and assuming a uniform prior on the basis vectors, then the posterior probability of the data for a given B and W is proportional to Equation 4.1.

$$\prod_{i=1}^{n} e^{-||\vec{x}_i - B\vec{w}_i||_2^2/(2\sigma^2)} e^{-\alpha||\vec{w}_i||_1} \tag{4.1}$$

The maximum a posteriori estimate for the basis and vectors can then be found by maximizing the log of Equation 4.1 with the following optimization problem [116] :

$$\begin{aligned} &\underset{B,W}{\arg\min} && \tfrac{1}{2\sigma^2}||X - BW||_F^2 + \alpha \sum_{i=1}^{n} ||\vec{w}_i||_1 \\ &\text{s.t.} && ||\vec{b}_i||_2^2 \leq c \quad \forall i = 1, ..., n \end{aligned} \tag{4.2}$$

where the constraints on the norm of the basis vectors are introduced to prevent them from growing infinitely large, and can be viewed as regularization on the basis vectors as well. Typically c is fixed, e.g., to 1, since allowing the basis norms to be bigger would allow the basis weights, the entries of W, to shrink (reducing the L_1 norm, and still produce the same reconstruction, so that the effect α has would change. Here α acts as a tunable regularization parameter, trading off between sparsity of the weights and approximation of X. The resulting new data representation is then given by W. We label this sparse coding feature extraction method as SC in our experiments.

The problem in 4.2 is non-convex, but fixing either W or B the problem becomes convex in the other (i.e., fix W and the problem is convex in B and vice versa). This was exploited in [116] along with a Lagrange dual solution for learning the basis to derive an efficient algorithm for solving this problem, by alternatively fixing W or B and solving for the optimal value of the other. We thus take a similar alternating optimization approach for our algorithms, as described in subsequent sections.

4.3.3 Advantages and Limitations of Sparse Coding for Feature Extraction in Knowledge Transfer

One benefit of sparse coding for knowledge transfer comes from the viewpoint of sparse coding as a way of learning higher-order more general representations of data from the given low level representations [139, 152]. By forcing the representations to be sparse combinations of the basis vectors it helps to ensure that the basis found is efficient at representing the set of patterns and generally captures the main patterns of interest in the low level input representations. The idea then is that while the low-level details for different data sets may be different, they will have some commonalities, or overlap, in the higher-level representation that allows general principles to be inferred in this higher order representation that are applicable to the different data sets. Such an approach has been applied to learning higher order representations for knowledge transfer using auxiliary data sources [152, 27, 117]. However the fundamental assumption here then must be that the data sets are identically distributed in this higher order representation - if they are not, then the higher order representation will still have the same issue as before - of non-identically distributed data, and will still not enable knowledge transfer. As it is, sparse coding provides no such guarantee.

Another way of viewing sparse coding which potentially offers more insight is from a geometric perspective; sparse coding can be viewed as a way of performing subspace clustering. By forcing the new data representations to be sparse the algorithm tries to find a set of representative vectors or directions with the representations only being active among a few of the basis vectors - the set of vectors for which a datum representation is non-zero could be seen as its subspace membership. It can be shown that if the data points lie in a set of independent subspaces, then sparse coding can be used to fully identify the subspace clusters [63]. In this sense sparse coding could be seen as being useful for knowledge transfer in the same sense as other cluster-based transfer learning methods: by identifying the shared cluster structure of the auxiliary data with the target data, it can in effect select only those auxiliary data belonging to the same clusters as the target data for extracting knowledge, or learning patterns, since only those data will have the same sets of features active as the target data. The active features in the new representations can then be viewed as the coordinates in the shared subspaces for the found basis. This ability to handle multimodal data is a major advantage of the sparse coding algorithm over other embedding algorithms such as principle component analysis [85] which only looks at directions of greatest variance completely missing any internal structure and further restricting all basis vectors found to be perpendicular. However a fundamental issue here with sparse coding comes from the case of target data and auxiliary data lying mostly in different subspaces. In the case of an auxiliary data set and a target data set lying in different subspaces, sparse coding will generally result in representations for which no active features are shared between the two data sets, since each will only have non-zero weights for those basis vectors belonging to its own subspace (see Section 4.4.1 for an illustration of this case). In this case no knowledge transfer is possible because the only non-zero features in the target data will always be zero in the auxiliary data, so the auxiliary data cannot be used to help determine patterns for those features and thus the target data. Nevertheless, just because the shared cluster assumption used by sparse coding and many other knowledge transfer methods no longer holds does not mean we should abandon our hope of utilizing available high-quality auxiliary data. In the next few sections we propose some modifications to sparse coding to allow knowledge transfer in such cases, and more generally for whenever the embedding found still does not result in identically distributed data.

Another issue with sparse coding comes from selecting the size of the basis. In an unsupervised setting where we learn a basis and weights that explain all of the data best, as we allow the basis to grow beyond a certain size, the possible generalization shrinks. It is easy to see that if we allow the basis dimension to equal the number of points, that a basis that minimizes the objective function is given by one basis vector in the direction of each input data point. First all basis vector norms will be maximized in order to allow minimum weights. Because the L1 penalty is used the additional penalty is the same for larger weight values, so the smallest weight possible always comes from a direct path to a data point. In this case sense each point would be assigned to its own coordinate, no patterns could be found from the data. As we allow the basis to grow, sparse coding basically becomes similar to a weighted k-nearest-neighbor algorithm [208].

4.3.4 Improving Sparse Coding with Regularization

A fundamental limitation as described in the last section is that sparse coding may actually find an embedding that hinders knowledge transfer - there is nothing forcing the data sets in the new feature representations to be identically distributed. Since our goal is to transfer knowledge when data distributions are not identical in order to utilize auxiliary data, it therefore makes sense to address this problem by trying to enforce the the embedded data sets to be identically distributed. To do this we propose to incorporate a distribution distance estimation between the embedded data sets. Following the regularized regression framework in Equation 4.2, to incorporate distribution distance, we add a tunable regularization term on the embedding weights for the two data sets that penalizes the estimated distribution distance between these sets of weights. This type of regularization could be viewed as a soft constraint that enforces the estimated distributions of the different data sets to be identical. The new optimization problem is given in Equation 4.3, where U and V are used to denote the weights for the training (source) and test (target) sets respectively, for convenience, p and q represent the probability density functions (pdfs) for each set respectively, and $d(,)$ some distribution distance function.

$$\begin{aligned} \underset{B,W}{\arg\min} \quad & ||X - BW||_F^2 + \alpha \sum_{i=1}^{n} ||\vec{w}_i||_1 + \beta d(p_U, q_V) \\ \text{s.t.} \quad & ||\vec{b}_i||_2^2 \leq c \quad \forall i = 1, ..., n \end{aligned} \tag{4.3}$$

Since the penalty only includes the weight terms, we can still perform the alternating optimization. Here β is another tunable regularization parameter which controls the importance given to enforcing small distribution distance. In this case most distribution distance measures will result in a non-convex problem for fitting W. Thus we can only find a local solution. Avoiding this non-convexity is an open problem since accurate distribution distance measures as functions of the finite-dimensional embedding can have multiple local minima (as illustrated in Section 4.4.1) unless simpler but also less accurate distribution distance measures are used. Note that since the distribution distance only depends on W the problem remains unchanged and is still convex when W is fixed.

In general, most probability distribution distance measures require the pdfs of the two distributions in question. One commonly used measure that is an exception is the maximum mean discrepancy (MMD) estimate [76, 140], that is useful in some kernel spaces, but in the original input space (i.e., with a linear kernel) provides only a weak measurement, for example, not being able to distinguish between two different distributions with the same mean. To use a more accurate distribution distance measure, we therefore need to estimate the pdfs of the two distributions. In order to do this we propose to use a nonparametric density estimation technique, kernel density estimation; this can be thought of as providing a smoothed histogram.

In general estimation tasks, the usefulness of kernel density estimation is somewhat limited due to the curse of dimensionality, with the risk of the estimator growing with the dimensionality of the data [195]. However in our case there are several benefits to using kernel density estimation. First, since we need to restrict the dimensionality of the data to some degree to allow generalization between data sources, this should alleviate to some extent the curse of dimensionality. Secondly, we are not actually concerned with estimating the densities, just determining a difference in the densities of two distributions and how this changes as the data changes, so as long as this difference and change is captured it doesn't matter how accurate the density estimation is. Finally, using a differentiable kernel function in the estimation enables straight-forward computation of derivatives which allows easy incorporation in standard optimization techniques like gradient descent. Since the specific kernel function chosen is not very important for kernel density estimation [195] we use the differentiable Gaussian kernel $k(\vec{x},\vec{y}) \propto \exp(-(1/(2h))||\vec{x}-\vec{y}||_2^2)$ where h is the kernel width, in our implementations.

With this approach we can then use a wide variety of distribution distance measures that use the pdfs, including f-divergences such as χ^2-divergence and Kullback-Leibler divergence and L_p-norm distance measures. Here we use the symmetric version of the common KL-divergence measure, the Jensen-Shannon divergence. The KL-divergence is given by $d_{KL}(P||Q) = E_P[log(p/q)]$ and JS-divergence is $d_{JS} = 0.5(d_{KL}(P||Q) + d_{KL}(Q||P))$. In general computing the KL-divergence for multivariate data with continuous variables is still an open problem, but by estimating the density we can then use the sample mean approximation to expected value given our data sample to predict the KL-divergence as the expected value of the log-odds of the pdfs. Below we derive expressions for the distance measure, and the gradient of the distance measure.

We use K, G, and S to denote the kernel matrices for U with itself, U with V and V with itself, e.g., G is an $n \times m$ matrix with entries $G(i,j) = \exp(-(1/(2h))||\vec{u}_i - \vec{v}_j||_2^2)$, where h is the kernel width. Then to calculate the probability vectors for each data set under each distribution, we have the following:

$$\begin{aligned} \vec{p}_u &= (1/(n(2\pi h)^{r/2}))K\vec{1}, \\ \vec{q}_u &= (1/(m(2\pi h)^{r/2}))G\vec{1}, \\ \vec{p}_v &= (1/(n(2\pi h)^{r/2}))G^T\vec{1}, \\ \vec{q}_v &= (1/(m(2\pi h)^{r/2}))S\vec{1}, \end{aligned} \tag{4.4}$$

where e.g., $\vec{p}_v$ represents the pdf for the first data set (U) evaluated at each point in the second data set V and $\vec{1}$ denotes a vector of all ones of the appropriate length. Then the JS divergence estimate is given with Equation 4.5.

$$d_{JS} = \frac{1}{2}(\vec{1}^T(log(\vec{p}_u) - log(\vec{q}_u))/n + \vec{1}^T(log(\vec{q}_v) - log(\vec{p}_v))/m) \tag{4.5}$$

Then the gradient for the l_{th} column of U and V is given in Equation 4.6.

$$\begin{aligned} \nabla_{\vec{u}_l} d_{JS} &= \frac{1}{2nh}(U - \vec{u}_l)(K_{:l}/\vec{p}_u + K_{:l}/p(\vec{u}_l)) \\ &\quad - \frac{1}{2mh}(V - \vec{u}_l)(G_{l:}^T/q(\vec{u}_l) + G_{l:}^T/\vec{p}_v) \\ \nabla_{\vec{v}_l} d_{JS} &= \frac{1}{2mh}(V - \vec{v}_l)(S_{:l}/\vec{q}_v + S_{:l}/q(\vec{v}_l)) \\ &\quad - \frac{1}{2nh}(U - \vec{v}_l)(G_{:l}/p(\vec{v}_l) + G_{:l}/\vec{q}_u) \end{aligned} \tag{4.6}$$

From Equation 4.6 we see that moving in the direction of the negative computed gradient makes sense intuitively as a rule to bring two distributions closer together. The distribution distance gradient component for a given embedded point x corresponds to summing the vectors from x to each of the other points, with vectors weighted proportional to the average of the ratio of the kernel value between the two points to the pdf evaluated at x and the strength of the kernel value in the total density estimate for that value. In other words, with gradient descent x will tend to move toward the points of the other data set, and away from the points in its own data set, in a weighted manner. However, by also including the term causing the embedding to represent the input matrix well this should help counter the diffusion effect for each data set. We refer to this method as sparse coding with distribution distance regularization (SCDD).

Here we considered one source data set, which could actually be a combination of several source data sets, and one target data set. It is straight-forward to extend the above approach to multiple data sets, e.g., one way is to simply add additional pairwise terms as above for the additional data sets.

4.3.5 Incorporating Target Data Label Information

A common data mining or knowledge discovery task, which is the focus of our experiments in this work, is classification, that is learning a predictive model from the data capable of determining which class a data instance belongs to from its feature representation. Specifically we have a set $\mathscr{C}$ of k classes, $\mathscr{C} = \{1, 2, \ldots, k\}$ and each data instance $\vec{x}_i$ has an (known or unknown) associated class label $y_i \in \mathscr{C}$. The final goal of classification is then to predict the labels of the target data well, generally by estimating $P(y|\vec{x})$ from the labeled data. Even for data where ground truth label information is expensive and time consuming to obtain, usually a small amount of label information can still be obtained. Thus we should be able to leverage this information for knowledge discovery when available.

Furthermore, distribution distance regularization may not always be enough for knowledge discovery. Enforcing small distribution distance for the distribution of the data instances for the two data sets does not guarantee the conditional distributions resulting from the embeddings will be identical. In fact since sparse coding with distribution distance will try to approximate the data well while decreasing the distribution distance, it can end up finding a local non-ideal minimum to the optimization problem 4.7 that misaligns the conditional distributions (e.g., compare synthetic experiments 1 and 2 in Section 4.4.1). In general unless it is certain the distributions of the source and target data sets are closely similar, some ground truth information for the target data is necessary to determine the correct embedding for the data.

We explored several options for incorporating conditional distribution information in the sparse coding formulation including estimating conditional and joint distributions with kernel density estimation. We found a class-based distribution distance estimation approach to work best, where we use the same distribution distance estimate as in the previous section, only calculated between the instances of the same class between the two data sets, for each class. The new objective is given by Equation 4.7.

$$\begin{aligned} \underset{B,W}{\arg\min} \quad & ||X - BW||_F^2 + \alpha \sum_{i=1}^{n} ||\vec{w}_i||_1 + \beta d_{JS}(p_U, q_V) \\ & + \beta_2 (d_{JS}(p_{U1}, q_{V1}) + d_{JS}(p_{U2}, q_{V2})) \end{aligned} \tag{4.7}$$

Here $U1$ denotes those embedded data instances in U that have label 1 and $U2$ those that have label 2 and similarly for $V1$ and $V2$. For simplicity we just described the case of only two classes, but our approach extends easily to multiple classes, simply by using a distribution distance term for each class. Then computing the divergence and gradient for the new component is the same as in the previous section, simply restricted to each class, specifically bringing together the distributions $P(\vec{x}|y = i)$ (that is the probability density of $\vec{x}$ given $y = i$) for each i in $\mathscr{C}$. We refer to this method as sparse coding with distribution distance and class-based distribution distance regularization (SCDDCD).

Importantly, in our implementation we only compute the gradient component for the auxiliary data, and not for the target data, since there are typically very few target data labels. If we updated the labeled target instances as well the few labeled instances would tend to quickly move toward the other data set without influencing the embedding found for the remainder of the target data set - failing to reduce the distribution distance of the true conditional distributions since the unlabeled points would be unaffected.

We can also motivate the incorporation of class-based distribution distance based on theoretical results for knowledge transfer. The general form of such theoretical upper bounds on test (target) error take the form of source (train) error plus distribution distance, based on the marginal distributions [15, 14] when conditional distributions are the same or the conditional distributions [204].

Since our approach enforces soft constraints that require the marginal distributions to be close, and conditional distributions of the data given class to be close, if the classes are roughly balanced, we are in effect enforcing that the conditional distributions of the class label given the data are close, by Bayes' rule. Additionally our approach can be viewed as directly aiming to minimize such theoretical bounds, since first the distribution distance is minimized (and kernel density estimation is consistent [195]) then a classifier is found to minimize training error.

4.3.6 Handling Missing Values: Weighted Loss Sparse Coding

A typical issue that arises in knowledge transfer between different sources of data is that the data have different feature sets, so that only some overlapping set of features is shared in common for different pairs of data sets, and additionally missing values are common. Our approach can easily be adapted to handle such cases by introducing a non-negative $p \times n$ weighting matrix P. This weight matrix is used to weight the reconstruction error described above, so that in the optimization problems more importance is placed on those more heavily weighted entries. This formulation can be used to perform sparse coding for data with missing values, by simply placing a zero in P at each missing entry, and ones elsewhere. The resulting optimization problem, the weighted loss sparse coding problem is given in 4.8, and the extensions for incorporating distribution distance regularization are the same as described previously for unweighted sparse coding.

$$\begin{aligned} \underset{B,W}{\arg\min} \quad & ||P \circ (X - BW)||_F^2 + \alpha \sum_{i=1}^{n} ||\vec{w}_i||_1 \\ \text{s.t.} \quad & ||\vec{b}_i||_2^2 \leq c \quad \forall i = 1, ..., n \end{aligned} \tag{4.8}$$

Here $\circ$ is the Hadamard product, the entry-wise product between two matrices.

4.3.7 Solving the Optimization Problems

The general approach we take to solving the optimization problems presented in the last few sections is one of block coordinate descent, or alternating optimization. We generate a random basis B

of input size r then continually update the weights W to minimize the objective value while holding the basis fixed, followed by updating the basis to minimize the objective value while holding the weights fixed, until convergence.

4.3.7.1 Updating the Basis

We originally tried several different approaches for fitting the basis B given fixed weight matrix W, including a Lagrange dual approach, and the popular Nesterov's method. We found that as the basis size r grew beyond only a very small size a simple projected gradient descent with a line search worked best in terms of efficiency and the embedding found. The gradient of the any of the objective functions we use from Equations 4.2, 4.3, and 4.7 with respect to the basis B is given by Equation 4.9.

$$\nabla_B \text{obj.} = -XW^T + BWW^T \tag{4.9}$$

To update the basis we first compute the negative gradient as the step direction. After computing the new basis by adding the negative gradient, we project it onto the L2 ball constraint for each basis vector, which amounts to scaling each vector to be of max length c. Then a line search is performed where the step size is decreased if the objective value does not decrease. The process is repeated until convergence.

For the weighted loss sparse coding formulation, the gradient computation is similar, e.g., the gradient with respect to B is given in Equation 4.10 and the gradient for W takes a similar form.

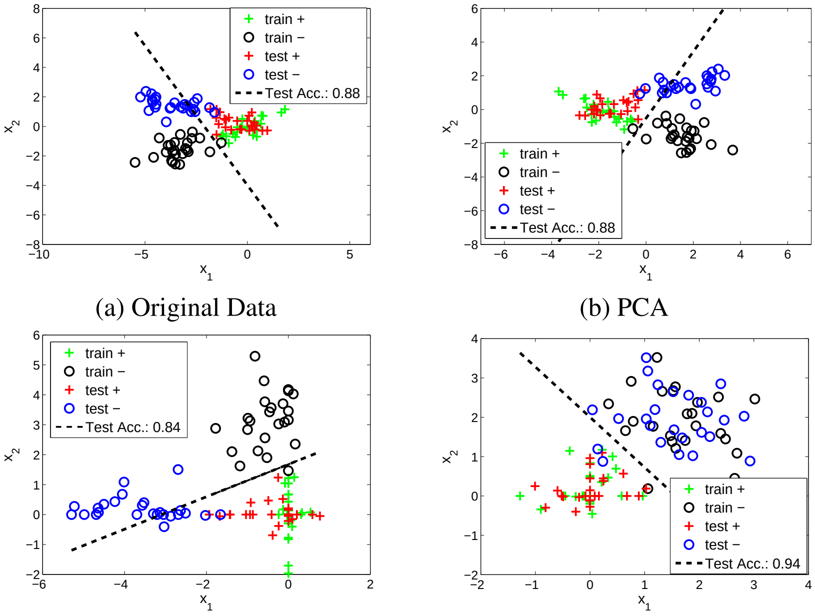
$$\nabla_B \text{obj.} = -(P^2 \circ X)W^T + (P^2 \circ (BW))W^T \tag{4.10}$$

4.3.7.2 Updating the Weights

The same approach for updating the basis is used for updating the weights, except that we use the sub-gradient to incorporate the non-differentiable L1 norm regularization term, and add in the gradient terms for the appropriate distribution distance regularization terms depending on the methods used as described in Sections 4.3.2 and 4.3.5 and Equation 4.6. Additionally no projection is necessary since there are no constraints on the weights. The sub-gradient of the objective functions for the weight matrix W excluding the distribution distance regularization terms is given in Equation 4.11.

$$\nabla_W \text{obj.} = -B^T X + B^T BW + \alpha \, \text{sign}(W) \tag{4.11}$$

Here sign() is the sign function which returns 1 if its input is greater than 0, 0 if equal to 0, and -1 if less than 0.



(a) Original Data

(b) PCA

(c) Sparse Coding

(d) Sparse Coding with Distr. Dist.

Figure 4.1: Comparison of features identified from different embedding methods for the Synthetic data set 1.

4.3.7.3 Convergence

Since the objective with respect to W is nonconvex and not quasiconvex, although the objective function value will not increase, insufficient decrease is potentially an issue with alternating optimization. In practice we check for such a situation by tracking the objective function value. Additionally other similar optimization approaches could easily be used instead to alleviate this issue, e.g., using block coordinate gradient descent instead [188] or regular gradient/pseudo-gradient descent. However we found this coordinate descent approach to be effective in practice. For the hyper-parameter setting most frequently selected in our chemical toxicity experiments (Section 6.5) across all trials the number of iterations to convergence never exceeded 19 and the mean number of iterations was 9.45.

4.4 Experimental Study with Synthetic Data Sets

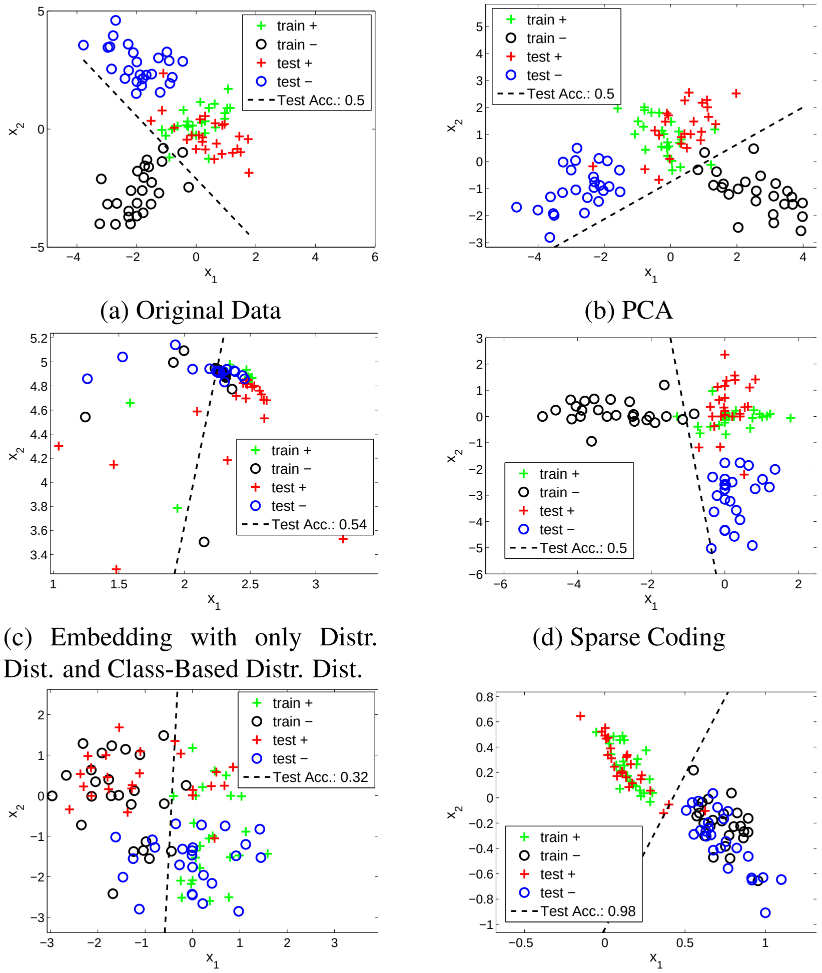
We have implemented our methods in Matlab. All experiments are run on a 178-node cluster where each node contains two Intel Xeon EM64T 3.2 Ghz processors and 4G memory. In order to evaluate the performance of the different feature extraction methods for knowledge transfer, we have created synthetic data sets and collected real-world data sets for chemical toxicity prediction for environmental protection. Below we show our experimental study results with synthetic data sets. We show results on real-world data sets in the next section.

4.4.1 Synthetic Data Experiments

For the synthetic data, we demonstrate the case where the target data set lies mostly in a different cluster than a source data set from which we want to enable knowledge transfer. To simulate this scenario, we generate two data sets, a source, or training, data set, and a target, or testing, data set. To generate data we randomly sample 25 points each from two simple 2D Gaussian distributions, one for each class. The first with mean $(0.6,0)$, the second with mean $(-3,0)$ and both with covariance matrix $\{\{1,0\},\{0,.5\}\}$. We then rotate the source data by some number of degrees θ and the target distribution by the same amount in the opposite direction $-\theta$, using the rotation matrix $R = \{\{\cos(\theta), -\sin(\theta)\}, \{\sin(\theta), \cos(\theta)\}\}$.

Synthetic Experiment 1 For the first experiment, we sample 50 points for each data set as described above and rotate the training data by $\theta = 25$ degrees and the testing by $\theta = -25$ degrees. No labeled test instances are provided for learning the embeddings.

Synthetic Experiment 2 We generate 50 points for each data set using the same set up as described above, except this time rotate the training data by $+55$ degrees around the origin, and the



(a) Original Data

(b) PCA

(c) Embedding with only Distr. Dist. and Class-Based Distr. Dist.

(d) Sparse Coding

(e) Sparse Coding with Distr. Dist.

(f) Sparse Coding with Distr. Dist. and Class-Based Distr. Dist.

Figure 4.2: Comparison of embeddings found for Synthetic Experiment 2 - see text for details.

testing data by -55 degrees, increasing their dissimilarity and hence the difficulty of the knowledge transfer. We then randomly provide only a single label from each class for the testing data to be used in learning the embedding and final classifier.

4.4.1.1 Experiment Protocol

In our experimental study, we did not do an extensive parameter search but simply picked a default value of 1 for the kernel width, the Lasso penalty weight of $\alpha_1 = .2$ (a larger value just tends to compress the points more along the basis directions found), and a heavy weighting of 2000 for the each distribution distance component when included. In the plots showing the results we also plot the support vector machine (SVM) decision boundary found from training on all labeled embedded data points (including the two labeled points of the test data), with default linear SVM parameter $C = 1$.

4.4.1.2 Experiment Results

The results for various embedding approaches are shown in Figure 4.1 and Figure 4.2, with all figures plotted on square plots. For the first experiment, sparse coding identifies the two major subspace clusters, and actual hurts the performance since it essentially assigns each data set to one dimension. The data sets are similar enough however, that just incorporating the distribution distance regularization allows for a very good embedding to be found (Figure 4.1d).

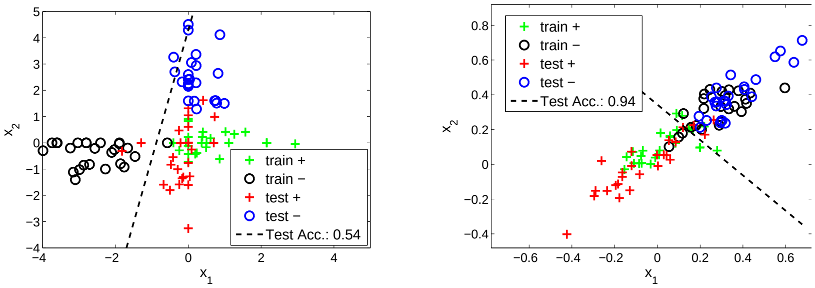
In the second experiment, as before, sparse coding (Figure 4.2d) identifies the two major subspaces or clusters the data belong to, which does not help transfer knowledge in this case since as before each cluster corresponds to a specific data set, so each is assigned its own dimension.

As we expected, just incorporating distribution distance (Figure 4.2e) may not help tremendously in this case, since the nearest alignment of the distributions happens by misaligning the two classes between the two data sets. Incorporating the very few available test labels with distribution distance regularization between the data points of the same classes as described in Section 4.3.5 allows for a very good embedding to be found for transfer learning - the points of each class are grouped together.

In addition we plot the results for PCA in Figure 4.2b. We see that PCA does not move the two distributions close and hence bears poor classification results. To show that the distribution distance minimizing alone is not enough and to demonstrate the utility of sparse coding, we show what happens if just the evenly weighted sum of distribution distance and class distribution distances are minimized with the same gradient procedure, without any sparse coding component, in Figure 4.2c. This results in a poor embedding.

Furthermore we note that this example also illustrates how even restricting the basis size for PCA can easily fail: the principal component found is in the direction $(0.040, -.999)$ which is nearly perpendicular to the best single projection direction for knowledge transfer in this case.

Finally in Figure 4.3 we show a more extreme case, where the same data generation process was used, but the rotation for each data set was increased by 10 degrees. In this example the basic embedding approaches completely fail whereas incorporating the distribution distance still allows high accuracy.



(a) SC

(b) SC with Distr. Dist. and Class Distr. Dist.

Figure 4.3: Comparison of embeddings found for Synthetic Experiment 3

4.5 Knowledge Transfer for Chemical Toxicity Prediction

We evaluated the performance of the aforementioned feature extraction approaches on an environment protection application. The overarching goal of the study is to identify efficient and accurate computational approaches to evaluate toxicity of chemicals and their effects on the environment. Collecting high quality data for chemical toxicity study is an expensive and time consuming process. For example, for the TOXCAST data set described below, the study to obtain the animal toxicity endpoints for about 320 chemicals cost nearly 2 million dollars and took over a year to perform. In reality there are millions of chemicals that need to be evaluated. There is no feasible experimental approach that we could imagine for collecting such data; modeling and computing are indispensable components in the battle for a clean and healthy environment.

The data engineering challenge here is to leverage high quality data collect from the EPA and to build models for chemicals that may deviate from the source distribution. Towards that end, we collected our data sets and designed our experiments as detailed below.

4.5.1 Source Data Set: TOXCAST

Environmental Protection Agency (EPA) has initiated a program called TOXCAST [103] (`http://www.epa.gov/ncct/toxcast/`) in which they have performed a series of *in vitro* tests to collect features for predicting toxicity of chemicals. The TOXCAST data set included results of 309 unique chemicals from pesticides, a serious concern for environmental prediction. A total of 624 different assays, which can be classified into 9 different technologies, were used to predict toxicity of these chemicals. In vivo toxicity responses of most of these chemicals have been compiled in another project by EPA called Toxrefdb (http://epa.gov/ncct/toxrefdb/). This study includes a complete toxicity profile of 474 different chemicals. To construct data set 1, test results from the TOXCAST data set and the chemical descriptors of the chemicals from the software Dragon were used to construct the feature space. The class labels of these chemicals were the toxicity of these chemicals as recorded in the Toxrefdb data set. The endpoint considered was “tumors on mouse liver”. After removing duplicates and compounds with missing or inconclusive endpoint results, the data set consists of 235 chemical compounds.

4.5.2 Target Data Set: CPDB

The Carcinogenic Potency Database (CPDB) (`http://potency.berkeley.edu/`) is a widely used data resource which contains the results for carcinogenic tests on 1547 chemicals. The results in the dataset are reported on rats, mice, hamsters, dogs and nonhuman primates. All the chemicals that proved carcinogenic on mouse livers in the CPDB dataset were selected. These were around 50 in number. Thus, around 50 drugs were randomly picked from FDA approved drugs list and these constituted the non-carcinogenic class. The carcinogenic chemicals selected from the CPDB dataset and the non-carcinogenic chemicals selected from the list of FDA approved drugs together formed the second dataset (Dataset 2) with a total of 112 compounds.

4.5.3 Features Used

For both data sets, we converted the chemical structures to vector-format data by computing chemical descriptors, computed using the DRAGON software (version 5) [187]. The descriptors that we used are a total of 120 atom centered fragments descriptors calculated for each chemical. In our experience (unpublished data), such descriptors are good candidates for chemical activity prediction. We removed any descriptors with variance 0 across both data sets, resulting in a total of 95 features. We then normalized each feature across all data to have mean 0 and variance 1. This set of features represents a common shared set that is readily available and easily obtainable for a given chemical data set. For the source data set, an additional set of features was obtained from the TOXCAST assay experiment results, as mentioned above. After similarly pre-processing these features as well, we obtained 460 additional features for the source data set. For our initial experiments we use only the shared feature representation. In subsequent experiments we analyze and discuss the effect of incorporating the additional features with the weighted loss sparse coding formulation, to see if this common scenario of extra source-specific features could offer some benefit. These experiment details are described in the Experiment Protocol section below.

4.5.4 Distribution Distance Between Source and Test Data

Recent work in chemical-protein interaction prediction demonstrating the effectiveness when enough data is available of using models local to specific regions of the chemical space suggests that distribution shift across the chemical space is a major issue for chemical data and associated prediction tasks [181]. As our source data is from a very specific set of chemicals, and the target set corresponds to a different distinct set of chemicals, as would most additional future prediction tasks, this chemical toxicity prediction task corresponds to a transfer learning scenario. In order to confirm that the two data distributions are different, we measured the KL-Divergence between the source and target data sets (for the shared set of features). Since our kernel-density-estimation-based estimator of KL-Divergence depends on the kernel width chosen and is thus more suitable for comparison than for obtaining an objective estimate, we use a k-nearest-neighbor density estimation approach recently shown to have almost sure convergence [145]. This method depends on selecting a number k of nearest neighbors to use in the estimate. We selected $k = 8$, resulting in the estimated KL-Divergence of 10.54. Varying k from 3 to 47, the minimum KL-Divergence estimate is 8.56 at $k = 13$ and the maximum is 20.02 at $k = 3$ (and the mean is 11.75). The estimated KL-Divergence of the source set with a version of the source set with random zero-mean Gaussian noise added to each feature with standard deviation 0.1 (so that the two data sets are nearly identical) for $k = 8$ is -0.94. Thus the KL-Divergence estimate suggests the difference in distribution between the source and target data.

The characteristics of the data are summarized in Table 4.1.

Table 4.1: Characteristics of the Chemical Toxicity Data Sets

Size source data set	**Size target data set**	**Num. shared features**	**Num. features unique to source data**	**Num. features unique to target data**	**KL-Div.**
235	112	95	460	0	10.54

4.5.5 Experiment Protocol

We use the fully-labeled source data TOXCAST and various increasing numbers of labeled samples from the target data set CPDB, along with all of the unlabeled data from the target set CPDB, to build a model. We then evaluate the accuracy of the model using the unlabeled CPDB data; this is referred to as transductive learning. For each run, we randomly sample the given number of labeled target instances from target data CPDB to be used in the training for the supervised model, and use internal cross-validation with the training data (with the cross-validation evaluation using only the labeled target data selected to be included in the training) to select model parameters for the embedding methods (with the exception of the "KMEns" methods, as described in Section 4.5.5.2). For the case of no labeled target data, for which cross-validation could not be performed, we report results for fixing the parameters to those in the search range resulting in the lowest model complexity, e.g., smallest basis size and largest kernel width. We also tried setting the parameters to the most frequently selected values when labeled target data was present, and obtained similar results, so we only show the former results. To simplify the model selection for the SCDD and SCDDCD methods, we fixed the kernel width h to be equal to the basis size of the embedding and for SCDDCD fixed the regularization parameters for the class distribution distance and data distribution distance to be equal.

For model comparison, we collect accuracy ((TP+TN)/S), sensitivity (TP/(TP+FN)), and specificity (TN/(TN+FP)) for the constructed models, where TP stands for the number of true positives, FP for the number of false positives, TN for the number of true negatives, FN for the number of false negative, and S stands for the total number of instances. All the values reported are collected from the testing data set only and are averaged across 100 experiments with mean and standard deviation reported.

We run a series of experiments to analyze the performance of the proposed feature extraction approach.

4.5.5.1 Experiment 1: Comparing Feature Extraction Methods in a Controlled Setting

For this first set of experiments we use only the shared features for the two data sets. Additionally in order to control for unknown factors for specific feature extraction approaches that may, for example, use arbitrary different base classifiers, or incorporate additional aspects such as manifold learning or other semi-supervised learning methods, we first evaluate representative approaches under the same controlled setting. Since the focus here is on feature extraction, and to have a fair comparison of the different feature extraction methods, we use a fixed classifier (SVM with fixed C and linear kernel) for all methods (including the baseline of no embedding, the original feature space). For each embedding approach, a default linear SVM classifier with parameter $C = 1$ is used on the embedded data to obtain the final predictions. The abbreviation used for each method is given in the following list.

- **SVM** - The SVM classifier trained in the original feature space using both the auxiliary (source) data and the labeled target instances.
- **SVMTG** - The SVM classifier in the original feature space using only the labeled target in-

stances.

- **PCA** - Principal component analysis used on the combined auxiliary (source) and target data.
- **SC** - Sparse coding [152] (Section 4.3.2 Equation 4.2) on the combined data.
- **SCDD** - Sparse coding with just distribution distance regularization (Section 4.3.4, Equation 4.3).
- **SCDDCD** - Sparse coding with both distribution distance regularization and class-based distribution distance regularization (Section 4.3.5, Equation 4.7).

The results for the first set of experiments showing initial comparisons under this same setting are shown in Tables 4.2, 4.3, and 4.4.

4.5.5.2 Experiment 2: Comparing Directly with State-of-the-Art Feature Extraction Transfer Learning Methods

For this set of experiments, we follow the same set-up as for the first set of experiments. We repeat the experiments for competitor state-of-the-art embedding approaches for transfer learning, summarized in the following list.

- **SSTCA** - Semi-supervised transfer component analysis [141].
- **KMEns** - The cross-distribution kernel map ensemble method [217].
- **KMSing** - The non-ensemble version of the cross-distribution kernel map method [217], i.e., this corresponds to using only the final embedding of the KMEns method.

In addition to incorporating MMD distribution distance estimates, the SSTCA method also incorporates semi-supervised learning components in the embedding for enforcing similar data variance and manifold structure in the original and embedded data, plus source label information for finding an embedding useful for classifying data - however this method still does not consider conditional distribution similarity between training and test data. The authors showed in their experiments that their method is largely insensitive to varying the hyper-parameters across a very broad range, so we took the hyper-parameters they found to work best across their experiments and included a range around these hyper-parameters in the grid search with cross-validation in order to select the hyper-parameters in the experiments. We found the performance was slightly worse if we allowed the basis size to be chosen via cross-validation, so instead we report results for a fixed basis size as the authors did in their experiments.

For the KMEns method, we obtained the code from the author's website, and also did not use cross-validation as cluster purity and error decrease is used to automatically determine when to stop clustering [217]. We chose the SMO (SVM) classifier as the base classifier and the ensemble version of their method, as these consistently had the best performance in their experiments. Additionally we also provide comparison with a non-ensemble version (KMSing), to give some idea of the effect of using an ensemble since our method could also be further extended to an ensemble approach. We followed the same approach as the authors and set the number of iterations to 10 - which they found to work best. In their hyper-parameter sensitivity study they found the performance to increase for increasing number of iterations and typically level off at or before 10 iterations, across their experiments. Additionally we tried different approaches for choosing the cluster labels and testing cluster purity, as well as varying the purity threshold from 0.9 to 0.6 and found no improvement over the authors' settings with threshold 0.9, for which we report results.

The results are shown in the Experiment Results section in Table 4.5 and Figure 4.4.

4.5.5.3 Experiment 3: Hyper-Parameter Sensitivity Analysis

For the next set of experiments we analyze the effect of the different components on the performance of our final method (SCDDCD), and also the sensitivity of the performance to the setting of the hyper-parameters, by varying the hyper-parameters. We chose the case of 30 labeled target data instances as the amount to use for the hyper-parameter sensitivity study, and the set-up of multiple experiment runs is the same as previously described. We took the mode of the hyper-parameter values selected across the cross-validation results across all of the experiments. These hyper-parameter values were: a basis size (r) of 16, an L_1 regularization parameter (α) of 0.1 and distribution distance regularization parameters ($\beta = \beta_2$) of 8000. To get the hyper-parameter sensitivity results we then repeated the experiments for num. labeled = 30 by fixing all but one of these hyper-parameter values to the mode values, and varying the other in a range around the found mode value. We report the results in plots in Figure 4.5.

4.5.5.4 Experiment 4: Incorporating Additional Source Data Features

For the final experiment we wanted to analyze the effects of incorporating the additional source data features that are missing in the target data. We compare a direct application of our sparse coding formulation for handling the missing values and also the regression plus singular value decomposition approach [201]. The idea is that incorporating the relationship of the additional potentially useful features with the shared features during the embedding could potentially help identify a better embedding. We label our method for this missing value case SCDDCD-M. To simplify these experiments we fixed the hyper-parameters for our method to the modes of those chosen via cross-validation as described in the previous paragraph on hyper-parameter sensitivity analysis. For the regression plus singular value decomposition approach [201] we obtained the code from the author's website to run on our data. We call this approach SVD when there are no missing values and SVD-M when there are. Essentially the only real difference of this approach from our previously tested PCA approach is that it uses a weighted k-nearest-neighbor classifier as opposed to the fixed SVM classifier after embedding. As the authors did in their experiments we fix the k-nearest neighbor parameter to 50 since it worked best in their experiments and also due to weighting the k-nearest neighbor votes the method is somewhat less sensitive to this value. We vary the embedding basis size and select this basis size via cross-validation. We report accuracy results both for each method without incorporating the additional source features, and each method with incorporating the additional source features, in Table 4.6, again results averaged over multiple trials with the same procedure as used previously.

4.5.6 Experiment Results

The results of the series of experiments are given in the following sections. The results are broken down into four sections corresponding to the four sets of experiments described in the previous Experimental Protocol section.

4.5.6.1 Experiment Results 1: Comparing Feature Extraction Methods in a Controlled Setting

Table 4.2: Mean and std. dev. of accuracy out of 100 runs for each method on EPA data set, for increasing amounts of labeled target data

Num. labeled	**0**	**4**	**10**	**20**	**30**	**40**
SVM	0.536±0.000	0.562±0.028	0.578±0.034	0.605±0.035	0.628±0.054	0.654±0.053
SVMTG	n/a	0.523±0.090	0.648±0.065	0.676±0.065	0.684±0.046	0.725±0.047
PCA [201]	**0.571±0.000**	0.610±0.041	0.643±0.035	0.658±0.033	0.683±0.039	0.684±0.039
SC [152]	**0.571±0.000**	**0.626±0.025**	0.636±0.029	0.659±0.036	0.672±0.037	0.682±0.037
SCDD (Eq. 4.3)	0.545±0.002	0.617±0.036	0.622±0.044	0.656±0.036	0.672±0.041	0.684±0.053
SCDDCD (Eq. 4.7)	0.545±0.002	**0.626±0.076**	**0.685±0.063**	**0.707±0.044**	**0.743±0.048**	**0.754±0.041**

Table 4.2 shows the accuracy results for the experiments, with each row corresponding to a method and each column corresponding to a number of labeled test instance used in training, in increasing order. For the sparse coding methods, these results are also shown in the next results section in the form of a plot of accuracy vs. number of labeled target instances for easier visualization, in Figure 4.4. Table 4.3 and Table 4.4 similarly show results for the specificity and sensitivity, respectively, which provide a measure of the bias of a method toward either reducing Type I errors (false positives) or Type II errors (false negatives).

From the results we see that sparse coding incorporating both distribution distance and the class-based distribution distance components (SCDDCD) in all cases obtains the best accuracy out of all the methods. With only 4 labeled test data instances, the SVM classifier trained using no auxiliary data (SVMTG) does little better than random guessing on average, but the SCDDCD embedding method is able to raise the mean accuracy by an addition of 10 percent. As expected with very little labeled target data, utilizing the available auxiliary data becomes a necessity. As the amount of labeled test data given increases, the performance of SVMTG increases correspondingly, but the SCDDCD method still consistency out-performs the SVMTG method. Even with as many as 40 labeled test instances, utilizing the auxiliary data with the SCDDCD method still offers significant improvement over using only target data (SVMTG). For 4 labeled test instances sparse coding (SC) achieves similar performance to SCDDCD - in this case the benefit of including the test instances could be masked by noise. However, sparse coding improves more slowly with increasing labeled test data and is quickly out-performed by SVMTG. Also just incorporating distribution distance with sparse coding (SCDD) slightly hurts performance for the smaller amounts of labeled test instances, and generally performs about the same as SC. In this case it is clearly not enough to just consider the distribution distance between the data sets. Except for the first set of experiments with the number of labeled test instances equal to 4 for which PCA performed worse than SC, PCA has similar performance to the SC method and is thus also not able to most effectively utilize the auxiliary data in these experiments.

From the specificity and sensitivity results (Tables 4.3 and 4.4) we see that all of the embedding methods that utilize the auxiliary data have a bias toward increased specificity at a cost of decreased sensitivity. However the opposite is true for the method using only the target data, SVMTG. The SCDDCD method however is somewhat more balanced.

Table 4.3: Mean and std. dev. of specificity out of 100 runs for each method on EPA data set

Num. lab.	**4**	**10**	**20**	**30**	**40**
SVM	0.67±0.03	0.68±0.04	0.70±0.05	0.72±0.07	0.73±0.08
SVMTG	0.37±0.40	0.54±0.17	0.64±0.14	0.62±0.11	0.68±0.07
PCA	0.90±0.07	0.90±0.06	0.91±0.06	0.91±0.06	0.91±0.06
SC	**0.95±0.06**	**0.93±0.06**	**0.92±0.06**	**0.92±0.06**	**0.92±0.06**
SCDD	0.90±0.06	0.90±0.070	0.89±0.05	0.91±0.05	0.90±0.05
SCDDCD	0.78±0.18	0.84±0.08	0.86±0.09	0.88±0.07	0.89±0.06

Table 4.4: Mean and std. dev. of sensitivity out of 100 runs for each method on EPA data set

Num. lab.	**4**	**10**	**20**	**30**	**40**
SVM	0.42±0.04	0.44±0.06	0.48±0.08	0.50±0.08	0.54±0.10
SVMTG	**0.73±0.37**	**0.80±0.16**	**0.73±0.11**	**0.78±0.12**	**0.79±0.10**
PCA	0.26±0.15	0.27±0.13	0.28±0.10	0.30±0.09	0.31±0.10
SC	0.21±0.09	0.24±0.10	0.26±0.09	0.27±0.09	0.29±0.09
SCDD	0.23±0.10	0.25±0.11	0.32±0.09	0.32±0.08	0.36±0.11
SCDDCD	0.42±0.19	0.48±0.15	0.49±0.12	0.55±0.12	0.55±0.11

4.5.6.2 Experiment Results 2: Comparing Directly with State-of-the-Art Feature Extraction Transfer Learning Methods

Table 4.5 shows the accuracy results for the second set of experiments - comparison with the two state-of-the-art transfer learning embedding methods SSTCA and KMEns, with the results of our method reproduced for comparison. These results along with the results for our sparse coding methods are also plotted in Figure 4.4 for easier visualization, in the form of accuracy vs. number of labeled target data instances used in training.

Table 4.5: Comparison with state-of-the-art, mean and std. dev. of accuracy out of 100 runs for increasing amounts of labeled target data

Num. labeled	**0**	**4**	**10**	**20**	**30**	**40**
SC [152]	0.571±0.000	**0.626±0.025**	0.636±0.029	0.659±0.036	0.672±0.037	0.682±0.037
SCDD (Eq. 4.3)	0.545±0.002	0.617±0.036	0.622±0.044	0.656±0.036	0.672±0.041	0.684±0.053
SCDDCD (Eq. 4.7)	0.545±0.002	**0.626±0.076**	**0.685±0.063**	**0.707±0.044**	0.743±0.048	0.754±0.041
SSTCA [141]	**0.598±0.000**	0.607±0.016	0.619±0.027	0.608±0.045	0.620±0.043	0.640±0.039
KMSing [217]	n/a	0.546±0.084	0.598±0.090	0.679±0.099	0.720±0.087	0.767±0.078
KMEns [217]	n/a	0.489±0.082	0.588±0.115	0.667±0.092	**0.761±0.087**	**0.791±0.080**

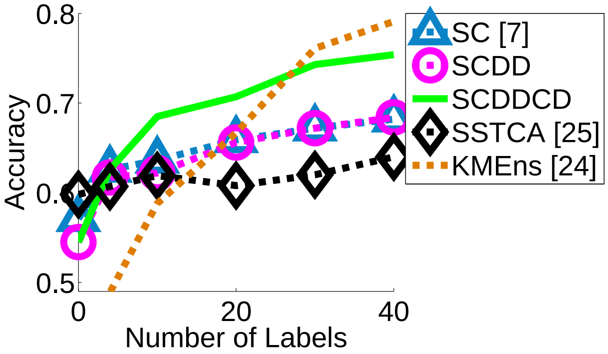
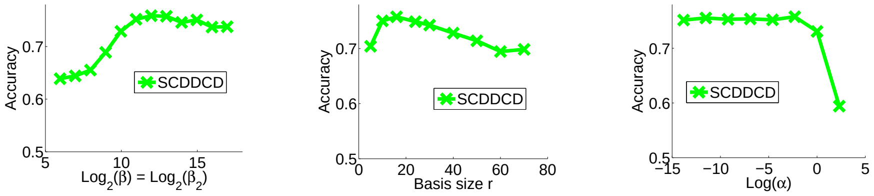


Figure 4.4: Accuracy vs. num. labeled target data instances

In this case, our method of sparse coding incorporating both distribution distance and the class-based distribution distance components (SCDDCD) obtains the best accuracy in comparison to the state-of-the-art methods for the case of small amounts of labeled target data, but as the amount of labeled target data gets larger the kernel map ensemble (KMEns) approach becomes more effective. However, the same kernel mapping approach without using the ensemble (KMSing), i.e., just using the embedding of the final iteration, remains comparable to our method for these increased amounts of labeled target data. We further note that our method might also potentially benefit from an ensemble approach in the same way as the KMEns method, and pose exploring ensemble approaches for our method as a direction for future work. For the SSTCA method, since its performance does not increase as rapidly as the other methods with increasing labeled target data, we suspect that its performance may suffer in part due to failing to consider the effect of the embedding on the conditional distributions as well as relying heavily on the source data in part due to additional components incorporated such as the supervisory component for the source data. On the other hand, the KMEns method does not seem to be able to take full advantage of the auxiliary (source) data, its accuracy is lower at first. We suspect that the different chemical data sets may to some extent lie in different regions of the chemical space so that more labeled target data is necessary to fully identify these regions and the correct cluster structure. Thus we hypothesize that with limited labeled target data such cluster-based approaches may mostly be reinforcing sub-optimal estimations about the class regions until more labeled target data becomes available, making such approaches less effective at fully utilizing the source data in such cases. The SCDDCD method however can still potentially allow knowledge transfer in such scenarios.

4.5.6.3 Experiment Results 3: Hyper-Parameter Sensitivity Analysis

The next set of results show the sensitivity of the SCDDCD method to the various hyper-parameters, the basis size r, the L_1 regularization parameter (α) and the distribution distance regularization parameters $\beta = \beta_2$ (set to the same value). These results are shown in Figure 4.5. The first plot,



(a) Accuracy vs. $\beta = \beta_2$

(b) Accuracy vs. r

(c) Accuracy vs. α

Figure 4.5: Hyper-parameter sensitivity results - accuracy vs. hyper-parameter settings

Figure 4.5a, helps illustrate the importance of including a distribution distance estimation component. As the weight for this regularization component decreases, the accuracy drops. Additionally, the accuracy remains comparably high for a wide range of larger values for the hyper-parameter (note the x-axis is on a log scale). The next plot, Figure 4.5b, shows the sensitivity to the basis size. Here the performance is relatively stable across various basis sizes tested. If the basis size is too small the performance deteriorates, and the accuracy also decreases past a certain point as the basis size grows too large, but the decrease is at a relatively slow rate. Finally Figure 4.5c shows the sensitivity to the L_1 regularization parameter. For this data it seems the performance is relatively insensitive to this parameter as long as it is not too large.

4.5.6.4 Experiment Results 4: Incorporating Additional Source Data Features

Table 4.6 shows the results for the methods incorporating the additional source data features.

Table 4.6: Results when incorporating additional source data features, mean and std. dev. of accuracy out of 100 runs for increasing amounts of labeled target data

Num. lab.	**4**	**10**	**20**	**30**	**40**
SCDDCD	**0.63±0.08**	**0.69±0.06**	**0.701±0.04**	**0.74±0.05**	**0.75±0.04**
SCDDCD-M	0.62±0.07	0.65±0.06	0.67±0.05	0.70±0.04	0.71±0.05
SVD [201]	0.57±0.01	0.58±0.01	0.60±0.02	0.62±0.03	0.64±0.02
SVD-M [201]	0.57±0.01	0.58±0.01	0.63±0.03	0.68±0.02	0.74±0.03

Here incorporating the additional source features actually hurts the performance of our sparse coding method slightly. While the performance of the SVD method improves slightly when incorporating the additional features for the larger amounts of labeled data, we believe this is mostly due to the k-nearest-neighbor algorithm and the nature of the regression. We found that the regressed values for the target data were all very different from the collective set of additional features for the source data, and all much more similar to each other. Thus when embedding the data with the regressed values, the target data is mapped much more closely together, so target data is much closer to target data than source data. Thus when computing the nearest neighbors and weighting their predictions by similarity, the SVD-M method ends up selecting and weighting the target data more highly and performance becomes similar to that of only using the target data (e.g., SVMTG).

We suspect that the additional features do not provide necessary additional information for predicting the label over just using the chemical descriptor (common) features. Therefore trying to find an embedding that also encodes these additional features as well when they are not needed may be difficult and hurt performance for our sparse coding method. We believe this task corresponds to the case of the additional features providing a second view of the data, and that each view itself is potentially sufficient. I.e., we suspect this may be a case where multi-view semi-supervised learning [25] approaches could be helpful. This case is an area of future work, and in particular we believe exploring a potential research direction combining multi-view semi-supervised learning, missing value imputation, and transfer learning could prove effective.

4.6 Conclusion

Data with little to no ground truth information coming from a different distribution motivate us to investigate approaches to leverage the available auxiliary data sources to aid in knowledge discovery. We have explored a feature extraction perspective, starting with the popular sparse coding approach which learns a set of higher order features for the data. After discussing the advantages and limitations of sparse coding for knowledge transfer we have proposed new feature generation algorithms to address those limitations and enable knowledge transfer, and verified the effectiveness of our approach on real and synthetic data. We have evaluated the proposed methods on both synthetic data sets and a real-world data set of chemical toxicity prediction, and found that incorporating both distribution distance estimates and class-based distribution-distance estimates was necessary to improve the sparse coding approach and provide comparable or better performance with state-of-the-art transfer learning methods. This confirmed our hypothesis that finding higher-level features alone is not enough to allow knowledge transfer. In the future we believe our proposed approach could provide a good starting point for addressing the complicated task of knowledge transfer from multiple heterogeneous data sources.

Chapter 5

Related Work on Multi-View Semi-Supervised Learning

This chapter presents a general overview of the related work in multi-view semi-supervised learning. More details and additional related work is presented in subsequent chapters for each specific topic of my thesis.

Here, following common terminology in the machine learning literature, we use the phrase "multi-view semi-supervised learning" to refer to learning methods that specifically exploit in some way the view-specific predictor consensus concept described in the introduction (Chapter 1). It is important to note that there are more general approaches, more commonly referred to as multi-modal data fusion methods, that do match the multi-view learning setting, and which could also be considered unsupervised or semi-supervised as they often use unlabeled data for the model estimation. The key difference is that these do not aim to exploit the main ideas and data characteristics underlying multi-view semi-supervised learning of view function consensus and the related assumptions of limited dependence between views and predictive sufficiency of separate views. Essentially the multi-modal fusion approaches generally make fewer assumptions about the characteristics of the data, which has the advantage of making the algorithms more general, but the disadvantage of failing to exploit these specific characteristics when present. Aside from attempts to estimate the characteristics of the data to determine if assumptions hold, a simple solution in practice is to additionally apply a multi-modal fusion approach as a backup when using multi-view semi-supervised learning approaches. That way if the more specialized multi-view semi-supervised learning approach does not work as well according to model selection approach, a more general multi-modal fusion approach can be substituted, while this ensemble approach is out of the scope of this thesis, it is a direction for future research. Most multi-modal fusion approaches try to find a single shared representation for multiple modes of the data such as text and images. One main approach for multi-modal data fusion is the use of latent probabilistic models for the data [115, 163, 24, 11, 136, 108, 194, 213, 206, 202, 205, 44]; other approaches include multiple kernel learning to combine different view kernels [112, 62, 207, 42, 102], general multi-modal dimensionality reduction techniques [77], feature vector merging [182], and single modality expert output merging [182, 97, 199]. Additionally multi-modal data fusion is a core problem in multi-media data analysis. Atrey *et al.* provide a recent survey on multi-modal data fusion for multi-media data [7]. Also many of the aforementioned multi-modal fusion approaches in the latent probabilistic models category are related in some way to dimensionality reduction techniques that do consider the consensus idea, described below in Section 5.2.1. Although they do not explicitly consider the consensus idea, these probabilistic models implicitly include mapping functions between views, and essentially take shared representations from components of those functions, which is similar to pre-processing and dimensionality reduction techniques using the consensus idea. However, computing a single shared representations as the multi-modal fusion approaches do excludes the potential benefit of applying a multi-view semi-supervised learning algorithm after this pre-processing stage to improve the predictive model estimation, as there are no longer multiple views.

Additionally, multi-view semi-supervised learning is just one of many approaches to semi-supervised learning, resulting from a particular set of assumptions. Different assumptions lead to different semi-supervised learning approaches, for instance the assumption that data lie on a low-dimensional manifold embedded in a high dimensional space corresponds to *manifold learning*. A recent survey on semi-supervised learning approaches has been provided by Zhu [224]; additionally the fairly recent book on semi-supervised learning edited by Chapelle, Schölkopf, and Zien is also informative [37].

As mentioned, methods for multi-view semi-supervised learning generally exploit in some way the idea of predictive agreement on unlabeled data for ideal functions from each view, whether explicitly or implicitly. This is used to reduce the size of the hypothesis spaces and thus reduce the variance of the model estimation. The following is an overview of work on multi-view semi-supervised learning divided into four major categories: pseudo-labeling approaches, which iteratively label unlabeled instances; co-regularization approaches, which incorporate the agreement idea into an optimization problem via constraints or regularization terms; work on active learning, which use the agreement idea to select unlabeled instances for labeling by a human; and extensions to multi-view semi-supervised learning.

5.1 Pseudo-Labeling Approaches

Among the first approaches proposed for multi-view semi-supervised learning were the pseudo-labeling approaches. The algorithms in this category proceed iteratively, and at each iteration labels or soft labels are assigned to some or all of the unlabeled instances, either based on view agreement or confidence of models in individual views. These pseudo-labeled instances are then used as labeled training instances for some or all of the views, thereby increasing the training set size, the models are re-trained with the new pseudo-labeled data, and the process repeats iteratively [53, 54, 21, 25, 164, 129, 58, 1, 133, 9, 8, 191, 47, 193, 137, 30, 22, 73, 221, 222, 28]. The archetypal and one of the first proposed multi-view pseudo-labeling algorithms is co-training [25]. The co-training algorithm involves training predictors for each view with the initial labeled data. Then, iteratively, the predictors in each view each label some number of unlabeled instances, and those instances are added as labeled instances to the training set for the other views. Typically the instances selected are the ones predicted with the highest confidence in terms of probability;

e.g., for a linear model, this corresponds to the instances furthest from the decision hyperplane. Much subsequent work following [25] has been on understanding co-training's effectiveness and establishing theoretical guarantees.

5.2 Co-Regularization Approaches

Instead of somehow pseudo-labeling the unlabeled data, co-regularization methods use semi-supervised agreement-based regularization, that is penalizing the disagreement of different view functions on the unlabeled data instances in the model estimation optimization problem [50, 111, 183, 178, 29, 65, 186, 158, 184, 210, 209, 179, 160, 159]. For example, the sum of the square differences between the unlabeled data projected onto the linear prediction hyperplane direction in different views is the most commonly used penalty [111, 178, 29, 158, 184, 210, 209, 179, 160, 159]. Co-regularization was adopted as an alternative to co-training-style methods, due to limitations of such approaches [178]. In particular, it was pointed out that co-training is a greedy maximizer that can get stuck in poor solutions by not implicitly considering multiple solutions as co-regularization does, and unlike co-regularization cannot be tuned to adjust the influence of different components [178]. Additionally simple test cases were shown in which co-training fails consistently due to its greedy nature but co-regularization succeeds [178].

5.2.1 Clustering and Dimensionality Reduction

Also closely related to co-regularization are pre-processing and clustering methods that use the agreement idea [105, 106, 61, 220, 218, 5, 40, 23, 123, 60, 93, 48]. Typically these methods reduce the dimensionality of the data by selecting those sets of basis vectors (or functions) in each view for which the projected (or evaluated) unlabeled data are highly correlated (agree), the relation being that underlying functions in each view that agree on unlabeled data will be combinations of the correlated (agreement) directions (or functions). These approaches in essence follow the same idea as co-regularization, except are usually unsupervised. Whereas co-regularization finds a single best function in each view for which the function predictions agree (are correlated) across views and also match the labeled data, the approaches in this category typically find multiple functions that agree and use these to form a basis for future learning tasks. This is usually captured via linear models in some feature spaces, so that functions correspond to vectors and agreement to correlation of projected data onto those vectors in different views. A common approach is to use canonical correlation analysis [92, 82]. Canonical correlation analysis finds a set of corresponding vectors for each view that are maximally correlated. Another approach is to use agreement in terms of graph cuts, e.g., finding a normalized cut that works well for multiple graph views of the data [218].

5.3 Active Learning Approaches

Active learning is a form of semi-supervised learning where the algorithm is sequentially allowed to choose the unlabeled data instances to obtain ground-truth labels for [170]. Methods in this category use the agreement idea for multiple views to help determine which unlabeled instances are most important to label first [131, 132, 130, 111, 135, 134, 79, 192]. The common idea utilized is to choose the unlabeled data instances which the models from different views disagree in their predictions the most, and the approach has been shown to perform better than single-view active learning approaches both in theory and practice [134, 79, 192].

Recently, the idea of actively obtaining missing views for a selected instance based on an estimation of the information it would provide under a specific probabilistic model was proposed [209].

5.4 Extensions, Including Missing View Considerations

A variety of extensions to multi-view semi-supervised learning approaches have been proposed. To allow the ideas of multi-view semi-supervised learning to be applied in cases where only as single view of data is available, different methods have been proposed including different ways of splitting the features of one view into multiple sets [137, 28, 43], using diverse predictors with the same data in place of different views [191, 73, 221, 222], using clustering to generate other views [155], and using a pre-existing view generation function [2]. Additionally, for the case of partially available view information, i.e., additional views available in some cases, Yu *et al.* proposed to marginalize out missing views in a Gaussian process model [209].

Additionally there has been subsequent work extending multi-view semi-supervised learning approaches to special cases such as structured non-identical outputs [68], transfer learning scenarios [212], multi-task learning [87], cases where there is no correspondence between views with transfer learning assumptions [83], and handling erroneous or noisy data resulting in view-disagreement [47, 46]. Additional work has been proposed combining multi-view semi-supervised learning with other semi-supervised learning approaches such as the transductive SVM [118] and manifold regularization [178, 179].

When we say that one view is “missing” in multi-view semi-supervised learning for a data instance, we mean that all the feature values in that view are not recorded. In this sense we are discussing structured missing values, which is dramatically different from handling random missing feature values, having differing assumptions and objectives. A recent thesis discusses machine learning with missing feature values [125].

Chapter 6

View Completion via Feature Generation

6.1 Introduction

With the fast development of cost-effective data collection methods in imaging, the health care industry, the web, social networks, and sensor networks, data from multi-sensory devices, i.e., multi-view data, become ubiquitous. In the multi-view data setting, information collected from each sensory device is a "view". Often individual views are sufficient for prediction tasks given enough labeled data. Multi-view semi-supervised learning methods aim to take advantage of large amounts of unlabeled data by enforcing view-specific predictor consensus on the unlabeled data. Multi-view semi-supervised learning (MVSSL) has been shown to be effective in a variety of applications including text mining [25, 209, 210], image annotation [65, 186], and chemical classification [53, 54].

A key limitation that restricts the wide application of existing MVSSL approaches to a wide range of real-world data sets is that those approaches require the completeness of the data set. Complete multi-view data, however, are rare and a much more common scenario is *incomplete* multi-view data where views may only be available for a subset of samples. For example, for prediction tasks involving chemicals, molecular structure features based on chemical graphs (view 1) can be readily obtained, but obtaining the chemical bioactivity data (e.g., chemical-protein interaction profiles) for a set of proteins (view 2) can be costly and time-consuming. As another example in medical diagnostics [209] where additional views correspond to expensive tests like MRI imaging, information from such views are subject to opportunity. Yet another example of incomplete views comes from webpage classification where incoming link text features provide a convenient second view [25]. Such information may not be always available for new webpages since it requires time and resources to collect.

This case of MVSSL with various amounts of incomplete view data, which we call *multi-view semi-supervised learning with partially observed views*, is commonly encountered in many real-world applications but has barely been addressed in the data mining and machine learning literature. The first method to claim credit for considering missing views in the MVSSL setting is the Gaussian process co-regularization (GPCR) approach [209]. Under this approach missing views are handled in a Bayesian framework by integrating out the missing view function values. Though it has achieved promising preliminary results, GPCR has several limitations. First, GPCR is built on a particular MVSSL framework, co-regularization, which is not always the best or most appropriate for a given application. Second, GPCR essentially ignores those unlabeled data points without a second view, limiting its applicability to cases with little-to-no second view data. A closely related direction to handling partially observed views is the study of MVSSL methods when there is no second view data [28, 43, 73, 137, 155, 191, 221, 222]. The most recent, state-of-the-art method in this category is pseudo multi-view co-training (PMC) [43], which is also the first in this category to explicitly consider conditions for the success of MVSSL algorithms. This method works by choosing a feature partition at each iteration in order to artificially derive two views. However all of the methods in this category completely ignore additional view data and hence cannot take advantage of such data when available. Furthermore, whereas appropriate real data inherently satisfies the desired conditions, with artificially constructed view data the satisfaction of such conditions can only be approximately estimated. In addition feature-splitting approaches like PMC will fail when all or most of the features in a view are needed for a predictor to achieve high-performance. Furthermore the transformation needed to result in two sufficient views may be more complex than a simple partition. Additionally these methods are also often tied to a particular MVSSL algorithm, e.g., PMC is closely integrated with the co-training algorithm and it is not clear if it could even be applied to a co-regularization algorithm, for example.

We aim to extend MVSSL to handle cases with partially observed views. In our study, we assume there is one view that is present in all data. The rest of the views may only be partially observed. Although this assumption may seem restrictive at first glance, it is quite generic in real-world examples. For example, in the chemical activity prediction example that we cited previously, features computed from chemical structures are always available (since those features are computed). As another example, in the webpage classification example, for every webpage, features computed from the content of the page itself (e.g., the bag-of-word representation of the page) are always available but the incoming link information may be missing.

To solve the problem, we have designed a unified approach, CoNet, which uses a feature-generation network for learning a mapping to fill in missing views. A motivating observation is that feature generation approaches are widely used to improve performance for standard supervised learning tasks, therefore we might expect a feature generation approach to also be helpful in the MVSSL setting. However, a key difference is that the goal for the generated data is different - in this case the generated view data should have properties making it useful for MVSSL, that is in conjunction with the original data. We start with the idea of using random nonlinear feature generation functions to generate new view data. Random nonlinear features allow variability in the generated view: the data points are "scattered" to some extent so that labeled data points may be closest to different unlabeled data points in the generated view. This helps ensure that conditions sufficient for the success of MVSSL algorithms are met, in particular the "expansion" condition [9] requiring that there is some chance that some unlabeled data instances can be labeled with "confidence" in one view but not the other. By incorporating these features together in a network structure, we can then fine-tune the collective set of feature generation functions to further ensure that the conditions for MVSSL algorithms are met, namely label consistency and view variability, and additionally that the generated features are consistent with any partial view data available.

This results in a very natural approach to generating features for MVSSL. Our approach has the key advantages of operating as a pre-processing step which allows the subsequent application of the most application-appropriate MVSSL algorithm to the completed data, efficient out-of-sample extension, and the ability to make use of additional view data when available. Our comprehensive experimental study demonstrates the utility of the CoNet method as compared to the state-of-the-art MVSSL methods GPCR and PMC.

6.2 Related Work

Multi-view semi-supervised learning has attracted significant research interest in recent years [47, 192, 193]. Methods for multi-view semi-supervised learning generally exploit in some way the idea of predictive agreement on unlabeled data for ideal functions from each view, whether explicitly or implicitly. MVSSL approaches can be roughly divided into three major categories: pseudo-labeling approaches, which iteratively label unlabeled instances [25]; co-regularization approaches, which incorporate the agreement idea into an optimization problem via constraints or regularization terms [65, 178, 218]; and active learning approaches, which use the agreement idea to select unlabeled instances for labeling by a human [134].

View Generating Functions. Theoretical results were established and verified in experiments showing that improved generalization error could be achieved by using pre-defined view-generating functions mapping one view to another to fill in missing views and effectively increasing the training set size for each view [2]. The limitation of this work is that the existence of “natural” view mapping functions (e.g., translators for cross language text categorization) is assumed. Such natural view mapping functions do not exist for many applications.

View Splitting for MVSLL. One extreme case of partially observed views is the case of having only a single view. There are several approaches that aim to extend the ideas of multi-view semi-supervised learning to single view learning, following a general idea of splitting the features of one view into multiple sets [28, 137]. Recently, one such approach was proposed in which features are split into two views according to criteria that included satisfying the expansion condition for co-training [9], by finding a split such that some unlabeled instances are labeled with confidence in one view but not the other given the current view models [43]. However feature splitting approaches rely on the assumption that the split sets of features will be sufficient for learning. This means they cannot be applied to data where most of the features are needed for learning a good predictor, for example, see Figure 6.3; splitting the features in this case would result in overlapping classes in each new view. Secondly, even if useful redundancy is present in a single view, this redundancy may be in the form of arbitrary linear combinations of the features or more complex functions of the features, as opposed to the more restricted mapping of feature partitioning.

Additionally for the single view case, several approaches based on using diverse predictors have been proposed [73, 191, 221, 222]. However, in addition to restricting the choice of algorithms, these approaches do not have a clear way for choosing which predictors to use. For instance in one approach co-training was performed using k-nearest-neighbor regressors with different distance metrics and/or values of k in place of different views, but mixed results were obtained depending on the arbitrary choices [222], and further this limits what methods can be used and diversity may come at the cost of worse performance for the individual predictors used.

It is also worth mentioning that many latent model, multi-modal fusion methods [44, 108, 136] might also be used to estimate missing views, but these approaches have the goal of combining different views into one as opposed to exploiting the variability in distinct views, and as such they do not consider the subsequent application of MVSSL algorithms.

When we say that one view is “missing” in MVSSL for a data instance, we mean that all the feature values in that view are not recorded. In this sense we are discussing structured missing values, which is dramatically different from handling random missing values [125].

6.3 Background

6.3.1 Notation and Setting

We use the following notations throughout the rest of the chapter. We use lowercase letters to represent scalar values, lower-case letters with an arrow to represent vectors (e.g., $\vec{x}$), uppercase letters to represent matrices, and uppercase calligraphic letters to represent sets. We use $||\vec{a}||_p = (\sum_{i=1}^{k} |a_i|^p)^{1/p}$ to denote the L_p norm of a k-dimensional vector $\vec{a}$. Unless stated otherwise, all vectors are column vectors.

In MVSSL with partially observed views, we have two sets of data. One set is a set of n labeled samples, e.g., $\{(\vec{x}_1^1, \vec{x}_1^2, \ldots, \vec{x}_1^V, y_1),$
$\ldots, (\vec{x}_n^1, \vec{x}_n^2, \ldots, \vec{x}_n^V, y_n)\} \in \mathscr{X}^1 \times \mathscr{X}^2 \times \mathscr{Y}$. Additionally we have a set of m unlabeled data points from the same spaces,
$\{(\vec{x}_{n+1}^1, \vec{x}_{n+1}^2, \ldots, \vec{x}_{n+1}^V), \ldots, (\vec{x}_{n+m}^1, \vec{x}_{n+m}^2), \ldots, \vec{x}_{n+m}^V)\}$
$\in \mathscr{X}^1 \times \mathscr{X}^2$. V is the number of views.

For simplicity we will restrict further discussion to the case of $V = 2$ views, though all the proposed methods can be extended to more than two views. We take $\mathscr{X}^1$ to be $\mathbb{R}^{p_1}$ and $\mathscr{X}^2$ to be $\mathbb{R}^{p_2}$ for some positive integers p_1 and p_2, i.e., view 1 has p_1 features and view 2 p_2 features. We also restrict the label space to $\mathscr{Y} = \{-1, 1\}$ since all of the applications discussed and tested in the experiments deal with binary classification. Additionally we assume that one view is always present but the other is potentially missing in some samples, for two reasons. First, this is the scenario encountered in all data sets used in the proposed experiments, and is the most commonly encountered one. Second, solving this case immediately provides a solution to the case of additional views that may also have missing view cases, simply by computing pair-wise feature generation functions for filling in each view.

6.3.2 View Expansion in Multi-view Learning

There has been much research on the conditions for which MVSSL may lead to improved predictive performance. There are at least four directions. First originally the condition of conditional independence of views given the class label was proposed as the required condition for the success of co-training [25]. Second for the co-regularization method [210] showed how the co-regularization approach was equivalent to using a special data-dependent kernel for the support vector machine. [179] simplified the theoretical analysis and established similar bounds as [158] and further proposed a co-regularized alternative to manifold regularization [12] that offered significant empirical improvement in their experiments. Following this direction [209] designed a Bayesian MVSSL algorithm that handles missing views.

We follow a different direction of view expansion. It has been shown that an "expansion" condition, weaker than conditional independence, is sufficient for MVSSL to improve over single view learning [9]. This condition requires that there exist some instances whose labels are not confidently[1] known in one view but are confidently known in the other view, so that labels could be propagated iteratively between views. One illustrative way of thinking about this is is with the following example with two data views. Suppose an unlabeled instance $\vec{x}^1$ in view 1 is in a region in which a given predictive model is confident corresponds to label y, e.g., due to being close to many y-labeled instances in that view. It may be reasonable to assume with confidence that the label of $\vec{x}^1$ is also y. Then the expansion condition would require that the same unlabeled instance, $(\vec{x}^1, \vec{x}^2)$ not be in such a confident region when restricted to the second view, $\vec{x}^2$ in view 2, at least for some such $(\vec{x}^1, \vec{x}^2)$ in the unlabeled data. For example, $\vec{x}^2$ may only be near other unlabeled instances in view 2. If this condition always holds as confident labels are propagated between views, than all of the instances can be labeled. This example is illustrated in Figure 6.1, where the solid rectangle corresponds to the positive class and the dotted box shows a possible "expanded" region for the

[1] In the theoretical results of the cited paper "confident" means "with probability one" i.e., absolute certainty. The authors consider particular scenarios where certain regions of the input space can be labeled with absolute certainty. In practice this is relaxed to mean "relative confidence" for the specific model being used, for example, if a linear model is used the unlabeled instances whose labels are considered to be the most confidently known are usually taken as those farthest from the hyperplane defined by the linear model.

location of the corresponding view 2 point. This potential shuffling means that labeled points can end up near different unlabeled points in the second view and therefore label confidence (based on proximity) can be transferred to the unlabeled points.

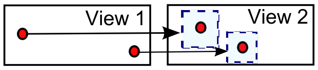


Figure 6.1: An Example Illustrating View Expansion.

This condition motivates the idea proposed here of using the distances between the *profiles* of the data in each view for determining if pairs of views provide sufficiently complementary information, when evaluating candidate values for filling in missing views. Here "profile" refers to a vector capturing the relationship between a data instance $\vec{x}^j$ in view j and all of the unlabeled data in that view, $\vec{x}^j_{n+1}, \ldots, \vec{x}^j_{n+m}$. Specifically here the profile vector $\vec{v}^j$ in view j of distances between $\vec{x}^j$ and each $\vec{x}^j_i$ is given by $\vec{v}^j_i = d(\vec{x}^j, \vec{x}^j_{n+i})$ for $i = 1, \ldots, m$ for a distance function d. An additional motivation for this idea comes from theoretical analysis for co-regularization [179]. In providing a generalization error bound, Sindhwani and Rosenberg also found that the key factor that reduced the bound was a sum of distances between the profiles of the labeled data in each view, with the profiles calculated using a kernel function [179]. The greater these differences in profiles between the views are, the greater the bound on generalization error is reduced.

This motivating difference in profiles idea is incorporated into the proposed approach through a term in the objective function for a feature generation mapping that encourages the sum of squared profile differences $\sum_i \hat{d}(\vec{v}^1_i, \vec{v}^2_i)^2$ to be large, where $\vec{v}^2$ is the profile in the second view which may be generate and $\hat{d}$ is a distance function, potentially different from d. We call this "contrasting view regularization" and this term is described in Section 6.4.4.

6.4 Methodology

6.4.1 CoNet Overview

The main idea behind our approach is to use random nonlinear feature functions to introduce variability in generated views, and to fine-tune these functions to match sufficient conditions for the success of multi-view semi-supervised learning methods and to be consistent with available view 2 data. Matching the available view 2 data also helps to ensure the generated second view is useful for classifying the data. To generate random nonlinear feature functions, we generate random projection directions by iteratively sampling a vector $\vec{w}$ from a p_1-dimensional spherical Gaussian and then normalizing $\vec{w}$ to have length 1. We than choose an initial offset uniformly at random in the range of the values taken by the projected data (both labeled and unlabeled). A sigmoid transfer function, $f(x) = 1/(1+\exp(-x))$ is then applied to introduce nonlinearity.

In order to allow easy fine-tuning of the feature functions, we group functions together into a multi-layered network, i.e., our approach fits naturally into a neural network framework. The final layer is the feature output layer of the network, and each feature function shares all lower layers to allow easier fine-tuning. Each layer is initially generated using the random projection procedure as described above. In our experiments we take the approach of using a single hidden layer followed by the feature output layer, as using a large enough number of hidden nodes can allow sufficient expressivity [49].

In addition we consider the recent advancement from the side of neural networks and explore the initialization strategy of deep belief networks - pre-training the network as a generative model using contrastive divergence [89]. This alternative for initializing the feature generation network potentially provides better performance and stability as it may capture the data manifold and prevent overfitting - identifying an accurate lower-dimensional feature representation for the data could facilitate the feature generation network learning.

Subsequently the first condition to ensure through fine-tuning is consistency with available labeled data, which we achieve by adding an additional output node to the network and using a typical loss function for this output node in an overall objective function for the network. Another term is added to the objective function penalizing the distance between generated view 2 instances and actual view 2 instances when available. Finally, although using random nonlinear features can already help to shuffle the distances between labeled and unlabeled points, we add a "contrasting view regularization" (Section 6.3.2) term to the objective to help ensure this characteristic. Details are given in the following sub-sections.

6.4.2 Proposed Feature Generation Method

A neural-network model is proposed for the feature generation network, mapping one view to another. The general model is depicted in Figure 6.2, which shows a particular network with three input features in view 1, three output features in view 2, and one hidden layer of three units.

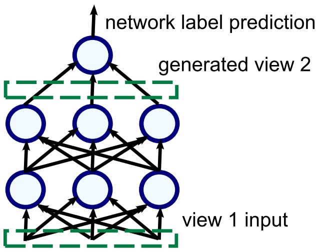


Figure 6.2: Example feature generation network model, where inputs are entered at the bottom and computations propagate through to the top.

An input $\vec{x}^1$ from view 1 is presented to the network, each set of values is transformed by a linear function at each node and passed through a nonlinear transformation $f()$ to get the output of the node, here we use the sigmoid transformation $f(a) = 1/(1+\exp(-a))$. Thus the vector of outputs for a layer j is given by $\vec{f}_j \triangleq \vec{f}(W_j \vec{f}_{j-1} + \vec{b}_j)$ where W_j and $\vec{b}_j$ corresponds to the weight matrix and bias vector for the j^{th} layer of the network, respectively, $\vec{f}_0 \triangleq \vec{x}^1$ for $j = 1, \ldots, K$ where K is the number of layers in the network. The generated feature view, which corresponds to the second view and also must have the same number of features as the second view if available, here corresponds to the output of the second-to-last set of nodes in the network, counting from the bottom. In order to also incorporate good performance on the labeled training data, the network's final output is the predicted label.

The weights and biases are then learned from the available data by attempting to find a local minimum of an objective function. In its most basic form, corresponding to a basic feature generation, or neural, network, the objective function is just the sum of a loss term approximating misclassification error. The basic objective function is given by Equation 6.1, where $\vec{f}_{j,i}$ is the output of the j^{th} layer on an input to the network of $\vec{x}_i^1$.

$$\text{argmin}_{W_j,\vec{b}_j,\forall j} \quad \frac{1}{n}\sum_{i=1}^{n} \log(1+\exp(-y_i f_{K,i})) \tag{6.1}$$

Since the objective function and all transfer functions are differentiable, gradients are straightforward to compute using the chain rule which results in backpropagation with the network structure. A gradient descent approach is then used to find a local solution.

Once the weights and biases are learned from the data, the model can be applied to each instance missing another view, to generate the missing view for that instance. To ensure generated view data is on the same scale as the available view 2 data, we first generate all view 2 data instances, normalize the data, and then (optionally) fill in the available real view 2 data. Afterwards, any desired multi-view semi-supervised learning algorithm can be applied to the completed data.

6.4.3 Incorporating Available Partial View Data

When another sufficient and contrasting view is known to exist, and is present in some cases, ideally the training for the feature generation model should take advantage of this available second view data, to help find a better feature generation function and ensure classification sufficiency of the generated view 2 data. The feature generation model should be biased toward a model that generates values close to the true second view values. This is easily accomplished in the proposed feature generation network model by incorporating an additional penalty term in the objective function. The penalty term is the sum of the square differences between the generated view 2 feature output and the true view 2 feature vector for an instance. Let $\mathscr{P}$ denote the index set of instances for which the second view is present, and $l = |\mathscr{P}|$. Then the basic objective function including available second view data is given by Equation 6.2, where $\vec{f}_{j,i}$ is the output of the j^{th} layer on an input to the network of $\vec{x}_i^1$ for i in a given index set and $j = 1, \ldots, K$, where λ_1 controls a trade-off between fitting the labeled data well and fitting the available second view data well.

$$\begin{aligned} \text{argmin.}_{W_j, \vec{b}_j, \forall j} \quad & \tfrac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i f_{K,i})) \\ & + \tfrac{\lambda_1}{l} \sum_{i \in \mathscr{P}} ||\vec{f}_{K-1,i} - \vec{x}_i^2||_2^2 \end{aligned} \tag{6.2}$$

The new term is differentiable so standard gradient descent approaches are still applicable, and gradient computations are accomplished succinctly with basic matrix operations.

6.4.4 Biasing the Model for Multi-View Semi-Supervised Learning

In order to incorporate the aforementioned differing profile idea in estimating the neural network model, an additional term is added to the objective function of Equation 6.2, given in Equation 6.3. This term biases the learning, forcing the generated view to differ more in its instances' distances to unlabeled data for larger values of the regularization parameter λ_2.

$$-\tfrac{\lambda_2}{nmp_2} \sum_{i=1}^{n} \sum_{j=n+1}^{n+m} (||\vec{x}_i^1 - \vec{x}_j^1||_2^2 - ||\vec{f}_{K-1,i} - \vec{f}_{K-1,j}||_2^2)^2 \tag{6.3}$$

Again this term fits within the backpropagation framework and allows computation with basic matrix operations.

Additionally, for huge amounts of unlabeled data a stochastic gradient approach can be used in estimating the unlabeled data profile distances - a sample of the unlabeled data in such cases could be used to estimate the difference in profiles, and thus a random sample could be taken at each gradient update.

The basic training and testing procedures for multi-view semi-supervised learning approaches combined with the proposed feature generation approach are given by Algorithms 1 and 2, respectively.

```
**Algorithm 1** Training with the Feature Generation Network
**Input:** A set of data $\mathscr{S}$ containing (view 1, view 2, label) triplets, in which view 2 and labels may be missing for a given instance, initial weights and offsets $W_j, \vec{b}_j, \forall j$, a multi-view semi-supervised learning algorithm $A$ which outputs a predictive function $f_A(\mathscr{S}) : \mathscr{X}^1 \times \mathscr{X}^2 \to \mathscr{Y}$ given complete training data. Additional parameters for the feature generation network, $\lambda_1$, $\lambda_2$, number of backpropagation iterations $T$, and whether or not to use use only the generated view 2 data.
**Output:** Final weights and biases for the network $W_j, \vec{b}_j, \forall j$, and the trained predictor $f_A$.
    -Use $T$ iterations of gradient descent to find an approximate local solution to Equation 6.2 with Equation 6.3 added to the objective.
    -Use the learned network ($W_j, \vec{b}_j, \forall j$) from the previous step to generate view 3 for all instances in $\mathscr{S}$. Normalize the generated view 3 data.
    -Fill in any missing view 2 instances of $\mathscr{S}$ with those from the previous step, the generated view 3; optionally replace non-missing view 2 instances with the generated ones as well. Denote the completed data $\hat{\mathscr{S}}$.
    -Apply algorithm $A$ to the completed multi-view semi-supervised data $\hat{\mathscr{S}}$ to obtain $f_A$.
```

```
**Algorithm 2** Testing using the Feature Generation Network
**Input:** A set of data $\mathscr{R}$ containing (view 1, view 2) pairs, in which view 2 and may be missing for a given instance, a trained feature generation network ($W_j, \vec{b}_j, \forall j$), and a trained predictive function $f : \mathscr{X}^1 \times \mathscr{X}^2 \to \mathscr{Y}$, and whether or not to use use only the generated view 2 data.
**Output:** Predictions $y \in \mathscr{Y}$ for each instance of $R$.
    -Use the trained network ($W_j, \vec{b}_j, \forall j$) to fill in any missing view instances of $\mathscr{R}$ and optionally replace the available second view data; denote the completed data $\hat{\mathscr{R}}$.
    -Apply $f$ to each instance in $\hat{\mathscr{R}}$ to obtain the predicted $y$ for that instance.
```

6.4.5 Connections to Modern Deep Network Approaches

The recent resurgence in interest in neural networks in the machine learning and data mining communities is the result of different interpretations of / assumptions about the networks; the models along with these new interpretations/assumptions are often referred to as "deep belief networks" due to a different generative probabilistic (i.e., belief) perspective being assigned to the multi-layer networks [64, 71, 88, 90, 146, 153, 162]. In general most modern approaches keep the same layered structures, and in terms of predictions and network outputs, in general the same feed-forward approach is used to generate layer and label outputs. Additionally backpropagation is commonly still used to fit the net to the data after pre-training. The key difference of the modern approaches are the assumptions of the underlying probabilistic models which can result in different pre-training strategies [64], for example, using layer-wise contrastive divergence [88] to pre-train networks layer-by-layer with unlabeled data. A key practical difference between past neural network methods and modern ones is in how the networks are pre-trained or initialized. Also, even standard neural network methods that do not use pre-training and just use the backpropagation have still been used recently to achieve state of the art performance [197]. Although our approach is for generating an additional, complementary set of features as opposed to replacing an existing one, this view generation problem could offer a new direction for work on deep network architectures, and our regularization terms could be viewed as additional ways to prevent overfitting with such architectures. An important component of our work is testing the combination of the deep belief network approach with our method, through pre-training the feature generation network.

6.5 Experimental Study

We test our method with synthetic and real data. For each experiment we report results in terms of test error if the data is balanced, and also Matthews Correlation Coefficient (MCC) and F1 Score if the data is unbalanced. Let tp denote the number of true positive predictions, fp the number of false positives, fn false negatives, and tn true negatives.

- Test error is given by: $\frac{fp+fn}{tp+tn+fp+fn}$.
- MCC is given by: $\frac{(tp)(tn)-(fp)(fn)}{\sqrt{(tp+fp)(tp+fn)(tn+fp)(tn+fn)}}$.
- F1 Score is given by: $\frac{2tp}{2tp+fn+fp}$.

Note that MCC and F1 score attain their best values at 1, and test error at 0, and MCC takes into account both false positive and false negative rates whereas F1 score does not take into account the false negative rate.

We compare our method CoNet with two state-of-the-art methods. The first method has the claim of being the first approach to handle missing view data in the MVSSL setting, gaussian process co-regularization (GPCR) [209]. The second is the most recent approach to applying

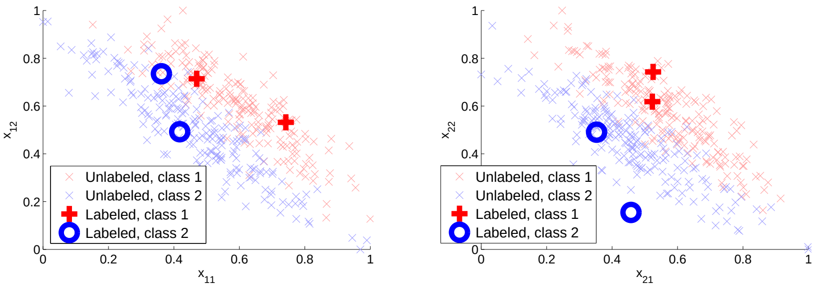
MVSSL to the single view case (completely missing second view - i.e., whatever second view data is available is ignored) and reported state-of-the-art results - pseudo multi-view co-training (PMC) [43]. We obtained the code for PMC from the authors, and used the "Gaussian Processes for Machine Learning Toolbox" version 3.1 [156] to implement GPCR. Note that for our experiments in general we cannot apply basic multi-view semi-supervised learning methods not designed to handle missing view data, such as co-training, as baselines. This is because view 2 is missing at random and may not be present even in the labeled data, or if it is it may only be present for one class due to the often highly imbalanced nature of the data. Additionally we compare with the baseline of only using the single omnipresent (first) view, using a Gaussian process classifier with this view (View 1 GP) [157]. For all methods, we use the same logistic loss model for fair comparison. PMC uses logistic regression models for the base classifiers, and we use logistic likelihood models in GPCR and in a Gaussian process classifier for the view 1 only baseline (View 1 GP). For the MVSSL algorithm used by CoNet we use either GPCR with logistic likelihood or co-training with L_1 regularized logistic regression classifiers as the base models. To simplify the experiments we choose either co-training or GPCR as the MVSSL algorithm used by CoNet based on which gave the best MCC when no second view data is available.

Additionally to allow straight-forward comparison with the GPCR method, all of our experiments are carried out in a transductive setting, i.e., the unlabeled data (or some portion of it) for a given trial also corresponds to the test data. Note that CoNet itself is not restricted to a transductive setting. For the real data experiments, we perform experiments for CoNet with both random initialization and the contrastive divergence pre-training and also both filling in ("fill") and not filling in ("no fill") the second view with the observed second view for intances when it is available (observed). For the CoNet methods we fix the number of backpropagation gradient descent interations to 100. For all methods we report the results for the parameters giving the best average performance, where averages are taken across 100 or more random splits of the data, which essentially corresponds to reporting results of model selection if labels were available for some or all of the unlabeled data. Thus we avoid the model selection issue which is common practice in this type of scenario (e.g., [5, 25, 118, 178, 179]), and esssentially shows the results achievable given an ideal model selection method for the scenario. Since there is usually a very limited amount of labeled training data in the MVSSL setting, standard model selection approaches like cross-validation often fail [176], so the common procedure of reporting subsequent performance after model selection would not be at all representative of the underlying methods' performances but rather of the (poor) performance of the model selection approach used. Model selection in this scenario is still an open problem [78]. We discuss the model selection issue in more detail and alternative model selection approaches in Chapter 8. In this chapter we propose and compare some semi-supervised model selection approaches that are good candidate methods and demonstrate their effectiveness for model selection for this scenario of MVSSL with very limited labeled data.

6.5.1 Synthetic Data Experiment

We present results for an illustrative 2D data experiment, for the task of learning a function to separate two overlapping sets of Gaussian-distributed data. Data for two views was generated independently from the same Gaussian distribution for each class. In this way the two views come from the same distribution, but are conditionally independent given the class label - an ideal scenario for multi-view semi-supervised learning algorithms. We vary the mean fraction of second view data available from 0% to the ideal case of 100%, by removing each data instance from the second view completely at random with fixed probability corresponding to each fraction. For each trial, 2 labeled training points and 200 unlabeled points, were generated for each class using the two Gaussian distributions. Figure 6.3 shows a sample of the generated data in each view.

This data set demonstrates a simple case where existing single-view approaches are generally not well-suited. In this case, feature-splitting cannot be effective since both features are needed for sufficiency; splitting the features would result in different data classes largely overlapping in both views. Additionally there are no clear clusters - the marginal distributions look similar to unimodal groupings of points.

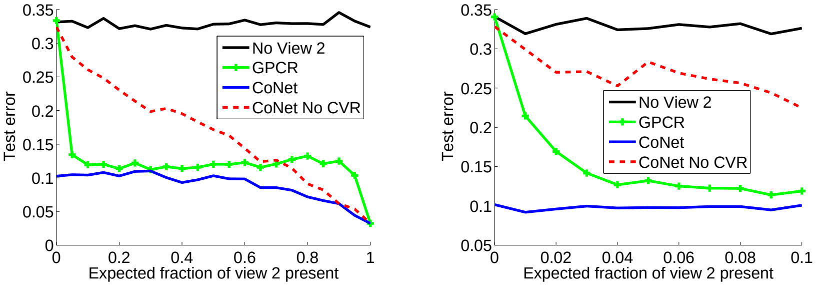


(a) View 1 (b) View 2

Figure 6.3: Sample of two views of data generated for an ideal 2D test case

We choose the state-of-the-art Gaussian process co-regularization algorithm [209] as the base algorithm to be applied after filling in the missing views with our CoNet method. In addition we use the version of this algorithm that can handle missing views to compare our method with, as it is the state-of-the-art approach [209]. In addition we report results for comparing with a view-mapping approach - an approach that only directly tries to learn a mapping from view 1 to view 2 using the available data. This corresponds to using our same feature generation network approach to generate the second view, without using the proposed bias, corresponding to Equation 6.2.

First we varied the mean fraction of second view data available from 0.0 to 1.0 in increments of 0.05. The experiment was repeated for 200 random samples of the data, and average test error and standard deviation is reported in Table 6.1 and Figures 6.4 and 6.4b.



(a) From 0 (no view 2) to 1 (all) (b) Zoomed in (0.0 to 0.1)

Figure 6.4: Test error vs. mean fraction of view 2 present for the 2-Gaussian data set

The proposed feature generation approach was found to perform significantly better than using the same base classifier with a single view of the data, or using the state-of-the-art GPCR method, especially in two extreme ranges of having very little view 2 data, and having close to the amount

Table 6.1: Mean ± std. dev. of test error from 200 trials for each method on the 2-Gaussian data, for 0% second view data available.

View1 GP	**PMC**	**GPCR**	**CoNet**
0.331±0.125	0.442±0.075	0.334±0.125	**0.103±0.058**

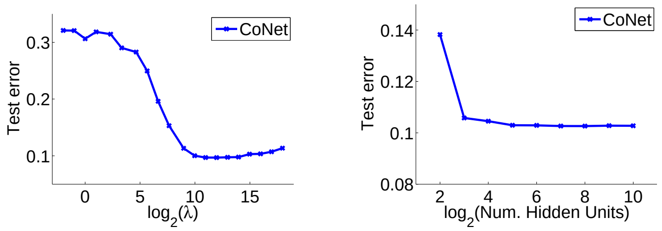


Figure 6.5: Performance criteria vs. contrasting view regularization parameter and vs. number of hidden units in hidden layer 1 for 0% second view data for the 2-Gaussian data set

of view 2 data needed to achieve the best performance. Additionally without the contrasting view regularization (CVR) term, and with the exact same network structure and approach to initialization and training, the feature generation approach ("CoNet CVR") took much more view 2 data to come close to the same level of performance as CoNet. We also show the results of repeating the experiment zoomed in more closely on the beginning region, this time varying the mean fraction of view 2 data present from 0.0 to 0.1 in increments of 0.01. The results are shown in Figure 6.4.

Furthermore, the results for the single view case - i.e., no view 2 data available are shown in Table 6.1, here also compared with the state-of-the-art single view method, pseudo-multi-view co-training (PMC). In this case PMC fails because the features cannot be partitioned in such a way to form sufficient views - in this case both features are needed to separate the classes well. This highlights the need for a more complex mechanism to generate the new view from the existing ones, which CoNet provides.

6.5.2 WebKB Course Data Experiment

The WebKB Course data set is a collection of 1051 websites from four universities, belonging to two categories: course websites or non-course websites. There are 230 websites in the course category, and 821 in the non-course category, making the data set unbalanced. The first view consists of text on the webpage itself, the second view consists of the link text of links from other webpages linking to the webpage. We use co-training as the base MVSSL algorithm to be used after filling in the missing views with CoNet for this data set.

We obtained the webpage and link text data[2] then applied standard text pre-processing using Weka [80] to obtain 2,168 features in the text view and 338 features in the link view. As in [25], for each experiment iteration we randomly sample 3 course and 9 non-course instances for labeled training. The remaining instances were used for the unlabeled data and also testing - a transductive setting so that we could compare with GPCR. We then varied the mean fraction of second view data available from 0.0 to 1.0 in increments of 0.1. Here the second view is missing completely at random - that is for a given fraction, each view 2 instance is present with probability given by that fraction. We repeated the experiment 100 times for each fraction value and report the mean results. For the base classifier for co-training we used $L1$ regularized logistic regression, with the the regularization parameters set to 0.001 for view 1 and 0.01 for view 2 throughout since these worked well for basic co-training when view 2 was completely available - though as long as these values were not too large (less than 1) the performance stayed basically the same. For the comparison state-of-the-art methods GPCR and PMC we varied all of the parameters by powers of 10 and report the results for the best set of parameters in each case.

6.5.3 Chemical Toxicity Data Experiment

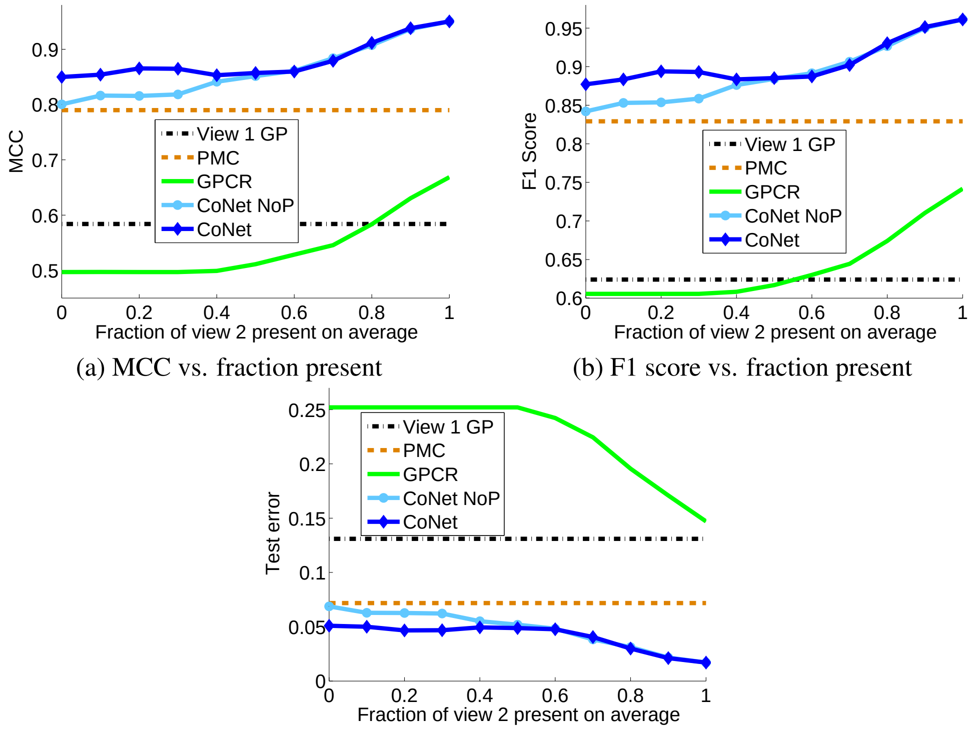
We next evaluated these methods on a chemical toxicity prediction task using a data set from the Environmental Protection Agency (EPA) TOXCAST program [103] (`http://www.epa.gov/ncct/toxcast/`) which includes experimental results conducted on 309 unique chemical pesticides. In vitro tests were performed with 624 different assays - we take the results of these tests as the feature set for the second view. Since both the animal toxicity endpoints and the in vitro second view data are time consuming and expensive to obtain (e.g the study cost millions of dollars and took more than a year), this data set fits the MVSSL with partially observed views

[2] Available here: `http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-51/www/co-training/data/`

scenario well. After basic pre-processing, e.g., removing duplicates and compounds with missing or inconclusive endpoint results, the data set consists of 225 chemical compounds with 597 view 2 features. For the class label we took the toxicity endpoint of "tumors on mouse liver", resulting in 68 positive and 157 negative instances so this data set is also imbalanced. To obtain a large set of related unlabeled data, we searched the PubChem database (`http://pubchem.ncbi.nlm.nih.gov/`) for all compounds with the keyword "pesticide" or "herbicide," resulting in an additional 1262 compounds added to the data set. To obtain the common, readily-available view 1, we extracted numerical chemical descriptors from the full set of compounds using the DRAGON software (version 5) [187] for the atom-centered fragment descriptors, resulting in a total of 103 features in view 1. For each trial, we randomly sampled half of the labeled data to be used as training data, and the other half to be included with all of the unlabeled data and for testing. Since only those data instances from the original TOXCAST collection have the second view available, the maximum obtainable fraction of view 2 data present is only approximately 0.15. Therefore for this data set we only tested two cases: no view 2 data (labeled fraction present of 0.0) and all available view 2 data (labeled fraction present of 0.15). For this data set we use GPCR as the MVSSL algorithm used by CoNet.

6.5.4 Results - WebKB Course

The overall results for the Course data are shown in Figure 6.6. This plot shows CoNet with pre-training (denoted as "CoNet") and without pre-training (denoted as "CoNet NoP") compared with the other methods for varying amounts of expected fraction of view 2 data present (observed), from no view 2 data (0.0) to all view 2 data (1.0). Again the other methods are the Gaussian process classifier with the single view ("View 1 GP") [157], the state-of-the-art Gaussian process co-regularization (GPCR) [209], and the state-of-the-art single view method, pseudo-multi-view co-training (PMC) [43]. GPCR required significantly more view 2 data to perform better than single view learning for this data. However CoNet was able to take advantage of the available second view data, obtaining the best performance. Also, in this case using pre-training resulted in a significant improvement for CoNet when limited view 2 data was available.



(a) MCC vs. fraction present

(b) F1 score vs. fraction present

(c) Test error vs. fraction present

Figure 6.6: Test error vs. mean fraction of view 2 present for the WebKB Course data set

In Table 6.2 we show the effect each component of CoNet has, and also the difference between filling in cases with available view 2 data (denoted "fill") and using only the generated view 2 data (denoted "no fill"). That is we correspondingly fix one or both of λ_1 and λ_2 to 0, i.e., "No Reg" corresponds to both fixed to 0, "VMR Only" to $\lambda_2 = 0$, and "CVR Only" to $\lambda_1 = 0$. We show results for the version of CoNet with pre-training and only for MCC, but the other performance criteria have similar trends, and the trends for no pre-training are also similar except that using the available view 2 data becomes the better strategy sooner, at the fraction of 0.5. Note that for fraction present equal to 0.0, the "fill" and "no-fill" results are the same since there are no available view 2 instances to fill in, and for 1.0 since view 2 is present for all instances all "fill" results are the same.

From these results we observe a general trend - at first, with less view 2 data available (observed), using the generated view 2 as opposed to filling in the real view is more effective, and further the contrasting view component is more important. As more view 2 data becomes available, so that a better mapping to view 2 can be learned, then filling in the available view 2 data becomes the better strategy, and the view-matching component becomes more important. Usually both components are needed for CoNet to achieve its best performance, and in most cases one or both components have a significant effect on performance. For the case of limited view 2 data one reason that filling in the available view 2 data does not help might be that the generated view 2 data is very different from the available view 2 data since there is not yet enough to learn a very accurate view mapping function. Another reason using the real view 2 where available becomes a better strategy as more view 2 data is observed is because the real view 2 data has built-in the desirable properties for MVSSL methods, e.g of sufficiency for classification, whereas for the generated view we can only estimate these properties.

Table 6.2: Mean ± std. dev. of MCC from 100 trials for each method on the WebKB Course data, for varying amounts of average second view data available in fraction of all data instances. Comparison for the case of using pre-training and both the view-matching and contrasting view components ("CoNet") with neither component ("No Reg."), just the view-matching component ("VMR Only") and just the contrasting view component ("CVR Only"). The first half, "fill" corresponds to filling in cases with available view 2 data, i.e., using whatever view 2 data is available and "no fill" to using only the generated view 2 data.

		0.0	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
fill	**CoNet**	**0.850** ±0.126	0.791 ±0.093	0.761 ±0.102	0.747 ±0.133	0.717 ±0.159	0.764 ±0.165	0.811 ±0.137	**0.879** ±0.062	**0.912** ±0.019	**0.938** ±0.017	**0.950** 0.010
	No Reg.	0.832 ±0.092	0.685 ±0.169	0.648 ±0.176	0.654 ±0.188	0.655 ±0.176	0.630 ±0.186	0.629 ±0.203	0.643 ±0.182	0.680 ±0.175	0.805 ±0.083	**0.950** 0.010
	VMR Only	0.832 ±0.092	0.690 ±0.159	0.643 ±0.183	0.661 ±0.174	0.634 ±0.181	0.698 ±0.228	0.732 ±0.226	0.801 ±0.069	0.910 ±0.020	0.937 ±0.014	**0.950** 0.010
	CVR Only	**0.850** ±0.126	0.789 ±0.103	0.753 ±0.122	0.743 ±0.146	0.712 ±0.173	0.702 ±0.223	0.736 ±0.235	0.848 ±0.149	0.881 ±0.095	0.874 ±0.074	**0.950** 0.010
no fill	**CoNet**	**0.850** ±0.126	**0.854** ±0.111	**0.865** ±0.066	**0.865** ±0.045	**0.853** ±0.120	**0.857** ±0.104	**0.860** ±0.099	0.857 ±0.111	0.856 ±0.109	0.856 ±0.111	0.858 0.105
	No Reg.	0.832 ±0.092	0.838 ±0.068	0.837 ±0.110	0.834 ±0.092	0.835 ±0.091	0.834 ±0.092	0.834 ±0.090	0.832 ±0.090	0.834 ±0.089	0.832 ±0.092	0.835 0.093
	VMR Only	0.832 ±0.092	0.834 ±0.094	0.836 ±0.089	0.814 ±0.102	0.830 ±0.100	0.837 ±0.065	0.837 ±0.077	0.834 ±0.085	0.834 ±0.088	0.837 ±0.085	0.838 0.083
	CVR Only	**0.850** ±0.126	0.843 ±0.134	0.858 ±0.101	0.849 ±0.126	0.852 ±0.119	0.850 ±0.125	0.851 ±0.119	0.850 ±0.126	0.852 ±0.120	0.851 ±0.119	0.851 0.125

6.5.5 Results - Chemical Toxicity

The results for the chemical toxicity data are summarized in Table 6.3. For this data set, unlike the text data set, using pre-training for the network (denoted as "CoNet") was somewhat detrimental

Table 6.3: Mean ± std. dev. of MCC, F1 score, and test error from 100 trials for each method on the Chemical Toxicity data, for varying amounts of average second view data available in fraction of all data instances.

	MCC		**F1 Score**		**Test Error**	
	0.0	**0.15**	**0.0**	**0.15**	**0.0**	**0.15**
View 1 GP	0.122±0.077	0.122±0.077	**0.456±0.041**	0.456±0.041	0.470±0.042	0.470±0.042
PMC	0.054±0.084	0.054±0.084	0.272±0.081	0.272±0.081	**0.359±0.032**	**0.359±0.032**
GPCR	0.113±0.078	0.159±0.085	0.417±0.055	**0.468±0.049**	0.415±0.035	0.433±0.041
CoNet NoP	**0.150±0.084**	**0.188±0.079**	0.440±0.057	0.463±0.055	0.396±0.040	0.378±0.038
CoNet	0.114±0.081	0.132±0.074	0.425±0.055	0.426±0.053	0.425±0.044	0.389±0.034

to performance compared to the randomly initialized net (CoNet NoP). Aside from the type of data (e.g., chemical descriptors as opposed to images or text), this may also be due to overfitting of the generative model since there are many more features in view 2 than view 1 in this case. Further improvement may be possible by more thorough experimentation with the pre-training approach used.

Although PMC achieves slightly lower test error than the CoNet methods, it has significantly worse scores under the balanced performance criteria (MCC and F1 score) which are more indicative of efficacy for this data. The results indicate that essentially the method cannot detect the positive cases well but still has low test error due to the highly imbalanced nature of the data. On the other hand CoNet scores highly under the more balanced performance criteria, and still manages to reach nearly the same test error in the case of the small amount of partial view data available. This is similar when CoNet (NoP - the no pre-training version - in particular) is compared with the other methods. With respect to MCC, arguably the most balanced criterion, CoNet obtains significantly better performance compared to all other methods. With respect to F1 score, the single view GP classifier has a slightly better score for the expected fraction of 0.0 view 2 data present and GPCR has a slightly better score for the fraction of 0.15. However these are not significantly different from the CoNet NoP scores. To give an idea of how the methods compare under the different criteria, we show the results of ANOVA with multi-comparison tests in Table 6.4. An entry of “1” indicates a significant difference in the means of the given performance criterion for the two methods at the five percent level.

Table 6.5 shows the comparison between CoNet with no pre-training (NoP) with both view

Table 6.4: ANOVA multi-comparison test results for each of MCC, F1 score, and test error criteria on the Chemical Toxicity data, for 0.15 fraction of view 2 data present. A “1” indicates significant difference in mean between the two methods at the 5 percent level.

	MCC					**F1 Score**					**Test Error**				
	View 1 GP	**PMC**	**GPCR**	**CoNet NoP**	**CoNet**	**View 1 GP**	**PMC**	**GPCR**	**CoNet NoP**	**CoNet**	**View 1 GP**	**PMC**	**GPCR**	**CoNet NoP**	**CoNet**
View 1 GP	0	1	1	1	0	0	1	0	0	1	0	1	1	1	1
PMC	1	0	1	1	1	1	0	1	1	1	1	0	1	1	1
GPCR	1	1	0	1	1	0	1	0	0	1	1	1	0	1	1
CoNet NoP	1	1	1	0	1	0	1	0	0	1	1	1	1	0	1
CoNet	0	1	1	1	0	1	1	1	1	0	1	1	1	1	0

matching regularization (VMR) and contrasting view regularization (CVR) and with one or neither, corresponding to setting the appropriate parameter/s to 0. For this data including both components was necessary to achieve the best performance.

Table 6.5: Mean ± std. dev. of MCC, F1 score, and test error from 100 trials for the CoNet method on the chemical toxicity data. Comparison for the case of using no pre-training and both the view-matching and contrasting view components (“CoNet”) with neither component (“No Reg.”), just the view-matching component (“VMR Only”) and just the contrasting view component (“CVR Only”). The first half, “fill” corresponds to filling in cases with available view 2 data, i.e., using whatever view 2 data is available and “no fill” to using only the generated view 2 data.

		MCC		**F1 Score**		**Test Error**	
		0.0	**0.15**	**0.0**	**0.15**	**0.0**	**0.15**
fill	**CoNet NoP**	**0.150±0.084**	**0.188±0.076**	**0.440±0.057**	**0.463±0.053**	**0.396±0.040**	**0.378±0.035**
	No Reg.	0.091±0.079	0.157±0.079	0.405±0.055	0.436±0.057	0.426±0.038	0.382±0.035
	VMR Only	0.091±0.079	0.157±0.080	0.405±0.055	0.436±0.057	0.426±0.038	0.382±0.036
	CVR Only	**0.150±0.084**	0.171±0.079	**0.440±0.057**	0.458±0.054	**0.396±0.040**	0.394±0.035
no fill	**CoNet NoP**	**0.150±0.084**	0.168±0.073	**0.440±0.057**	0.451±0.053	**0.396±0.040**	0.387±0.033
	No Reg.	0.091±0.079	0.147±0.086	0.405±0.055	0.434±0.060	0.426±0.038	0.392±0.039
	VMR Only	0.091±0.079	0.137±0.087	0.405±0.055	0.427±0.062	0.426±0.038	0.397±0.039
	CVR Only	**0.150±0.084**	0.148±0.078	**0.440±0.057**	0.446±0.053	**0.396±0.040**	0.407±0.035

6.6 Conclusion

An obstacle for multi-view semi-supervised learning approaches when applied to real world data is the lack of complete multiple view data. For example, a common scenario is that one data view is readily and cheaply available, but additional views may only be available in some cases and may be costly to obtain. Current work to address such scenarios is limited and also each previous approach has some limitations. In summary, existing approaches either are not able to incorporate partial view information when available or are not applicable or effective with limited amounts of additional view data. Additionally, the previous works either make restrictive assumptions, are method-dependent, or fail to incorporate a way of enforcing the approach to be useful for subsequent application of multi-view semi-supervised learning algorithms. To address these limitations, we introduced a unified approach for multi-view semi-supervised learning with missing views that can be applied to the full range of problems with incomplete view information. We propose a feature-generation learning approach, based on fine-tuning random nonlinear feature functions, for learning a mapping to fill in missing views, with a particular bias incorporated that is motivated by theoretical results on multi-view semi-supervised learning. This is carried out using additional terms in the objective function of a feature generation network model that encourages the data instances in distinct views to be nearby different unlabeled instances. We demonstrated the efficacy of our method with synthetic and real data experiments and for these experiments our method achieved superior performance to two recent state-of-the-art approaches designed for the case of MVSSL with missing views.

Chapter 7

Active View Completion

7.1 Introduction

An active learning method is a machine learning method that actively queries data instances to obtain additional information from an oracle about those instances, for example, the information could be a class label and the oracle could be a human annotator. Active learning has been extensively studied, particularly for the case of active labeling, and a consistent improvement over passive strategies, where selected instances are chosen at random according to the underlying distribution, in terms of achieving the same accuracy with fewer samples, has been clearly demonstrated in practice [170, 171], and recently asymptotically in theory for label-querying [81, 10]. This concern for selecting useful samples is especially motivated by the consideration of costs associated with obtaining ground truth information - often there is considerable cost associated with invoking an oracle in terms of time, resources, and money, for instance hiring human annotators.

Here we investigate a new research direction of active learning at the interface of active learning and multi-view semi-supervised learning. Multi-view semi-supervised learning exploits the idea of consensus for predictors in distinct sets of features called views, for instance a web-page can be characterized by multiple views including the text on the webpage and the anchor text of the hyperlinks of pages that link to the webpage. This predictive consensus concept is specifically exploited for the case of semi-supervised learning. For instance co-training, one of the most widely used multi-view learning algorithms, works by identifying unlabeled instances that can be confidently predicted in one view but not the other, allowing these instances to be labeled and used in improving the hypothesis for the other view, so that in some cases even with very few labeled instances, with enough unlabeled data perfect or high accuracy hypotheses can be identified [25, 9]. The consensus idea has also been exploited for standard label-query active learning; the active learning approach of querying those unlabeled instances for which the view specific hypotheses disagree has been shown to generally out-perform single-view active learning methods both in theory and practice [130, 134, 79, 192]. Here we consider a new type of active multi-view semi-supervised learning scenario, where the instances are not queried for labels, but for missing data views, and the goal is to find the most useful queries to complete for the purposes of performing multi-view semi-supervised learning.

In many problems, one view of the data is readily available or relatively inexpensive to obtain, but additional views can have a significant cost associated with them, so that we cannot just ignore this cost and assume the additional views are ubiquitous as multi-view learning approaches generally require. Furthermore, obtaining ground truth information for the labels can be too expensive in terms of time, cost or resources, or even be infeasible, so that it may be preferable to take advantage of other less expensive information that can be queried in hopes of improving our learned model for the data; in this case this information takes the form of additional view data. In many real-world problems obtaining additional view data is expensive and time consuming, though still preferable to obtaining ground truth label information. One area this is particularly apparent in is informatics for the life sciences, such as the bio-, chem-, and health-informatics areas, for example, predicting chemical toxicity, drug viability, diagnosis, pathology, etc. There are various additional profiles or views that can be obtained with some associated expense, but obtaining true endpoints can cost millions of dollars and take years. As a specific example, the standard chemical toxicity endpoints are the result of extensive animal testing that require a large amount of both time and money to obtain, but there are intermediate, potentially indicative in-vitro features that can be obtained for a fraction of the cost [103]. Furthermore, with multi-view learning the performance typically levels out after a certain amount of unlabeled data is included, in other words, if we can select specifically those most useful instances to obtain the additional views for, we may not need to waste additional expense on completing the rest of the instances to still allow multi-view semi-supervised learning to be successful.

Recently, this active view completion idea has been explored under the Bayesian Gaussian process framework [209]. However there is still much remaining work for understanding active view completion scenarios. This previous work does not consider at all when an active strategy may or may not be useful. Additionally the methods proposed for active selection are not directly applicable to multi-view semi-supervised learning methods in general, as they require, for example, estimates of predictive variance, which happen to be convenient to compute under their proposed framework. Furthermore these methods have only been tested on data of very low dimension (3 features or less in each view) and may have trouble with data in higher dimensions, which is more commonly encountered. A key overlooked issue with applying active selection strategies to view completion for multi-view semi-supervised learning is that in some cases an active strategy may not offer a benefit over a passive (i.e., random selection) one. For example, if two views are conditionally independent given the class label, then no matter which selection strategy is used to select an instance to complete with the second view, we have a fixed chance to obtain any possible point in the second view belonging to the same class. This means aside from influencing which class is selected there would be no difference in the active and passive strategies in this case since given the class each possible value for the second view is equally likely for both selection strategies. In other words, if two different strategies select two (possibly distinct) instances, as long as those instances are from the same class then the completed values for the missing view are equally likely for both strategies.

In this chapter we further explore this new research direction of active view completion, and attempt to shed some light on the issue of when an active strategy can be beneficial. We consider two important questions. First, are there selection strategies that can offer improved performance over the passive strategy for common multi-view learning approaches? Second, to what extent and under what conditions can an active approach offer an improvement over a passive one? To help answer these questions, we give a theorem that is essentially a bound on the expected number of useful instances the passive strategy can find based on a measurement of the expansion between views. This suggests that if there is less expansion between views, i.e., views are more dependent, than an active strategy can be more effective since the passive one will have a smaller chance of selecting useful instances, whereas an active strategy can be chosen to maximize the chance of selecting useful instances. To more clearly answer and analyze these questions, we also propose algorithms and run experiments on some synthetic cases that demonstrate when an active strategy can offer improvements and to what extent these improvements depend on underlying conditions of the data, with co-training, one of the most widely used approaches for multi-view semi-supervised learning. These experiments confirm our theoretical analysis, that the utility of an active strategy depends on the specific view relation of limited expansion, and in the case of large expansion (e.g., conditional independence of views given the labels), the passive strategy can be just as effective as an active one. We then conduct additional experiments on two real world text classification data sets, including comparison with the state-of-the-art approach of [209], which further supports our hypothesis.

7.2 Background

In the field of active learning, the most closely related work to ours is that of "active feature acquisition" [170]. This can be viewed as feature selection in reverse - we start with incomplete sets of features and the goal is to select which features to fill in by estimating which features will be most useful for the decision function based on some criteria such as confidence [216, 126], or a utility function [161]. However, this is completely different from our problem, where we consider each view as a complete feature set, and already by itself sufficient for estimating the decision function if enough labeled instances were available - so we do not need to actively acquire more features, just views to exploit redundancy when we cannot afford or are unable to obtain additional labels. Note also that the end goals are very different: the goal of the selection under our setting is to offer as much benefit as possible to the subsequently applied multi-view semi-supervised learning algorithms - i.e., the selection strategy should specifically take such algorithms into account, whereas other active acquisition settings do not take multi-view semi-supervised learning into consideration at all.

Another related line of work is on multi-view active learning, as mentioned in the introduction, where it is assumed all views are present, and the queries are for filling in the missing labels [130, 134, 79, 192]. Previous theoretical results for this scenario [79, 192] follow a similar α-expansion setting as that of [9] in which the authors proved the success of the co-training algorithm under an expansion condition on the underlying data distribution, and this is the same type of setting we consider for our theory.

There is also some related work from the field of multi-view learning. In [2] the authors consider multi-view learning when only some views are present in some instances, and a view mapping function for filling in the missing views is available, and provide error bounds comparing using the completed views and not using the completed views. However, theirs is not an active setting, they fill in all the unlabeled data at once, whereas in our work we want to avoid this potentially costly approach, and instead want to actively, sequentially, select instances to complete. Furthermore their theory gives no way of distinguishing a difference in generalization error for different orderings of filled in instances - i.e., they only consider a passive approach.

Recently, Yu *et al.* proposed two active strategies for view completion [209]. The first is to use a conditional density estimate with Gaussian Mixture Models for computing an expectation of a posterior distribution with their Gaussian process model. This is used to compute the expected decrease in entropy if a missing view is observed according to the learned Gaussian process model, and the instance with the greatest expected decrease is selected for completion. The second approach is to select the unlabeled instance with the greatest predictive variance. They applied their approaches to two cancer prognosis prediction data sets and found improved learning rates for the active strategies as compared to a random one. This previous work has a couple of key limitations. Outside of the proposed Gaussian process framework, these selection criteria are usually more difficult to estimate and can require making additional modeling assumptions. Secondly, there is no analysis of when an active strategy might be useful.

7.3 Methodology

For our theoretical analysis, we consider the case of iterative co-training in the realizable case. This is an ideal case where there is no base error rate and in which the best possible zero-error classifiers exist in the hypothesis space for each view. This is a starting point for much theoretical analysis in multi-view semi-supervised learning [25, 58, 9, 79], and co-training is a popular and widely used multi-view semi-supervised learning algorithm.

7.3.1 Preliminaries and Assumptions

Our notations and setting follow those of some previous theoretical works providing sample complexity bounds for co-training and multi-view active learning in the realizable case, those using the assumption of α-expansion [9, 79]. For simplicity we consider the case of two views here, X_1 and X_2, and corresponding instance space $X = X_1 \times X_2$, and label space $Y = \{-1, 1\}$, and assume instances are drawn according to some distribution D over X. We assume labels are given by some underlying functions $h_1^* : X_1 \to Y$ from hypothesis space H_1 and $h_2^* : X_2 \to Y$ from hypothesis space H_2 and that for all $x \in X$ with non-zero probability mass according to D $h_1^*(x_1) = h_2^*(x_2)$. Whenever we state probabilities, e.g., $\Pr(Z \subseteq X)$ these are always with respect to the distribution D.

In order to apply iterative co-training we need some measure of confidence for a given hypothesis. Similar to [9] we assume we have a way of determining confident set $S_i \subseteq X_i$ for a given hypothesis $h_i \in H_i$, $i = 1, 2$, for which $h_i(x_i) = h_i^*(x_i)$. For instance in [9] the authors give an example of the hypothesis class of axis-parallel rectangles and the algorithm that takes the smallest enclosing rectangle of positive examples. We also use the same notation, with the boldface

$\mathbf{S}_i, i = 1,2$ denoting the event that an instance (x_1,x_2) has $x_i \in S_i$. So if S_1 and S_2 are the confident sets in view 1 and view 2 respectively, then $Pr(\mathbf{S}_1 \wedge \mathbf{S}_2)$ denotes the probability mass on instances confident in both views, and $Pr(\mathbf{S}_1 \oplus \mathbf{S}_2)$ the mass on instances confident in one and only one view.

As with general theoretical work on active learning, we assume that we have access to an unlimited pool of unlabeled instances, and there is an initial, small, set of labeled complete instances, However, in the active view completion case, we assume that only one view, without loss of generality X_1, is present in the unlabeled data, and that we must iteratively select an instance from the unlabeled data to obtain the second view for. We call such instances incomplete. Unlike typical active learning, we do not obtain labels from an oracle, only missing views for selected instances. We assume at each iteration an unlabeled view-incomplete instance is selected according to a specific selection strategy, the missing view is obtained for that instance, and the basic co-training algorithm is run, i.e., if the new complete instance is confident in one view but not the other than we can transfer the label and update one of the hypotheses, otherwise we cannot use the completed instance at the current iteration, though it may become useful at a later iteration. The process is iterated for some number T of iterations.

As in [9, 79, 192] we are interested in the set $S_1 \oplus S_2$, which we use as shorthand to denote those instances for which we are confident in one and only one view. Note the underlying hypothesis is only necessarily updated if we find an unlabeled instance $x \in S_1 \oplus S_2$ since we need confidence in one view in order to transfer the label to the other view, and we need a lack of confidence in the other view in order for the label transfer to provide new information. We say a selected instance $x \in X$ is useful if $x \in S_1 \oplus S_2$ and that these instances are the ones that will cause the hypotheses to be updated. Thus we are particularly interested in estimating how many useful instances, which we denote by n_u, will result from T iterations with a given selection strategy. Since for active view completion we don't have the second view present, we can only estimate if an instance will be useful, so we are interested in $E[n_u]$ given a selection strategy. The baseline selection strategy is that typically used in work on active learning, that of random selection, i.e., at each iteration choosing an unlabeled, incomplete instance at random according to D. We denote this strategy as **RAND**. Since no effort is made in selecting which instance to complete, this is also called the **passive** approach. As mentioned in the introduction, we are interested in exploring whether active selection strategies can offer an improvement, i.e., an increased number of expected useful instances, over the random (passive) selection strategy, and under which conditions we can expect this improvement. The co-training view completion procedure is summarized in Algorithm 3.

Algorithm 3 Co-Training with View Completion

```
Input: Complete labeled data $L = \{(\vec{x}_{1i}, \vec{x}_{2i}, y_i)\}_{i=1,\ldots,n}$, incomplete unlabeled view 1 data $U_{I1} = \{\vec{x}_{1i}\}_{i=n+1,\ldots}$, hypothesis spaces $H_1$ and $H_2$ and associated learning algorithms $A_1$ and $A_2$, number of iterations $T$, and selection strategy $\mathbf{G}$.
Output: Final hypotheses $h_1^T \in H_1$ and $h_2^T \in H_2$.
  Obtain initial $h_1^0$ and $h_2^0$ and initial confident sets $S_1^0$ and $S_2^0$ using algorithms $A_1$ and $A_2$ with data $L$
  Assign unlabeled complete ordered data set $U_C$ to be the empty set
  $i \leftarrow 0$
  while $i < T$ do
    Select $\vec{x}_1 \in U_{I1}$ according to $\mathbf{G}$ and remove from $U_{I1}$
    Obtain the $\vec{x}_2$ corresponding to selected $\vec{x}_1$ from oracle
    if $(\vec{x}_1, \vec{x}_2) \in S_1^i \oplus S_2^i$ then
      Set $y$ equal to label given by $h_j^i$ for $\vec{x}_j$ in the confident region
      Add $(\vec{x}_1, \vec{x}_2, y)$ to $L$
      Obtain $h_1^{i+1}$, $h_2^{i+1}$, $S_1^{i+1}$, and $S_2^{i+1}$ using $A_1$ and $A_2$ with $L$
      Cycle through $x \in U_C$ in order until $x$ is found such that $x \in S_1^{i+1} \oplus S_2^{i+1}$; if found move to $L$ and update $h_1^{i+1}$, $h_2^{i+1}$, $S_1^{i+1}$, and $S_2^{i+1}$ as above, and repeat cycle.
    else if $(\vec{x}_1, \vec{x}_2) \in S_1^i \wedge S_2^i$ then
      Set $y$ equal to label given by $h_j^i$ for $\vec{x}_j$ (must agree by assumptions); add $(\vec{x}_1, \vec{x}_2, y)$ to $L$.
      Set $h_j^{i+1} = h_j^i$, $S_j^{i+1} = S_j^i$.
    else
      Add $(\vec{x}_1, \vec{x}_2)$ to the end of $U_C$
      Set $h_j^{i+1} = h_j^i$, $S_j^{i+1} = S_j^i$.
    end if
    $i \leftarrow i + 1$
  end while
```

7.3.2 Active Approach and Definitions

Ideally we would directly choose an instance x_1 with corresponding x_2 having opposite confidence; instead we can only hope to maximize our chances of choosing an $x \in S_1 \oplus S_2$. Thus we propose to alternatingly choose $x_1 = \arg\max_{x_1 \in S_1} Pr(x_2 \in \bar{S_2}|x_1)$ and $x_1 = \arg\max_{x_1 \in \bar{S_1}} Pr(x_2 \in S_2|x_1)$, since hypotheses in both views must be expanded. However, we do not know the conditional distribution between views in advance. The simple alternative approach we use in our experiments is to iteratively select an instance closest to the current confident (labeled) region in view 1, followed by selecting a confident (labeled) instance that is as far as possible from the other confident points. We denote this simple strategy as **ACTIVE**.

Since in general the probability of success of an active strategy is unknown, and to avoid results based on a particular active strategy, our results here focus on a bound on the expected number of useful instances selected by the random strategy, as a function of the number of iterations, and characteristics of the relationship between the views. Therefore we give the following definitions. The first quantities of interest for characterizing this relationship are the average probability mass of the useful region for given confident sets over sequences of T iterations, and also the maximum mass of this region. When discussing the relationship between data views, it is helpful to think of the expansion idea as presented by [9]. Here we use "expansion" to mean the general probability mass associated with the useful region, i.e., $Pr(\mathbf{S}_1 \oplus \mathbf{S}_2)$ for given confidence regions, which captures how much the confident regions can expand into the rest of the data space.

Definition 1. *Supremum of average probability mass of useful region Given distribution D, hypothesis spaces H_1, H_2, and learning algorithms A_1 and A_2,over any initial data L from D with $Pr(\mathbf{S}_1^0 \vee \mathbf{S}_2^0) < \rho_0$, and any possible consequent sequences $x^0, x^1, \ldots, x^T$, $(S_1^0, S_2^0), (S_1^1, S_2^1), \ldots, (S_1^T, S_2^T)$, given by Algorithm 3 with $Pr(\mathbf{S}_1^{T-1} \wedge \mathbf{S}_2^{T-1}) < 1$, $p^*(\rho_0)$ is defined as follows.*

$p^*(\rho_0) = \sup\{\frac{1}{T}\sum_{i=0}^{T-1} Pr(\mathbf{S}_1^i \oplus \mathbf{S}_2^i)\}$

Definition 2. *Supremum of probability mass of useful region*

Under the same setting as for Definition 1, r is defined as follows.

$r = \sup\{Pr(\mathbf{S}_1^i \oplus \mathbf{S}_2^i)\}$

Note the dependence on the initial size of the confident region ρ_0. This is done in order to avoid stronger assumptions using maximum or minimum over all sets, since for example, we would generally expect the size of the useful region to start small, grow, then shrink, so though it's maximum size may be larger, on average it is reasonable to assume it is not too large. When trying to match these definitions with data, these averages bounds only make sense however if we start from an initial limited size of the confidence region, since otherwise, if applied to any sequence, trivially they would become lower and upper bounds. In general for examples we will usually assume ρ_0 is relatively small, corresponding to a small set of initial complete labeled data, and an associated small region of confidence. We simply denote $p^*(\rho)$ with p^*.

For a given S_1 and S_2, $Pr(\mathbf{S}_1 \oplus \mathbf{S}_2) = 1 - Pr(\mathbf{S}_1 \wedge \mathbf{S}_2 \cup \bar{\mathbf{S}_1} \wedge \bar{\mathbf{S}_2})$, so that as the set of instances that are confident in both views, denoted by $S_1 \wedge S_2$, grows large $Pr(\mathbf{S}_1 \oplus \mathbf{S}_2)$ will eventually become smaller. In general we would expect the most benefit from an active strategy when $Pr(\mathbf{S}_1 \oplus \mathbf{S}_2)$ is small, that is the distribution is not expanding [9] too much. A large $Pr(\mathbf{S}_1 \oplus \mathbf{S}_2)$ can imply the confident region in one view expands to most or all of the other view, which can be unrealistic for real world data [9].

Finally, we upper bound the amount the unconfident region shrinks after each successfully chosen instance for view completion, i.e., each time a new labeled instance is added for one of the views, Pr$(\bar{\mathbf{S}_1} \wedge \bar{\mathbf{S}_2})$, the mass of the region we are unsure of the labels for, decreases by at most β. In general it is reasonable to expect such an upper bound to be relatively small, since otherwise a single iteration could result in going from most of the space being unconfident to all or most of the space being as little as a step away from becoming confident (i.e., we could finish after just a few iterations of co-training even with a random strategy).

Definition 3. *Supremum of decrease in probability mass of unconfident region*

Under the same setting as for Definition 1, β is defined as follows.

$\beta = \sup\{Pr(\bar{\mathbf{S}_1}^i \wedge \bar{\mathbf{S}_2}^i) - Pr(\bar{\mathbf{S}_1}^{i+1} \wedge \bar{\mathbf{S}_2}^{i+1})\}$

7.3.3 Theoretical Result

Theorem 1. *Under the same setting as for Definition 1, the expected number of useful instances selected for the passive strategy, $E[n_u|\boldsymbol{RAND}]$ is upper bounded by*

$$p^*T + r\beta T(T-1)(\beta(T-2)+3)/6$$

.

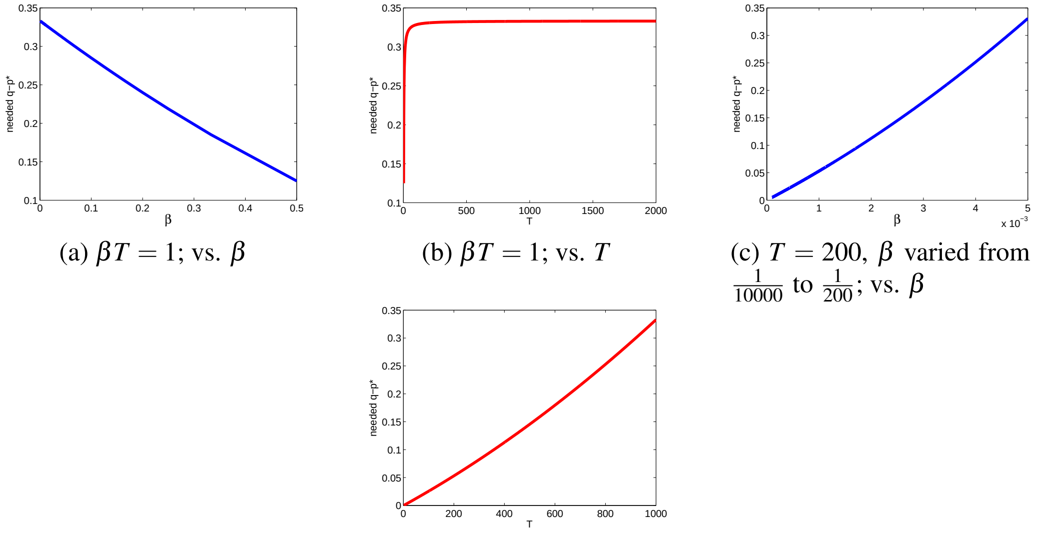
This theorem says that the average probability of success of an active approach only needs to be some term depending on T, β, and r greater than the average probability mass of the useful region for an active strategy to offer an improvement over T iterations in terms of the number of useful instances found. Since the probability of success of the proposed active strategy depends on how easy it is to predict the conditional distribution, we can interpret this as saying that the predictive structure between views must be sufficiently strong as compared to the average expansion between views, as captured by the average size of the useful region, and there is something of a trade-off between these two quantities since a large expansion means one point in one view could correspond to many different points in the other view so we may not be able to estimate where the other point will be with high confidence.

If β is sufficiently small and for smaller T, the difference needed can be quite small. However if the number of iterations becomes very large, eventually the random strategy may catch-up if enough of the initially selected instances that were not useful become useful in the future, and we need greater increase in probability of success of our confidence estimation strategy to still guarantee an increased number of useful instances selected by the active strategy. See below for additional illustration of this bound.

Proof sketch Bounding the passive selection strategy would be trivial if we were just comparing the success of selecting a useful instance at each step - which is just given by average size of the useful region. However, the difficulty lies in the fact that a selected instance that was not useful may become useful in the future. At each step, if at step i we select x and $x \notin S_1^i \oplus S_2^i$ then it must either be in $S_1^i \wedge S_2^i$, in which case it will never become useful in the future, or in $\bar{S_1}^i \wedge \bar{S_2}^i$, in which case it may become useful at a future step. Given T iterations, and starting with $E[n_u|\mathbf{RAND}]$, ideally we would like to be able to compute the probability mass associated with each number of useful selections, however since we only have upper bounds, we cannot do so. Instead we can treat each selected (sampled) instance as a Bernoulli random variable with success denoting the instance becoming useful, so that the number of useful instances is the sum of these random variables. Then each selected instance has some chance of success throughout the whole set of T iterations, and then we can upper bound this success. Under the same setting as for Definition 1, let $(S_1^0,S_2^0),(S_1^1,S_2^1),\ldots,(S_1^T,S_2^T)$ and $x^0,x^1,\ldots,x^{T-1}$ be any achievable sequence of confident sets and selected instances for the given selection strategy, and let $p^i = Pr(\mathbf{S}_1^{i-1} \oplus \mathbf{S}_2^{i-1})$ for $i = 1,2,\ldots,T$. Then $E[n_u|\mathbf{RAND}] = Pr(\text{First selection succeeds}|\mathbf{RAND}) + \ldots + Pr(\text{Tth selection succeeds}|\mathbf{RAND})$. For the i^{th} step, $Pr(i^{th}\text{ succeeds}|\mathbf{RAND}) = p^i + Pr(x^i\text{ becomes useful in the future})$, i.e., the probability it succeeds when it is first drawn plus the probability it fails when drawn and becomes useful later on. Then $Pr(x^i$ succeeds at $(i+j)^{th}$ step) is upper bounded by $r\beta(1+(j-i)\beta)$. This is because at each future step, if it failed until that step, it will only get another chance to succeed if the sample drawn at that step is a success (since otherwise the confident regions don't change), and if it belongs to a region that was previously all unconfident but became confident after an update, whose probability mass is upper-bounded by β. Additionally it has a chance of multiple chances to succeed at that step, if it fails again but other samples drawn after failed and then succeeded (note the set order of retrying previously failed instances is important for this bound to hold for any instance). Finally each instance drawn next has one fewer future steps to succeed at, so adding everything up we get: $E[n_u|\mathbf{RAND}] \le p^*T + r\beta T(T-1)(\beta(T-2)+3)/6$ $\square$.

As a specific example, if we have $\beta = 0.005$, $T = 100$, and $r = 0.4$, then we need 0.1152 greater average probability of selecting a useful instance with a given active strategy than the average probability mass of the useful region in order to guarantee that the expected number of useful instances produced by the active strategy is greater than that produced by the passive strategy over the T iterations.

To help visualize what the bound in the condition means for different values of β and T here we show a series of graphs in Figure 7.1. In all plots, we plot the difference between the average probability that a given active strategy selects a useful instance, denoted by q, and p^* needed for the theorem to guarantee improvement, i.e., $r\beta(T-1)(\beta(T-2)+3)/6$. We might assume the unconfident region actually decreases by a roughly constant amount each time given by β so that we may have no more than $1/\beta+1$ iterations if we succeed each time, so we should not evaluate the result for $\beta T > 1$. Therefore, for the first two plots, we fix $T\beta = 1$ and fix $r = .5$, a relatively large upper bound that particularly makes sense as an upper bound if a uniform distribution is given over the whole space $X_1 \times X_2$ for certain X_1 and X_2, e.g., if $X_1 = X_2 = [a,b]$. Then for the first two plots, we vary T from 2 to 2000 and set $\beta = 1/T$, and plot the needed $q - p^*$ vs. β in the first one, and T in the second one. For the third plot we fix $T = 200$ and vary β from $\frac{1}{10000}$ to $\frac{1}{200}$. In the final plot, we fix $\beta = 1/1000$ and vary T from 2 to 1000;



(a) $\beta T = 1$; vs. β

(b) $\beta T = 1$; vs. T

(c) $T = 200$, β varied from $\frac{1}{10000}$ to $\frac{1}{200}$; vs. β

(d) $\beta = 1/1000$, T varied from 2 to 1000; vs. T

Figure 7.1: Needed differences $q - p^*$ with $r = 0.5$ and $\beta T \leq 1$ vs. β and T for different values of β or T.

These plots show that at first, over fewer number of iterations than those needed to finish labeling the whole space X, i.e., when βT is significantly less than 1, we can easily expect more useful selected instances with an active strategy versus the passive one, since the needed probability of successes for the average strategy only needs to be slightly larger than the average probability mass of the useful region. But as the number of iterations increases toward the number needed to completely fill in the space X, the random selection strategy catches up - i.e., previously non-useful selections become useful, so the active strategy must be more effective to offer a benefit.

7.3.4 Active Approach for General Classification Problems

The algorithm and active selection approach of the previous section is specifically designed for a basic and ideal scenario - in general it may not be applicable or effective. Here we introduce a modified algorithm and active selection approach to apply to real world classification problems. The modified algorithm is shown in Algorithm 4. Since true confidence values cannot usually be known, the base multi-view semi-supervised learning algorithm is called after each update to relearn the model from scratch given the current set of complete view data, which further allows any multi-view semi-supervised learning algorithm to be used.

Algorithm 4 Active View Completion

Input: Complete labeled data $L = \{(\vec{x}_{1i}, \vec{x}_{2i}, y_i)\}_{i=1,\ldots,n}$, complete unlabeled data $U_c = \{(\vec{x}_{1i}, \vec{x}_{2i})\}_{i=n+1,\ldots,m}$ (possibly empty), incomplete unlabeled view 1 data $U_{I1} = \{\vec{x}_{1i}\}_{i=n+m+1,\ldots}$ multi-view semi-supervised learning algorithm A, and selection strategy $\mathbf{G}$, number of selection iterations T.

Output: Hypothesis h.

```
Apply $A$ to $L$ and $U_c$ to obtain hypothesis $h$
$i \leftarrow 0$
while $i < T$ do
    Select $\vec{x}_1 \in U_{I1}$ according to $\mathbf{G}$ using results of $A$, and remove $\vec{x}_1$ from $U_{I1}$
    Obtain the $\vec{x}_2$ corresponding to selected $\vec{x}_1$ from oracle
    Add $(\vec{x}_1, \vec{x}_2)$ to $U_c$
    Apply $A$ to $L$ and $U_c$ to obtain hypothesis $h$
    $i \leftarrow i + 1$
end while
```

There are three main issues when applying the previous active selection approach to general data. First, in general learners require both classes to learn - this can create an issue when making selections, especially for unbalanced data, as one class may be preferred in the selections. This in turn could cause an increasing bias toward selecting the same class as the algorithm progresses. To avoid the issue of estimating/preserving class ratios we use sampling instead of selecting extreme values for the active strategy. A "top fraction" is taken as input specifying what fraction of unlabeled data should be used to select the next instance from. Second, unlike ideal scenarios, or the synthetic experiment we describe in Section 7.4.1, usually confidence cannot be determined with certainty. I.e., we can only estimate the confidence in a prediction. For this reason, we use ranking instead of selection based on a confidence threshold. This also allows the approach to be directly applicable to cases without probabilistic output like support vector machines, as instances can be ranked based on distance from the decision boundary. Third, the previous approach assumes unlimited unlabeled data, e.g., it assumes we can always select some point from the unconfident or confident regions. However, real data is limited; even when there is much more unlabeled data than labeled, it may be that there are no unlabeled points in a given region. To address this, aside from the ranking criteria which directly avoids relying on a confidence threshold that may not be met by any unlabeled instance, we fix the size of the selection set to a top number to select from, based on a top fraction parameter. Also since we randomly select the index to complete from a set, this modified selection strategy already has some exploration built into it.

Algorithm 5 Active Selection Strategy

```
Input: Incomplete unlabeled view 1 data $U_{I1} = \{\vec{x}_{1i}\}_{i=n+m+1,...}$, current model $h$, top number $k$, and binary indicator $s$.
Output: Instance $\vec{x}$ to complete.
  Use $h$ to assign confidence scores $c_i$ to each $\vec{x}_{1i} \in U_{I1}$
  Rank $c_i$ in ascending order if $s = 1$, o.w. descending order
  Choose and return an instance $\vec{x}$ at random from set corresponding to top $k$ ranked $c_i$
```

The general active selection strategy, which we use in our real data experiments, is summarized in Algorithm 5. Given the results of training a multi-view semi-supervised learner on the current set of complete data and labeled data, the learned hypothesis is used to assign a confidence score to each incomplete unlabeled instance - e.g., for non-probabilistic models this could just be the absolute distance from the decision boundary. The instances are ranked by confidence scores, and one in some top number of instances (with the number determined by a top fraction parameter) is randomly selected for completion. This process is alternated between least and most confident sets.

Another possibility for an active strategy is to try to directly estimate either the missing view data itself, or the predicted confidences for the missing data for the associated view model. However, since most real data is of high dimension, this estimation task is very challenging, especially since the amount of complete data to use for the estimation is very limited until much has already been filled in.

7.4 Experimental Study

We first give results on synthetic experiments, for which we could control the expansion between views. We follow Algorithm 3, Co-Training with View Completion, for our experiments with three different selection strategies.

For the real world data, we use the modified algorithm, Algorithm 4 - Active View Completion, as unlabeled data is no longer unlimited and ground truth confidence is unknown. We compare with the active view completion approach discussed in [209] using predictive variance estimates.

In our experiments, we assume all of the labeled data is already complete. This is reasonable since an obvious initial choice would be to fill in the missing views for the labeled instances, especially since this set is small, and performance for most multi-view semi-supervised learning methods is dependent on this set - e.g., with no complete labeled data most multi-view semi-supervised learning algorithms could not even be applied at all.

7.4.1 Synthetic Data

For our synthetic experiments, we use the axis-aligned rectangle problem, where the positive class corresponds to the interior of an axis-aligned rectangle in 2D. We fix this rectangle to have corners $(.1, .15)$ and $(.9, 85)$, so that about half the points are in each class. To generate the two views

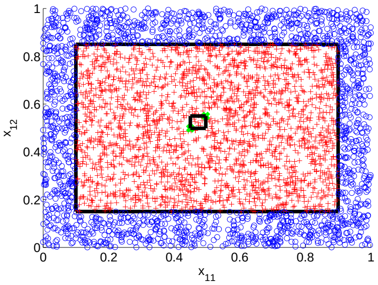


Figure 7.2: Axis-aligned rectangle, sample data generated

with controlled expansion, we alternatively sample a point uniformly at random from $[0,1] \times [0,1]$ for each view. To generate the corresponding point in the other view, we sample from a uniform square region centered at the starting view point, with radius (distance from the center to a side) given by a_{exp}, so that the larger a_{exp} the more the greater the expansion between views, with the further restriction that the point must belong to the same class. We automatically select small starting rectangles by selecting two points in the center of the rectangle, separated by around 0.05 units. An example of one set of data generated for one view is shown in Figure 7.2, the large black rectangle is the ground truth hypothesis, the smaller one the starting hypothesis. We run the three selection strategies with the active view completion for co-training for a few thousand iterations, and repeat 500 times with a different random data samples of 6000 points each time. We do this for 3 increasing a_{exp} values of 0.02, 0.04, and 0.1 - note $a_{exp} = 0.1$ essentially means a point in one view can correspond to any point in a specific region with width .2 in the other view.

7.4.1.1 Experiment Set-up: Confidence Estimation and Selection Strategy

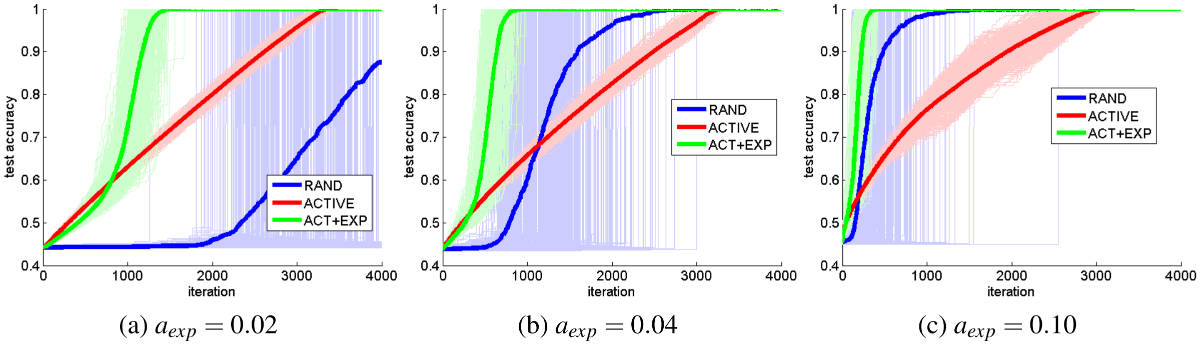
For the active strategy, as described previously, we alternate between choosing a point that is confident that we expect to be unconfident in the other view, and choosing a point that is unconfident that we expect to be confident in the other view. In order to estimate if a point will be confident, here we propose a simple and efficient approach for the synthetic data. We note that since confident regions must agree, we can view confident (unconfident) points closest to the unconfident (confident) region as being more likely to be unconfident (confident) in the other view. Therefore we select the confident (unconfident), unlabeled, incomplete point closest to the unconfident

(confident) region, and denote this strategy **ACTIVE** in our experiments.

Then we use three selection strategies in our experiments. The first is the passive strategy, where unlabeled points are selected at random, denoted by **RAND**. The second is the active strategy described above. Finally, we found the active strategy can be too conservative when expansion is larger, and additionally it may be desirable to explore uncertain regions of the data space to reveal previously-unknown connections between confident regions in different views. Therefore our final strategy, denoted **ACT+EXP**, combines exploration with the active strategy, and repeats the cycle of employing the active strategy for two round followed by randomly selecting an unconfident point in the next round.

7.4.1.2 Experiment Results

We plot the results in terms of test accuracy vs. the number of iterations in Figure 7.3, where test accuracy is the number of correctly predicted labels divided by the total number of predictions. Though we collected number of useful selections vs. iteration, we do not plot these results here due to space constraints, but describe them below. The base colors are blue for passive (random) strategy, red for active, and green for active plus exploration. To clearly compare the results of the 500 trials, each individual trial is plotted in a lighter color shade, and the means are plotted in thick darker lines. From these results, it is clear with small expansion between views ($a_{exp} = 0.02$) the active strategy completely out-performs the passive (random) one. The typical pattern for



(a) $a_{exp} = 0.02$ (b) $a_{exp} = 0.04$ (c) $a_{exp} = 0.10$

Figure 7.3: Test Accuracy vs. Iteration for 3 selection strategies on the synthetic data set, averaged over 500 random trials

a single run of the passive strategy results in very little improvement (in terms of accuracy or useful selections) for a long time, followed by a sharp jump in improvement, when many of the previously non-useful selections become useful after certain updates. Also both active approaches have significantly less variance than the passive strategy. As the expansion between views grows larger, the number of iterations where the active strategy achieves better test accuracy than the passive strategy decreases, and the passive strategy reaches perfect accuracy sooner on average. However, the active plus exploration strategy clearly dominates the other two strategies in all cases - this simple modification greatly speeds up the accuracy improvement for the case of potentially greater expansions to the confident regions - which occur when the a_{exp} value is larger. Since the active strategy has a conservative approach of choosing points close to the confident region, it is more likely to choose useful points, but at the cost of not usually choosing the points that allow the most expansion in the confident regions when a_{exp} is large. Thus, each useful point for the conservative active strategy only increases the size of the confident set by a small amount as compared to the other strategies.

7.4.2 Real World Data Sets

7.4.2.1 WebKB Course Data Set

The first data set we use is a webpage classification one. The WebKB Course data set is a collection of 1051 websites from four universities, belonging to two categories: course websites or non-course websites. There are 230 websites in the course category, and 821 in the non-course category, making the data set unbalanced. The first view consists of text on the webpage itself, which is something that is always available. The second view consists of the link text of links from other webpages linking to the webpage; in general this view could be missing as it takes extra time and resources to find and gather incoming links and their associated text for a given website. We obtained the webpage and link text data[1] then applied standard text pre-processing using Weka

[1] Available here: `http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-51/www/co-training/data/`

[80] to obtain 2,168 features in the text view and 338 features in the link view.

7.4.2.2 Modified Course Data Set

The Course data set was originally used as an example of a data set for which co-training would be successful based on a sufficient condition of view conditional independence given the class label [25]. Therefore we would expect the expansion between views to be large, and a passive (random) selection strategy to be effective on this data set. To see if restricting the expansion between views effects the performance of the different selection strategies, we also modified this data set and repeated the experiments with the modified data set. To create the modified Course data set with restricted expansion, we assigned the view 2 data as the new view 1 data. Then we took the original view 2 data and divided it into positive and negative classes. Next, for each class, we found the nearest neighbor of each instance in that class. Then we swaped pairs of nearest neighbors, starting with those farthest apart, removed the swapped pair from further consideration, and repeated, until no pairs remain. This new set then became the new view 2 - containing the same data but in a different order. The idea is that an instance in view 2 should still be close to its corresponding value in view 1 since it was only swapped with a nearest neighbor - a nearby data point. In this way within each class view 2 should be dependent on view 1. Also since the view 2 points are shuffled around they will be near different instances so there should still be significant expansion between views so that multi-view semi-supervised learning algorithms should still be effective.

7.4.2.3 Citeseer Data Set

In addition we evaluate the methods on the Citeseer data set. The Citeseer data set is a collection of scientific articles split into six categories (“Agents”, “AI”, “DB”, “IR”, “ML”, and “HCI”). The first view consists of the text from the abstract of each article, and the second view is the citation profile, the list of papers a given paper is cited by or cites in the database. We obtained a version of the data set[2] with binary vectors for each article indicating if a word is present or not in that article,

[2] Available here: http://www.cs.umd.edu/ sen/lbc-proj/LBC.html

and built binary citation vectors for each with a 1-entry for a feature indicating the paper cites or is cited by the other paper given that corresponding index. We removed all papers with fewer than five other papers in the collection that cite or are cited by the paper. This resulting data set contains 1164 documents with 3703 features in view 1 and 1164 features in view 2. As in [210] we take the largest class ("DB") as the positive class and the remainder as the negative class, resulting in 222 instances in class 1 and 942 instances in class 2.

7.4.2.4 Experiment Set-up

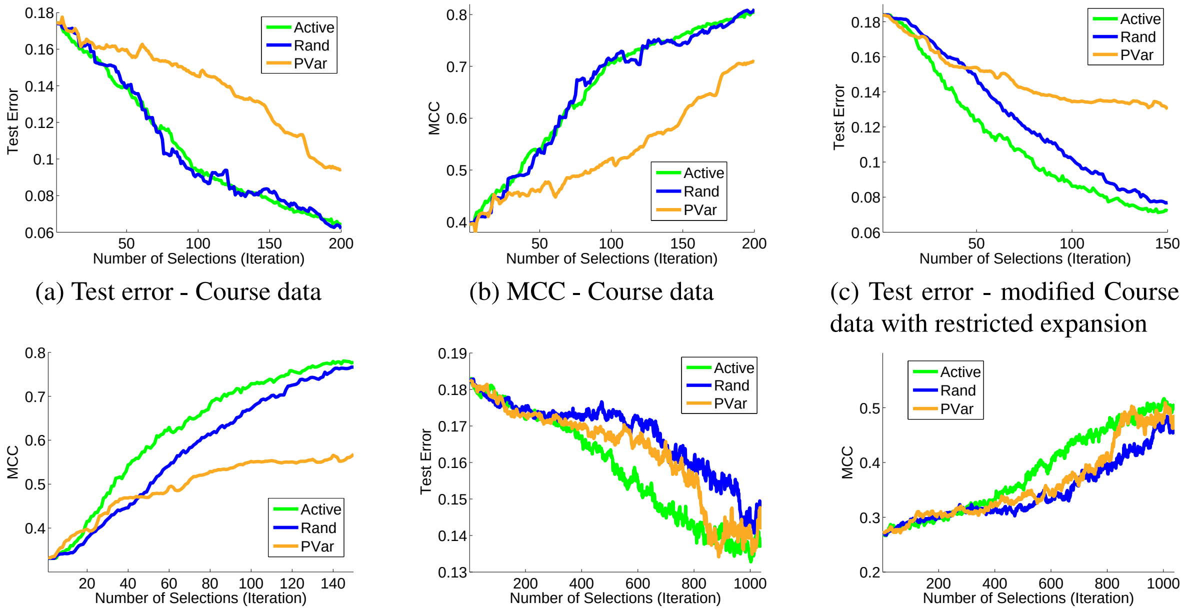
For the Course data set, as in [25], for each data set and each experiment iteration we randomly sample 3 course and 9 non-course instances for labeled training data. For the Citeseer data set, as in [209] we randomly sample 4 positive and 20 negative instances for labeled training data. The remaining instances were used for the unlabeled data, initially missing view 2, and also for computing performance. Since the data is imbalanced we report both test error and Matthews Correlation Coefficient (MCC). If tp denotes the number of true positive predictions, fp the number of false positives, fn false negatives, and tn true negatives, then MCC is given by

$$\frac{(tp)(tn)-(fp)(fn)}{\sqrt{(tp+fp)(tp+fn)(tn+fp)(tn+fn)}}.$$

Then for each experiment iteration we ran the active selection algorithm of Section 7.3.4, Algorithm 4, with a given selection strategy for a few hundred iterations (until performance started to level off). The selection strategies used were: passive (random) selection denoted **Rand** and the active strategy described in Section 7.3.4 and Algorithm 5 denoted **Active**. The top fraction for the Course data sets was set to 0.25, and 0.05 for the Citeseer data set, and we also report results for varying these top fraction values. This difference is explained by the larger expansion present in the Course data set as discussed in the Results section. Additionally we compare with the state-of-the-art active view completion selection strategy of selecting based on estimating predictive variance in a Gaussian process co-regularization model [209]. We denote this method by

PVar. We use co-training as the base multi-view semi-supervised learning algorithm to be used with active view completion as it works best on these data sets when the data is complete. To stay consistent with the Gaussian process model, which uses a logistic likelihood, we use L_1 regularized logistic regression as the base classifier for co-training.

7.4.2.5 Experiment Results



(a) Test error - Course data

(b) MCC - Course data

(c) Test error - modified Course data with restricted expansion

(d) MCC - modified Course data with restricted expansion

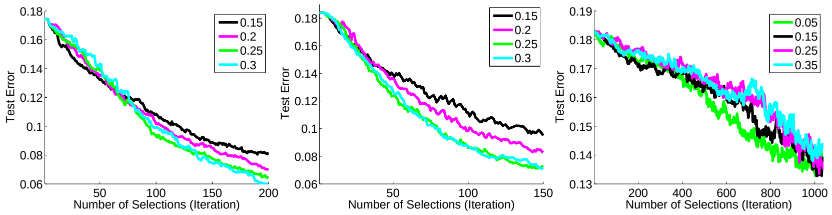
(e) Test error - Citeseer data

(f) MCC - Citeseer data

Figure 7.4: Test error and MCC vs. iteration for the different selection strategies on the Course data set, modified Course data set, and Citeseer data set, averaged over 100 random trials

The results are shown in Figure 7.4, for both the original Course data set, the Course data set modified to have restricted expansion between views, and the Citeseer data set averaged across 100 random trials, for the first few hundred iterations. The results agree with our hypothesis - in the original Course data set, the passive strategy (**RAND**) works just as well as an active strategy (**Active**), due to the large expansion between views. For the modified data set with the restricted expansion, the active approach starts to out-perform the passive one and it takes more iterations for the passive strategy to catch up. The difference between the original data set and the one modified to have restricted expansion between views can be clearly seen with respect to the difference in performance between the active and passive strategies. Also in both cases the state-of-the-art method using predictive variance under a gaussian process co-regularization model (**PVar**), does not achieve good performance. This is likely due to a mismatch between the model and the data - i.e., the particular model may not be well suited for this data. For the Citeseer data set, without modification our proposed active approach offers an improvement over the passive strategy, and also out-performs the PVAR active approach. PVAR does improve over random selection, but not by as much early on as our proposed approach.

The next set of plots in Figure 7.5 shows results with the experiments repeated, but with varying top fractions used for the active strategies. Here there is a trade-off for the Course data set, a smaller value resulted in a slightly sharper performance increase at first, but the corresponding method was then overtaken by the passive strategy after more iterations. Also it is interesting to note that for the Course data set, a larger fraction worked best, whereas for the Citeseer data set a much smaller fraction was better. This is consistent with having larger expansion for the course data so more exploration (i.e., a strategy closer to random selection) worked better.



(a) Test error - Course Data

(b) Test error - modified Course data with restricted expansion

(c) Test error - Citeseer Data

Figure 7.5: Test error vs. iteration for active selection for varying top fractions of data to choose select from, on the Course data set, modified Course data set, and Citeseer data set, averaged over 100 random trials

7.5 Conclusions and Future Work

We have explored a new research direction, active view completion, and analyzed when an active strategy can be useful, with new algorithms, theoretical results, and experimental study. One key observation from our study is that active view completion is different from active label acquisition in that the benefit of the active strategy may depend on the relationship between the views. Designing an effective active selection strategy may be more challenging for active view completion. Our experimental results demonstrated cases where a passive strategy is as or more effective than active ones. We feel active view completion is an interesting new area of research that offers new challenges and has much potential for further study. An example of one future direction is the combination of active view completion and active label acquisition, in particular the combination with the highly effective co-testing approach [134], which may work best with a different approach for choosing the instances for view completion.

Chapter 8

Model Selection for Semi-Supervised Learning

8.1 Introduction

For every multi-view semi-supervised learning algorithm, and every general semi-supervised learning algorithm, in practice it is necessary to select the specific tuning or hyper parameters for the method, using the available training data. This process is called model selection. Model selection is a major issue for semi-supervised learning problems involving very limited labeled data, since the small amount of labeled data makes it difficult to reliably estimate predictive performance of a model.

However, in the work on multi-view semi-supervised learning, and semi-supervised learning in general, the issue of model selection is most often avoided entirely [25, 29, 22, 137, 164, 129, 155, 30, 111, 5, 220, 53, 178, 118], for example, by reporting the best results found over a grid of hyper-parameters, the idea being that this is the best performance a particular method could achieve if there was some way to select that best model. However this best performance is meaningless for real wold applications if there is not some way to select the model. Additionally, for the case of extremely limited training data, the performance of general approaches to model selection like cross-validation deteriorates, and most other semi-supervised model selection methods are only designed to work for specific semi-supervised learning mehods and so are not generally applicable. A recent survey article lists model selection for semi-supervised learning as one of five open problems in model selection: “Very little has been done for model selection in semi-supervised learning problems, in which only some training instances come with target values. Semi-supervised tasks can be challenging for traditional model selection methods, such as cross-validation, because the number of labeled data is often very small” [78].

This analysis leads us to propose an alternative, general approach to model selection for semi-supervised learning with extremely limited labeled data. Like cross-validation the approach is based on re-sampling and re-training, and also like cross-validation is already parallelized and thus can be efficiently carried out with modern parallel computing resources. The basic idea is to generate many data sets that are similar to the target one, by re-sampling the labeled and unlabeled data from the given data. In each case labels are sampled using estimated conditional distributions derived by averaging the predictions on each data instance of all models in the set of models under consideration. In this way, if most models agree on a prediction for a label, than that label will consistently be generated, but if models largely disagree on a label, then that label will vary more across the generated data sets. Additionally the prior weights given to the models can be iteratively updated, with the goal of making the generation distribution more similar to the target data distribution. By estimating the average performance for each model across the generated similar data sets, this provides a rough estimate of its performance on the target data. This approach can also be seen as an alternative way of estimating the stability of a model by evaluating its performance on many different but similar data sets. If the model does not provide stable estimates, then its performance may vary greatly for slight changes in the data set, and this will be captured by a larger averaged test error over the generated data sets.

We evaluate our similar data sampling approach on four data sets with different amounts of labeled data, and compare to existing model selection approaches, including a state-of-the-art semi-supervised model selection method, discussed in the next section. Our experimental results demonstrate the efficacy of the proposed approach.

8.2 Related Work

One of the most commonly used approaches to model selection is cross-validation. In k-fold cross-validation, the labeled training data is partitioned into k roughly equal sized sets. Then each of the k sets takes a turn as being the held-out set used for testing, and the remaining $k-1$ sets are used for training each model to be evaluated. The average performance on the held-out sets is used to estimate the models' performance, and this can be repeated and averaged over multiple random partitions. Cross-validation is one model selection approach commonly used for general semi-superised learning methods [39, 214], however it has been found that its performance can suffer when only small amounts of labeled data are available [176]. Aside from standard supervised model selection methods like cross-validation, we can roughly break related work into two categories: work that avoid avoids full model selection in some way, and work either focused on the problem of semi-supervised model selection or that uses some form of semi-supervised model selection.

8.2.1 Avoiding the Model Selection Issue

A large amount of the work on multi-view semi-supervised learning, and semi-supervised leading in general, in the literature avoids the model selection issue in some way. Therefore we briefly mention some common approaches used that essentially avoid model selection, before discussing specific model selection approaches. We can further break this category down into two sub-categories.

8.2.1.1 Reporting the Performance for Fixed Values or Best Over Hyper-parameter Grids

The work in this category trains the methods used either by arbitrarily picking fixed values for some or all hyper-parameters or by using default or heuristic hyper-parameter values or by training the methods over hyper-parameter grids, sets of different hyper-parameter combinations, and returns the best results on the test error found, sometimes with hyper-parameter sensitivity results as well [25, 29, 22, 137, 164, 129, 155, 30, 111, 5, 220, 53, 178, 118].

8.2.1.2 Selecting using a validation set typically only available for model selection

The work in this category uses a separate validation set for model selection and selects the best model according to performance on the validation set, e.g., [179, 160, 40, 52]. Note that this is also an artificial scenario, since if extra labeled data were available for model selection it could also be used for model estimation, likely making the semi-supervised learning approaches unnecessary or at least reducing the benefit they offer and most likely changing the best model as well. For example, in one work [179], for one data set, the semi-supervised learning method has access to only 2 labeled instances, but 250 are used for validation - if these had been available for training after validation, supervised learning would most likely have been sufficient.

8.2.2 Model Selection Approaches

Various approaches do exist that address the model selection issue partially or fully for the case of semi-supervised data. However most are method-dependent - specific to the probabilistic models and frameworks proposed for the particular learning algorithm. Here we discuss such approaches as well as general semi-supervised approaches.

8.2.2.1 Approaches that are restricted to certain model classes

One common category of methods is the approach of estimating the marginal likelihood also called maximum likelihood type II approaches [209] or evidence-based model selection [176]. Given specific probabilistic models, the model parameters are approximately integrated out of the data likelihood equation leaving the marginal likelihood as a function of the hyper-parameters. The hyper-parameters maximizing this marginal likelihood are then typically chosen. However this requires assuming a particular probabilistic model for the different components of the model and the data, and is thus not applicable to general semi-supervised learning methods, for instance co-training with arbitrary base classifiers in each view. Additionally, depending on the model, this approach could suffer from overfitting with limited labeled data. A different type of marginalization strategy in which some of the hyper-parameters are marginalized when estimating the model parameters has also been proposed [111]. In this case, for hyper-parameter selection, some hyper-parameters are arbitrarily fixed, and the remaining hyper-parameters are treated as missing values. The conditional probability distributions defined by the model are then used with the expectation-maximization algorithm to fit the model parameters, essentially integrating out these specific hyper-parameters. Similar to the marginal likelihood approaches with gaussian processes, Zhu, Ghahramani and Lafferty proposed a Gaussian random field model with a label entropy model selection approach used for learning some hyper-parameters [226].

Another approach is to use information criteria. For instance, Culp, Michailidis and Johnson propose a generalized additive model with transductive smoothers for multi-view semi-supervised learning [54]. The associated proposed model scoring uses the likelihood or error penalty on the labeled data in combination with an estimate of degrees of freedom for the linear smoothers, which corresponds to the trace of a smoother matrix. In addition to being method-dependent, this approach only considers the performance on the labeled data with the unlabeled data effecting only the trace of the smoother matrix. For very limited labeled training data, this could result in poor solutions since many models could fit the labeled data very well so that the estimated degrees of freedom is the determining factor, potentially resulting in overfitting for cases of many hyper-parameters.

8.2.2.2 General Approaches

Several general approaches also exist for semi-supervised model selection. An interesting state-of-the-art approach for semi-supervised model selection is metric-based model selection [167, 168, 169], which was generally found to out-perform previous model selection methods including cross-validation and various information criteria. The first approach in this category uses estimated distances between hypotheses in different classes and the target hypothesis and tests a sequence of hypothesis classes in order until the triangle inequality is violated with some previous hypothesis class. Since the sequence traversal can be terminated early at a sub-optimal model, a second approach with an adjusted distance estimate was proposed using ratios of function distance estimates to score models. Bengio and Chapados consider metric-based model selection extensions to time series data, cases without unlabeled data, and a hybrid with cross-validation [16]. However, a major limitation for the metric-based model selection occurs with extremely limited labeled training data since many or all hypothesis classes considered could all achieve perfect training error. This means if the first approach is used, the sequence traversal is terminated immediately, and if the second approach is used, all methods have equal scores of zero, so there is no way to decide between them. Additionally this method requires a nested ordering of hypothesis classes, limiting its applicability for general learning methods, since the correct sequence of hypothesis classes which should be monotonic in terms of complexity is not always clear, particularly with multiple hyper-parameters. Schuurmans *et al.* addressed this issue by proposing a new model evaluator, called ADA, as the product of the training error and a function of the ratios of the distance between a learned function on the labeled and unlabeled data from a constant function, using Kullback-Leibler divergence for classification [169] since the original distance approach did not work well for classification. Collectively these metric-based model selection approaches were demonstrated to improve over the state-of-the-art in model selection, compared against a wide variety of model selection approaches. Like the proposed method of this thesis, this method can also be applied to a grid of hyper-parameters for model selection. However, if class conditional probabilities for any instances are zero the approach has a divide-by-zero problem, which can happen for some tasks with very small amounts of labeled training data. Additionally, small amounts of labeled data may not provide reliable enough information for the estimated labeled data function distances, and many semi-supervised learning methods already generally enforce similarity in the learned function evaluated on labeled and unlabeled data in some way so this method may not be as useful with semi-supervised learning algorithms. Furthermore, to our knowledge, this method has never been analyzed in conjunction with semi-supervised learning algorithms, which is part of what is provided here.

Madani, Pennock, and Flake proposed a co-validation approach [124] in which two functions are trained on different partitions of the labeled training data, and their disagreement is measured on the unlabeled data and used along with training error to estimate test error. However this approach requires enough labeled data to allow representative functions to be learned with half the amount of training data, making it an unsuitable choice for very small amounts of labeled training data. Additionally in their semi-supervised learning experiments the approach did not improve the model selection over cross-validation (though could be helpful for active and transfer learning). A similar approach is proposed in [104], in which cross validation is extended with a disagreement measure on the unlabeled data; also similarly the approach did not improve over cross-validation, but did offer more reliable generalization error guarantees. Another similar approach was proposed called stability selection, and extended these ideas to unlabeled data for the problem of estimating the number of clusters to use in a clustering model [113].

8.3 Methodology

We assume a set of labeled and unlabeled data instances $\mathscr{D}$ are generated by a fixed joint distribution P_{XY} over $\mathscr{X} \times \mathscr{Y}$. We further make the standard assumptions that the data $\{X_i\}$ is i.i.d. and that the distribution of Y_i depends only on X_i. Since typically for semi-supervised learning, a large amount of unlabeled data is available, this means the marginal distribution P_X is well characterized by the data sample. Sampling from the marginal distribution P_X can therefore be simply accomplished by re-sampling from the full set of labeled and unlabeled data. The intuition behind the proposed approach is then that, given marginal samples $\vec{x}$ that are close to the true distribution, if we can at least approximately sample associated labels, then we can come up with a way to sample data sets that are similar to the target data set. By training different models on these synthetically generated data sets, we can get an idea of how consistently they perform on similar tasks to the target one by averaging their performance over many randomly generated similar tasks. Since we have the ground truth labels for these similar data sets, we can directly evaluate each model for them. Intuitively, if a model works well for these similar data sets then we would expect it to work well for the target data set as well. We hypothesize that considering model performance across these similar data sets can result in better estimation of model performance than relying on one particular data sample (the target data sample) with only a small amount of labeled data to perform model selection, since with the proposed approach real performance is evaluated on similar data sets for which the labels are known and test error can be directly computed.

8.3.1 Estimating Expected Test Error by Re-sampling

The goal can therefore be defined as estimating the expected test error for each model using a sampling approach, and a particular loss function $L(,)$. Typically $L(,)$ is taken to be the 0-1 loss, given by $L(a,b) = 1$ if $a \neq b$, 0 o.w. Specifically, $\mathrm{Err} = E[L(Y, \hat{f}(X))|\mathscr{D}]$ where $\hat{f}()$ is the predictive function estimator using $\mathscr{D}$ corresponding to a particular model (i.e., set of hyper-parameters), and the expectation is over both the training data $\mathscr{D}$ of a particular size and the random variable X. Each model corresponds to a distinct estimator which maps a data sample $\mathscr{D}$ to a function from $\hat{f} : \mathscr{X} \to \mathscr{Y}$ and so $\hat{f}$ is a random variable. We assume the goal is to evaluate a finite set of models $\mathscr{M}$ of size k, with some initial prior distribution over the models P_M which would usually be taken to be uniform.

Since the expected test error is just an expectation over different data samples, we can approximate it via the law of large numbers as follows

$$\mathrm{Err}_m \approx \frac{1}{d} \sum_{j=1}^{d} \frac{1}{t} \sum_{i=1}^{t} L(y_{j,i}, f_{m,j}(\vec{x}_{j,i})) \tag{8.1}$$

Here each $\mathscr{D}_k$, for $k = 1, \ldots, d$, is obtained by independently sampling a data set with the same number of labeled and unlabeled instances as $\mathscr{D}$ and t test instances by sampling $(\vec{x}_{j,i}, y_{j,i})$ from P_{XY}, and $f_{m,j}()$ is the predictive function learned for the particular model (i.e., set of hyper-parameters) m for training set $\mathscr{D}_j$. Note these training sets contain both labeled and unlabeled data. In the transductive setting, the unlabeled data is also the test data, so in this case t is the number of unlabeled instances.

Since sampling from P_X can be approximated by re-sampling (in our implementation we use without replacement so that we can partition the data) from the full set of labeled and unlabeled data, if the conditional distribution $P_{Y|X}$ were known at least at each data instance in the training data, then we could also sample Y given an X sample and thus sample from P_{XY}. Furthermore since the amount of unlabeled data is large we can estimate the test error using the generated (sampled) unlabeled data. If we assume $P_{Y|X}$ corresponds to a mixture of models in $\mathscr{M}$, which are in the form of the models that return probabilistic outputs, then $P_{Y|X=\vec{x},\mathscr{D}} \propto \sum P_{Y|X=\vec{x},M=m,\mathscr{D}} P_{M=m}$. Note also that this mixture could correspond to a single model. Therefore if the probability of each model, P_M, were known, we could generate data sets very similar to the target data set.

Therefore we propose the following iterative procedure to estimate average test error for a set of models, where we view the probabilities of the models as hidden variables. The procedure starts with an initial P_M usually taken to be uniform (i.e., $P_{M=m} = 1/k$), and a target training data set $\mathscr{D}$ with n labeled instances.

1. **Step 1:** For each $m \in \mathscr{M}$ compute $P_{Y|X,m,\mathscr{D}}$ and $f_m()$ by training the model on the target training data set $\mathscr{D}$. Average these conditional estimates together according to the current estimate for P_M. In particular, we define:

$$\hat{P}_{Y=y|X=\vec{x}} = \sum_m P_{Y=y|X=\vec{x},M=m,\mathscr{D}} P_{M=m} \tag{8.2}$$

2. **Step 2:** For each of some number d data sets, randomly sample without replacement n instances from D to use as labeled training data, and use the remainder as both unlabeled training data and test data. To each instance, assign a label by sampling from conditional distribution estimates found in the previous step, $\hat{P}_{Y|\vec{x}_i}$ for each i.

3. **Step 3:** Estimate the average test error for each model m according to Equation 8.1.

4. (*Optional*) **Step 4:** Taking the likelihood for a given model to be the exponential of the negative test error, multiply these by the current probability for each model P_M and normalize across all models for each data set. Average the result across data sets to obtain the new estimates for the hidden variables P_M.

5. (*Optional*) **Step 5:** Repeat for several iterations, or until convergence.

If we stop after one iteration, then the average estimated test error is computed using a conditional distribution with equal weight for each model, i.e., the uniformly-weighted average, which might be preferable, in particular for the sake of computational efficiency. In practice we found this approach to be effective. Also note, for continuous Y, densities are used for its distribution.

8.3.2 Addressing Additional Issues

One issue with computing the conditional distributions is that, even if all of the models agree in their label prediction, depending on the method used, the probability outputs might still be close to 0.5. In this case, the sampled data could still vary largely, with samples not too similar to the target data. Therefore we also use the average of predicted labels to estimate the conditional probabilities:

$$\hat{P}_{Y=y|X=\vec{x}} = \sum_{m} \mathbb{1}(y = f_m(\vec{x}))P_{M=m}, \tag{8.3}$$

where $\mathbb{1}(.)$ is the indicator function which returns 1 if its argument is true and 0 otherwise. Note that this definition assumes discrete labels. For other types of target variables Y, some modification is necessary. In particular, considering continuous Y and regression, conditional densities would be computed instead. In order to use the fixed output predictions in this case, a one-dimensional distribution can be fit to the set of model predictions of y for a given $\vec{x}$, using kernel density estimation [26].

Another issue arises with the combination of limited labeled data and imbalanced data. In this case, many instances might be predicted as belonging to the same class by most models. This can be an issue when sampling then, since the sampled label set might be all of one class - it might take a much larger set of sampled data sets to get a significant number that have labeled data from both classes. Therefore we also propose a balance modification in which we sample data sets until each has at least one labeled instance from each class.

8.3.3 Relationship to Expectation Maximization, Bootstrapping, and Stability Selection

If the iterative re-weighting strategy is used, and we consider the missing labels to be hidden variables, this is in some ways similar to expectation-maximization-type approaches for learning with hidden variables - another category of semi-supervised learning algorithms [224]. However there are a few key differences. First the hidden variables are used mainly for evaluation of the models, as opposed to being an integral part of the models themselves. I.e., when training the models across the different random samples of the data and labels, instead of trying to incorporate all of the estimates for the labels of the unlabeled data in the training process, these estimated labels are mainly used in evaluating the trained model. The focus is on keeping the training conditions the same as for the actual training data. Second, maximization is not performed over the expectation, since each model is trained (maximized) over its local sample and then an expected value is taken. This emphasizes the key point that, instead of using this procedure to try to infer likely values for the hidden variables, i.e., the missing labels, which could be unreliable, or update a single model, the goal is to estimate the performance of the models. In this way, if most labels cannot be predicted with certainty, a model that most consistently achieves better performance, across all of these sample data sets, will be preferred, even if it is not the most likely (assuming there is even a fully defined probabilistic model). Therefore, this method may be more closely related to the metric-based and stability selection approaches mentioned in the related work (e.g., [169, 124]), but tries to estimate this stability by fully re-training models on similar data sets as opposed to either data subsets or a one-dimensional stability criterion of similarity between function values on labeled and unlabeled data.

The proposed resampling approach is also similar to bootstrapping [85], which samples from the labeled data with replacement to generate the similar data sets, and evaluates these on the held out data for each set. However, in this limited labeled data setting, there is only a small set of labeled data to resample from, e.g., in one experiment we only have 4 labeled points. In this case most of the data sets will not be very different - there is a limited number of unique data sets that can be sampled. Furthermore there is a risk that a sample will contain only a single class, in which case the algorithms being evaluated might not even be applicable. If stratification is used to avoid this issue, the possible samples are reduced further. Enumerating possible train/hold-out combinations with stratification essentially amounts to nearly the same approach as cross-validation so this approach would suffer the same limitations as mentioned for cross-validation. Also for this reason in our experiments we only compare with cross validation as it is more widely used in this setting. Therefore another way of looking at our approach is that it extends a bootsrapping approach by using estimated labels with the unlabeled data instead. Data sets are resampled each time, but from the entire set of data with our approach, which includes the large set of unlabeled data allowing more possible data sets, and evaluation is performed on the large set of unlabeled data as well for each sample, as opposed to a very small hold out set.

8.4 Experimental Study

Here we provide experimental study of various model selection approaches for different multi-view semi-supervised learning (MVSSL) algorithms evaluated on 4 different data sets.

8.4.1 Data Sets

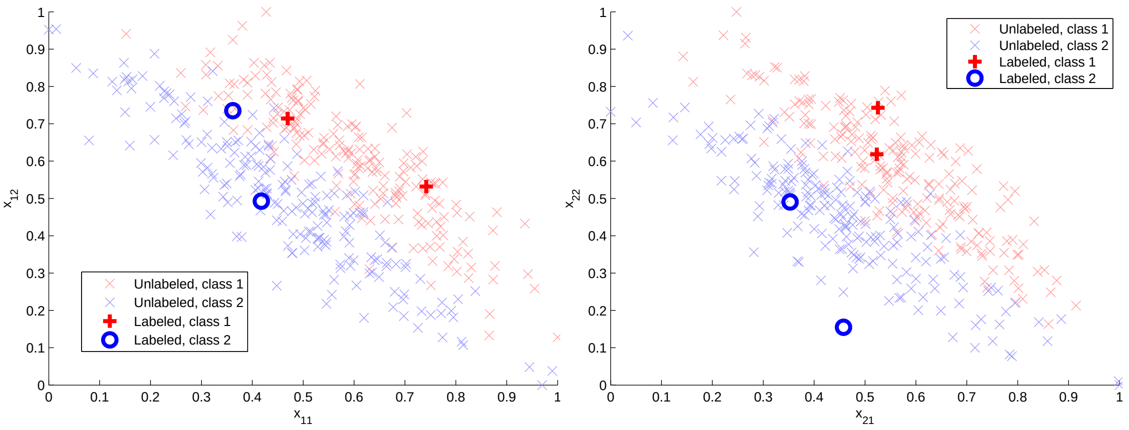
We evaluated the model selection approaches with four different data sets. The first data set is a synthetic 2-dimensional data set, the second is a webpage classification data set, the third a document classification data set, and the fourth an image classification one. Below we describe these data sets, and their characteristics are summarized in Table 8.1.

Table 8.1: Data sets, characteristics, and multi-view semi-supervised learning algorithm used.

Data Set	**Num. Labeled**	**Num. Unlabeled**	**Num. View 1 Features**	**Num. View 2 Features**	**Class Ratio num. pos. / num. neg.**	**MVSSL Method Used**
Synth	4	400	2	2	1.000	Co-Regularization [209]
WebKB	12	1039	2168	338	0.280	Co-Training [25]
Citeseer	24	1164	3703	1164	0.236	Co-Training [25]
Coil	20, 40	1420, 1400	1024	n/a	0.818	Manifold Co-Regularization [179]

8.4.1.1 Synthetic Data Set

The synthetic data in each view was generated from two slightly overlapping 2D Gaussian distributions, with the same pair of distributions used for both views. Specifically, for each class data was sampled from a zero-mean Gaussian distribution in two-dimensions with covariance {{ 16, 0 },{ 0, 1 }}, and was then transformed with the rotation matrix {{ $\cos(\frac{\pi}{4})$, $-\sin(\frac{\pi}{4})$ },{ $\sin(\frac{\pi}{4})$, $\cos(\frac{\pi}{4})$ }}; then the offset of $\{1,1\}$ was added for the positive class, and $\{-1,-1\}$ for the negative class. To generate the data each instance was sampled from the distribution for one of the classes in view 1, and independently from the same class distribution in view 2. This way the two views are conditionally independent given the class label - an ideal scenario for multi-view semi-supervised learning algorithms. The data in each view was normalized to have minimum 0 and maximum 1 after the sampling. For each trial, 2 labeled training points and 200 unlabeled points, were generated for each class. Figure 8.1 shows a sample of the generated data in each view.



(a) View 1

(b) View 2

Figure 8.1: Sample of two views of data generated for 2D test case

8.4.1.2 WebKB Course Data Set

The WebKB Course data set is a collection of 1051 websites from four universities, belonging to two categories: course websites or non-course websites. There are 230 websites in the course category, and 821 in the non-course category. The first view consists of text on the webpage itself, the second view consists of the link text of links from other webpages linking to the webpage.

We obtained the webpage and link text data[1] then applied standard text pre-processing using Weka [80] to obtain 2,168 features in the text view and 338 features in the link view. As in [25], for each experiment iteration we randomly sample 3 course and 9 non-course instances for labeled training. The remaining instances were used for the unlabeled data.

8.4.1.3 Citeseer Data Set

The Citeseer data set is a collection of scientific articles split into six categories ("Agents", "AI", "DB", "IR", "ML", and "HCI"). The first view consists of the text from the abstract of each article, and the second view is the citation profile, the list of papers a given paper is cited by or cites in the database. We obtained a version of the data set[2] with binary vectors for each article indicating if a word is present or not in that article, and built binary citation vectors for each with a 1-entry for a feature indicating the paper cites or is cited by the other paper given that corresponding index. We removed all papers with fewer than five other papers in the collection that cite or are cited by the paper. This resulting data set contains 1164 documents with 3703 features in view 1 and 1164 features in view 2.

As in [210] we take the largest class ("DB") as the positive class and the remainder as the negative class, resulting in 222 instances in class 1 and 942 instances in class 2. Also as in [210], for each experiment iteration we use 4 randomly sampled class 1 instances and 20 randomly sampled class 2 instances to make up the labeled training set, and the rest for the unlabeled data.

[1] Available here: `http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-51/www/co-training/data/`

[2] Available here: http://www.cs.umd.edu/ sen/lbc-proj/LBC.html

8.4.1.4 Coil Data Set

The Coil data set is built from the Coil20 data set, commonly used as a benchmark data set for manifold-based approaches to semi-supervised learning. The data set consists of 1440 32x32 pixel greyscale images of 20 different objects, taken at various various angles. We obtained the data set from the website[3] of the first author of a previous work on manifold regularization [177]. We created a binary classification task by splitting the objects into the categories of "toys," corresponding to 9 objects, and "other household objects," corresponding to the remaining 11 objects. We followed the same approach as [179] for computing the fixed kernels and adjacency matrices for the data (using 1-nearest-neighbor and fixed kernel width). To form the labeled and unlabeled sets we also followed the approach of [179], using stratified sampling to sample 2 images from each category to form the labeled set and used the rest as unlabeled data, for each experiment trial.

8.4.2 Preliminary Synthetic Data Study

We first performed some preliminary study with the synthetic data to get an idea of the effect of updating the weights in the resampling approach, and to generate plots for qualitative comparison of the model selection methods showing how the estimated scores compare to ground truth.

For the preliminary synthetic we used the two Gaussians data set described in the previous section, and co-regularized logistic regression. The figure showing a sample of the data in both views is re-produced here for convenience in Figure 8.1. Here the L_1 regularization hyper-parameters for each view are fixed to be equal, so that results can be displayed in 3-D plots. Therefore, there are two regularization hyper-parameters, the L_1 regularization hyper-parameter, λ, and the co-regularization hyper-parameter μ. The set of models to evaluate are taken to be a grid of combinations of these two hyper-parameters, with λ ranging from 2^{-20} to 2^3 by incremental powers of 2, and μ similarly ranging from 2^{-40} to 2^{25}, but multiplied by a starting value equal to the number of labeled instances over the number of unlabeled instances.

Since other approaches are method-dependent, the state-of-the-art metric-based model selec-

[3] Available here: http://vikas.sindhwani.org/manifoldregularization.html

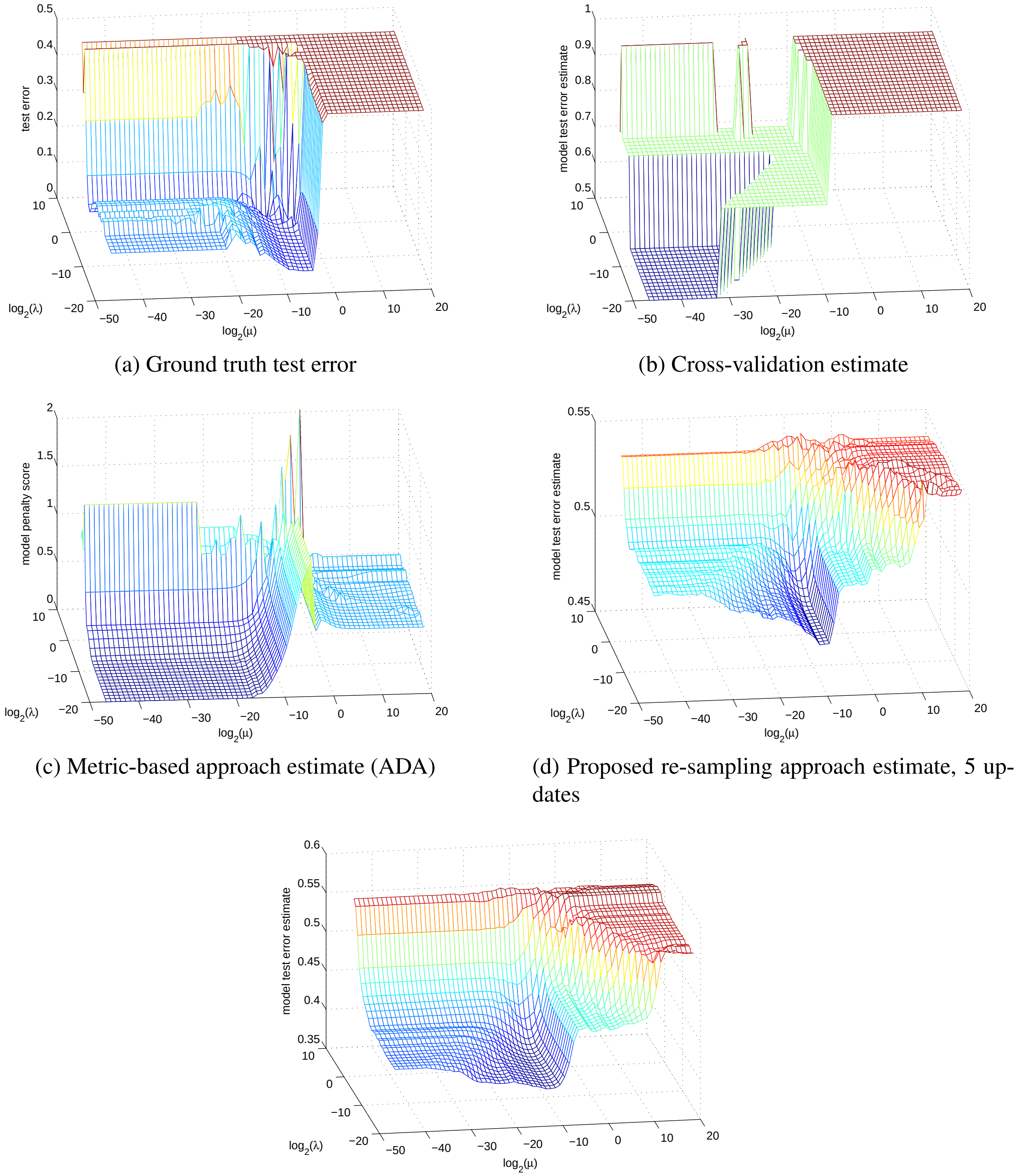
tion approach (ADA) [169] is taken as the main competitor to the proposed model selection approach, with cross-validation used as a baseline. When computing the ADA evaluator to avoid divide by 0 scenarios a small amount is added to probabilities of zero in the experiments.

Results are shown in Figure 8.2 for a particular training data set, as 3D meshes showing the estimated model score for each pair of hyper-parameters. The results are shown for both 5 and 20 updates of the conditional probability estimates for the proposed re-sampling approach. The figure shows the average score or test error estimates for each method evaluated over the grid of hyper-parameters, with the hyper-parameters on the x and y axes and the z axis corresponding to the estimate. The true test error is also shown in Figure 8.2(a). In this case, both the cross-validation and metric-based model selection approaches estimate their best scores for models in a sub-optimal region of the joint hyper-parameter space; the estimates corresponding to these methods are not accurate for this limited amount of labeled data in this case, and choosing the models with the best estimated performance would result in selecting sub-optimal models. The proposed re-sampling method however is able to come quite close to correctly estimating the low test error regions of the joint hyper-parameter space, and models with the best estimated performance result in lower test error.

We found applying the re-weighting updates generally smoothed-out the model score plots but did not have too much of effect on which models achieved the best scores. Therefore for the following experiments, for simplicity, we did not perform any weight updates for our resampling method.

8.4.3 Experiment Procedure

As mentioned in the previous section, the state-of-the-art metric-based model selection approach (ADA) [169] is taken as the main competitor to the proposed model selection approach, which we denote **ADA**. We denote cross-validation [85] as **CV**. For the cross-validation approach we use leave-one-out cross-validation as the size of the labeled data is small. We compare with the .632+ bootstrap estimator [85] as well, and denote this method **.632+**. We also compare with the model



(a) Ground truth test error

(b) Cross-validation estimate

(c) Metric-based approach estimate (ADA)

(d) Proposed re-sampling approach estimate, 5 updates

(e) Proposed re-sampling approach estimate, 20 updates

Figure 8.2: Ground truth and estimated test error (z-axis) vs pairs of hyper-parameters for different model selection methods

selection approach of maximum marginal likelihood (also referred to as ML type II or evidence-based approach) [209] when possible. In order to be able to compute a marginal likelihood we use the Gaussian process co-regularization model (GPCR) [209] for any data set for which we use co-regularization. This model is a Bayesian probabilistic model and allows for approximate computation of the marginal likelihood, see [209] and [157] for details. We implemented this method with the "Gaussian Processes for Machine Learning Toolbox" version 3.1 [156]. For the Coil data set, we used the manifold co-regularization approach described in [179] to compute the kernels and used the GPCR method with the computed kernel matrices so that we could obtain marginal likelihoods. We denote the maximum marginal likelihood method as **MML**. Finally we denote the proposed Similar Data Sampling approach as **SDS**, and as mentioned we do not update the weights for each model - i.e., we use uniform weighting. Furthermore, we compare with the version of SDS that uses the average of predicted outputs for each model (Equation 8.3) as opposed to probabilistic outputs, which we denote **SDS-L**. Additionally for the Citeseer data set, since it is highly imbalanced and the lowest achievable test error is not close to 0, we use MCC as opposed to test error with the SDS methods for scoring models. However, for CV we still used test error, as using MCC causes significantly worse performance, due to having only a single test instance, i.e., using MCC with CV is not really an option for limited labeled data. In addition we test the combination of the state-of-the-art method ADA with our method and denote this combination **SDS+ADA**. This combination is accomplished by ranking the models with each selection method then adding the ranks to obtain new scores for each model - the model with the lowest score is then selected. There are two more baselines we provide as well. First, a non-semi-supervised learning approach using the Gaussian process classifier with both views if available. For the data sets we tested each view individually, stacking the views together to form a single view, and taking the average of classifiers trained on each view separately. Of the three, the averaging approach gave the best results, or not significantly different from the best, on all data sets, so we report these results. We denote this method as **GP - No SSL**. The final baseline reported is the result obtained when fixing the hyper-parameters across trials to the best set of fixed hyper-parameters from the grid of hyper-parameters (the hyper-parameter combination that gives the lowest test error averaged across all of the trials). This is the ideal result obtained if we had a model selection method capable of exactly determining the average performance of each model. We denote this method as **Best Fixed**. For each of the methods that use sampled sets (i.e., .632+, SDS, SDS+ADA, SDS-L, and SDS-LB) we sample 100 sets. The methods used are summarized in Table 8.2.

Table 8.2: Model selection methods used.

GP - No SSL	Gaussian process classifier [157] that does not do semi-supervised learning
CV	Leave-one-out cross-validation [85]
.632+	The .632+ bootstrap estimator [85]
MML	Maximum marginal likelihood approach [209]
ADA	State-of-the-art metric-based approach [169]
SDS	The proposed Similar Data Sampling approach
SDS+ADA	SDS combined with ADA by adding model ranks given by the two approaches to obtain new scores
SDS-L	SDS using the average of the predicted labels with each model (Equation 8.3) as the class conditional probability as opposed to averaging probabilistic outputs. Also excludes sampels with labeled data having only one class.
Best Fixed	The fixed model corresponding to the set of hyper-parameters that gave the lowest test error averaged across all trials.

We chose the multi-view semi-supervised learning algorithm that worked best for each data set, and these choices are shown in Table 8.1. For all methods, we use the same logistic loss model. We use logistic likelihood models in GPCR and in a Gaussian process classifier for the non-semi-supervised learning baseline. For the co-training algorithm we use L_1 regularized logistic regression classifiers as the base models.

The hyper-parameter grids used for model selection are as follows. GPCR has two hyper-parameters, σ_1 and σ_2 [209]. For the synthetic data σ_1 and σ_2 were varied on a grid of values $\{10^2, 10^1, \ldots, 10^{-5}\}$, resulting in 64 different models to choose from. For the Coil data set we follow the approach of [179] and vary σ_1 and σ_2 over $\{10^6, 10^4, 10^2, 10^0, 10^{-1}, 10^{-2}\}$, resulting in 36 models to choose from. For the co-training method, there are 3 hyper-parameters to select. The first is the ratio of the number of positive to number of negative estimated confident points to update the labeled set with at each iteration. We varied this ratio over {1:1, 1:2, 1:3, 1:4, 1:5}, as in all training sets the ratio for the labeled data indicates imbalance with fewer positive instances than negative. The other hyper-parameters are L_1 regularization hyper-parameters for view 1 and 2, λ_1 and λ_2, respectively. We varied these over $\{10^0, 10^{-1}, \ldots, 10^{-4}\}$. This resulted in 125 different models for the co-training method.

Since GPCR is transductive, we used a transductive approach for the experiments - that is for each trial, the data was randomly partitioned into num. labeled and num. unlabeled data instances as described in Table 8.1, and the unlabeled data is also used as the testing data for evaluating performance. We report results averaged over 100 random trials for each data set.

We report test error, Matthews Correlation Coefficient (MCC), and F1 Score for each data set, described below. Let tp denote the number of true positive predictions, fp the number of false positives, fn false negatives, and tn true negatives.

- Test error is given by: $\frac{fp+fn}{tp+tn+fp+fn}$.
- MCC is given by: $\frac{(tp)(tn)-(fp)(fn)}{\sqrt{(tp+fp)(tp+fn)(tn+fp)(tn+fn)}}$.
- F1 Score is given by: $\frac{2tp}{2tp+fn+fp}$.

Note that MCC and F1 score attain their best values at 1, and test error at 0. MCC are balanced performance measures, and MCC takes into account both false positive and false negative rates whereas F1 score does not take into account the false negative rate.

8.4.4 Experiment Results

The experimental results are summarized in Table 8.3. Additional significance testing is provided in Table 8.4, comparing the SDS method to other methods for each data set, in Table 8.5 for the SDS+ADA method, and in Table 8.6 for SDS-L. The testing is performed with respect to the test error for all data sets but the Citeseer data set, in which MCC is used instead as the data set is highly imbalanced and test error close to 0 is not achievable.

For the synthetic data, we found that the GPCR method was more sensitive to the hyper-parameters than the co-regularized logistic regression approach used in the preliminary study, which is likely part of the reason why most of the methods had higher variance and were farther from performing as well as the best fixed model. In order to take into account the sensitivity of the methods to the hyper-parameters for each data set, as well as how difficult the selection task

Table 8.3: Mean ± std. dev. of MCC, F1 score, and test error over 100 trials for each data set for the different model selection approaches, with best scores shown in bold. The data sets are ordered by increasing amount of labeled data.

		GP - No SSL	**CV**	**.632+**	**MML**	**ADA**	**SDS**	**SDS +ADA**	**SDS-L**	***Best Fixed***	***Frac. Close***
Synth (num. lab.=4)	Test Error	0.298 ±0.107	0.217 ±0.154	0.191 ±0.138	0.294 ±0.139	0.168 ±0.128	0.171 ±0.127	0.164 ±0.112	**0.040** ±0.050	0.030 ±0.014	0.078 (5/64)
	MCC	0.403 ±0.214	0.566 ±0.308	0.619 ±0.277	0.412 ±0.278	0.664 ±0.255	0.659 ±0.253	0.671 ±0.224	**0.921** ±0.100	0.941 ±0.027	
	F1 Score	0.701 ±0.107	0.783 ±0.154	0.809 ±0.139	0.706 ±0.139	0.832 ±0.128	0.829 ±0.127	0.835 ±0.113	**0.960** ±0.050	0.970 ±0.013	
WebKB (num. lab.=12)	Test Error	0.216 ±0.060	0.048 ±0.043	0.057 ±0.054	n/a	0.038 ±0.021	**0.028** ±0.017	0.029 ±0.008	0.031 ±0.008	0.017 ±0.003	0.352 (44/125)
	MCC	0.559 ±0.086	0.881 ±0.082	0.840 ±0.180	n/a	0.891 ±0.047	**0.917** ±0.061	0.914 ±0.024	0.909 ±0.026	0.950 ±0.010	
	F1 Score	0.651 ±0.065	0.905 ±0.069	0.860 ±0.189	n/a	0.911 ±0.040	**0.932** ±0.070	0.931 ±0.020	0.927 ±0.022	0.961 ±0.008	
Coil (num. lab.=20)	Test Error	0.410 ±0.009	0.075 ±0.047	0.081 ±0.057	0.095 ±0.066	**0.047** ±0.010	0.068 ±0.034	0.060 ±0.024	0.055 ±0.010	0.047 ±0.010	0.444 (16/36)
	MCC	0.225 ±0.028	0.859 ±0.082	0.848 ±0.100	0.823 ±0.116	**0.909** ±0.019	0.870 ±0.060	0.885 ±0.043	0.893 ±0.019	0.909 ±0.019	
	F1 Score	0.164 ±0.034	0.906 ±0.073	0.895 ±0.090	0.874 ±0.105	**0.945** ±0.012	0.916 ±0.049	0.928 ±0.033	0.935 ±0.013	0.945 ±0.012	
Citeseer (num. lab.=24)	Test Error	0.435 ±0.029	0.140 ±0.063	0.258 ±0.089	n/a	0.149 ±0.083	0.140 ±0.087	0.137 ±0.094	**0.135** ±0.090	0.117 ±0.070	0.184 (23/125)
	MCC	0.267 ±0.052	0.501 ±0.264	0.365 ±0.196	n/a	0.576 ±0.212	0.585 ±0.226	0.595 ±0.234	**0.605** ±0.213	0.602 ±0.240	
	F1 Score	0.423 ±0.027	0.556 ±0.271	0.456 ±0.202	n/a	0.659 ±0.164	0.664 ±0.180	0.674 ±0.183	**0.681** ±0.167	0.669 ±0.197	
Coil (num. lab.=40)	Test Error	0.375 ±0.010	0.033 ±0.016	0.031 ±0.016	**0.024** ±0.016	**0.024** ±0.016	0.035 ±0.018	0.030 ±0.017	0.031 ±0.014	0.024 ±0.016	0.500 (18/36)
	MCC	0.314 ±0.024	0.935 ±0.031	0.940 ±0.031	**0.954** ±0.031	**0.954** ±0.031	0.932 ±0.035	0.942 ±0.033	0.939 ±0.027	0.954 ±0.031	
	F1 Score	0.287 ±0.033	0.961 ±0.019	0.964 ±0.020	**0.973** ±0.019	**0.973** ±0.019	0.960 ±0.022	0.966 ±0.021	0.964 ±0.017	0.973 ±0.019	
Average	Test Error	0.347	0.103	0.123	n/a	0.085	0.088	0.084	**0.058**	0.047	n/a
	MCC	0.354	0.749	0.722	n/a	0.799	0.793	0.802	**0.853**	0.871	
	F1 Score	0.445	0.822	0.797	n/a	0.864	0.860	0.867	**0.893**	0.903	

Table 8.4: Significance testing results at the 5 percent level for paired t-tests between the proposed approach, SDS, and other model selection approaches for MCC on the Citeseer data set and test error on the rest. A “1” indicates a significant difference in means, “0” not significant, and a “+” indicates SDS did better, “-” worse.

	Synth	**WebKB**	**Coil (n=20)**	**Citeseer**	**Coil (n=40)**
GP- No SSL	+1	+1	+1	+1	+1
CV	+1	+1	0	+1	0
.632+	0	+1	0	+1	-1
MML	+1	n/a	+1	n/a	-1
ADA	0	+1	-1	0	-1

is for a given data set, we also report how many models out of the total number considered are close to the best model in the set of all models (i.e., the hyper-parameter grid), including the best. Specifically we report the fraction of models in the set considered that give test error within 0.025 of the Best Fixed Hyper-Parameters model. This corresponds to the last row of the table, with the

Table 8.5: Significance testing results at the 5 percent level for paired t-tests between the rank sum combined approach, SDS+ADA, and other model selection approaches for MCC on the Citeseer data set and test error on the rest. A "1" indicates a significant difference in means, "0" not significant, and a "+" indicates SDS+ADA did better, "-" worse.

	Synth	**WebKB**	**Coil (n=20)**	**Citeseer**	**Coil (n=40)**
GP- No SSL	+1	+1	+1	+1	+1
CV	+1	+1	+1	+1	+1
.632+	0	+1	+1	+1	0
MML	+1	n/a	+1	n/a	-1
ADA	0	+1	-1	0	-1
SDS	0	0	+1	0	+1

Table 8.6: Significance testing results at the 5 percent level for paired t-tests between SDS using label outputs, SDS-L, and other model selection approaches for MCC on the Citeseer data set and test error on the rest. A "1" indicates a significant difference in means, "0" not significant, and a "+" indicates SDS-L did better, "-" worse.

	Synth	**WebKB**	**Coil (n=20)**	**Citeseer**	**Coil (n=40)**
GP- No SSL	+1	+1	+1	+1	+1
CV	+1	+1	+1	+1	+1
.632+	+1	+1	+1	+1	0
MML	+1	n/a	+1	n/a	-1
ADA	+1	+1	-1	0	-1
SDS	+1	0	+1	0	+1
SDS+ADA	+1	0	0	0	0

entry **"Frac. Close"**.

Across the first four tasks, those with the smallest amount of labeled data, the SDS method either achieves comparable or significantly better performance than the other methods, and is only out-performed by ADA on the Coil data set. Cross-validation (CV), the .632+ bootstrap estimator (.632+), and maximum marginal likelihood (MML) clearly suffer performance deterioration for very small amounts of labeled data. ADA remained competitive, but SDS-L obtained better scores on three out of the four data sets, with ADA still giving the best results for Coil even when reducing the number of labeled data instances to 20, though this did narrow the gap between the two methods. The combination SDS+ADA sometimes offered an improvement over SDS and ADA, but this was usually not very significant. SDS-L had the best average performance, that is the performance averaged across all of the tasks (corresponding to the bottom row of Table 8.3). The most drastic difference is seen for the smallest amount of labeled data, i.e., for the Synth data SDS-L was able to attain mean test error of 0.040, close to the mean test error of the best single model, 0.030, as compared to 0.217 for CV, 0.294 for MML, and 0.168 for ADA.

Additionally, all of the model selection methods performed well on the Coil data set with 40 labeled instances - coming close to achieving the same performance as the best fixed model. This data set was particularly easy for model selection, which is also indicated to some extent by "Frac. Close" of 0.5 meaning half of the models to select from had performance close to the best fixed model.

A key observation is that using averaging with labels to estimate class probabilities (SDS-L), as opposed to probabilities (SDS) generally worked better, since even if most trained models agree on the labels exactly, the models themselves might output class probabilities close to 0.5, so that the generated data sets would still have high variation. This could cause the SDS method to perform more poorly when the probabilistic models are not well-calibrated, but offer some improvement when they are. Checking the average test errors computed by the SDS method for the Coil data set, we found they did not vary far from 0.5 (the minimum was 0.485 and the maximum 0.498), even though the majority of models actually had low test error. A similar issue occurred with the Synth data. SDS-L avoids this issue and also allows the similar data sampling approach to be used with models that do not have probabilistic outputs.

8.5 Conclusion and Future Work

We have proposed a new approach to model selection for semi-supervised learning algorithms, based on estimating performance by re-training and evaluating each model on many generated, similar data sets, which we called SDS (Similar Data Sampling) for short. In the experimental study on four data sets, we found the version of SDS using the average of label predictions to estimate conditional distributions (SDS-L) to improve over the widely used cross-validation approach and the Bayesian approach of maximum marginal (type II) likelihood, for smaller amounts of labeled data, i.e., for the tasks in our experiments with less than 40 labeled instances. We also compared with a state-of-the-art metric-based approach to semi-supervised model selection, ADA, which to our knowledge, has not yet been evaluated for the case of model selection for semi-supervised learning algorithms, and found SDS-L achieved better performance on three of the four data sets.

A key area of future work is to apply SDS to a broader range of learning scenarios where effective model selection methods are lacking. Indeed this unique broad applicability is a key advantage of SDS - this approach can be applied to scenarios where traditional model selection methods cannot, because complete data sets are sampled. A particular example of interest is the active learning scenario [170], in which an algorithm iteratively selects which instances to obtain information about from an oracle, e.g., labels for unlabeled data. If the active selection algorithm has tuning parameters that must be set, there is no way to do this with traditional model selection approaches since it requires estimating the performance of the selection strategy as labels are obtained for the unlabeled data. However this is easily accomplished with SDS as the entire active acquisition process can be simulated using the complete data sets generated (for each actual iteration). The main challenge for future work is extending the similar data sampling/generation approach to handle these different scenarios, including, for example, active view completion (see Chapter 7 for details on this scenario). Another key scenario for future work, which is also an open problem for model selection [78], is transfer learning. This line of future work involves applying the SDS strategy to transfer learning problems which can have no labeled target data at all.

Chapter 9

Conclusion and Future Work

The conclusion is broken into three parts, corresponding to the three contributions of this thesis on multi-view semi-supervised learning (MVSSL). After, key future work is discussed.

9.1 Conclusions

For the first part of the thesis on generating data for missing views, there were two main lines of previous work: using artificially generated view 2 data only and only using the available complete view data for MVSSL algorithms. We found that these two lines of previous work were limited. The first method created data to approximately match conditions for the success of MVSSL methods, but failed to use the real data, which actually fulfilled these conditions, when available. The second method failed to make effective use of the unlabeled data when both views were not present, and so performed poorly with limited complete data. We were able to achieve the best results in our experiments with the proposed CoNet method, an approach that is a hybrid of both cases - both generating view 2 data and utilizing available view 2 data through learning a biased view mapping / feature generation function. Our method essentially finds a compromise, by using the available view 2 data which is more reliable than artificially generated view 2 data, while also generating useful view 2 data for that which is missing. Furthermore this generated data is made to match the view data as more data becomes available. We found that at first, when few instances were complete with view 2 data, the generated data was most useful and offered the most benefit over using the limited view 2 data. As more view 2 data became available, matching and using this data became more important. This work allowed the gap between these two approaches for addressing missing view data for multi-view semi-supervised learning to be bridged, and is a key step toward making the many effective mutli-view semi-supervised learning methods more applicable to real-world data that often is not complete for all instances. This in turn is a step in one direction for improving the estimation of predictive models for low-quality data scenarios.

The key conclusion from the second part of the thesis is on the dependence of the benefit of an active view completion strategy over the passive (random) one on the relationship between the views. We demonstrated how the benefit of the active strategy is dependent on the dependence between views, i.e., if the views are conditionally independent given class labels, the passive selection strategy can be just as, if not more, effective than an active one. This work points out a previously overlooked issue for active view completion - that an active approach may not always help in this scenario, and the characteristics of the data should be taken into account. A key observation resulting from this work is that the case when an active strategy appears to offer the most benefit also corresponds to the case where MVSSL is harder due to less expansion between views, and thus may also require more complete unlabeled data to attain low test error. Though we did also demonstrate the benefit of our proposed active approach for a few data sets, an additional conclusion is that even when an active approach can offer a benefit over a random one, the best active approach can depend on the data. This highlights the potential of this area for future work, as different data and tasks may require different active selection approaches, as well as the potential of active view completion as a solution for addressing real world MVSSL scenarios with missing view data.

Finally, the third component of this thesis, on model selection for semi-supervised learning, provides a first step toward making model selection possible for limited labeled data scenarios, an open problem that limits the applicability of semi-supervised learning methods. By addressing this issue, this work makes it possible to apply multi-view semi-supervised learning methods, and semi-supervised learning methods in general, to real learning tasks with limited labeled data. In addition we conclude that there is still much work to be done in this area of research. Even though the proposed approaches using Similar Data Sampling (SDS) were able to improve over existing methods for small amounts of labeled there was still a gap in performance for some data sets between the model selection approaches and the best fixed model. Furthermore, there are many related machine learning scenarios growing in prevalence, such as transfer learning, for which model selection has barely been considered, if at all, so these areas are ripe for future work on model selection.

9.2 Future Work

Overall this thesis is just the beginning of the work needed for multi-view semi-supervised learning with missing views, and more generally learning with low-quality data and related areas of research. Below are some key directions of future work.

One key future direction is to incorporate transfer learning with multi-view semi-supervised learning, as well as handling low-quality data issues such as missing view data, for example, extending CoNet so that it can be used for pre-processing for knowledge transfer as well. While there is recent work on the combination of multi-view semi-supervised learning with transfer learning [212], the conjunction with additional low-quality data aspects such as missing views has yet to be considered. More generally, a key direction of future work is to provide comprehensive low-quality learning approaches capable of addressing all low-quality data aspects, and, as a key goal, incorporating all potentially beneficial and diverse sources of information. Already there has been some work suggesting comprehensive consideration of low-quality data aspects can be most beneficial [17]. Exploring such approaches can also lead to new lines of work similar to that of feature selection as these approaches could suffer from information overload or useless information. Therefore another direction in addition to information fusion is information *selection*.

Another key area of future work that is also related to transfer learning is model selection for transfer learning. Model selection becomes even more important in the typical transfer learning setting as there is often no labeled data at all from the target domain. In this case standard model selection methods are generally not applicable - they result in fitting models to the source domain, and since this data comes from a different distribution than the target data distribution, the source-fitted models usually do not work well in the target domain. While transfer learning methods are designed to address the change in distribution, they also generally have hyper-parameters that must be set for them to work, so that model selection is still a key issue. Since transfer learning with little or no target data labels is a type of semi-supervised learning, the SDS approach of this thesis is applicable and may offer a good starting point for model selection for transfer learning. In this case, however, the sampling approach should be modified to maintain the difference in source and target data distributions for the sampled data sets.

In general, while it is true that when enough labeled data is available standard model selection methods like cross-validation will work well, since many applications continue to arise involving low-quality data having very few or no labeled data, model selection in this scenario is an important area of future study. While this thesis provides a starting point, future work, in addition to improving the approach proposed here and providing more theoretical study, involves extending this approach to other scenarios including transfer learning and active learning. As one example, this approach could be applied to, as in the preliminary study, the case of small labeled sample data with structured feature relationships, or one-class learning where there is no negative class examples, again posing an issue for model selection. One direction for this extension to scenarios without a large pool of unlabeled data to sample from is to instead sample from other approximations to the data distribution - in particular to approximate the marginal distribution of the data in addition to the label (conditional) distribution, and sample from both. In general this type of sampling approach may be possible to extend to all such scenarios where the ground truth has partial or limited representation in the available data. Therefore a key direction is to generalize the method to a more abstract scenario applicable to a variety of specific learning scenarios, and testing the approach in these different areas.

In addition to model selection for active learning, there is much future work possible for the case of active view completion. This includes analyzing different types of missing view data scenarios (e.g., missing entire views or not at random), considering the cost-sensitive combination of different types of active learning (e.g., active label acquisition combined with active view completion), and deriving special selection strategies for specific data sets and tasks.

References

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

[2] Amini, M., Usunier, N., & Goutte, C. (2009). Learning from multiple partially observed views–an application to multilingual text categorization. *Advances in neural information processing systems*, 23.

[3] Ando, R. & Zhang, T. (2007a). Learning on Graph with Laplacian Regularization. In *Advances in Neural Information Processing Systems: Proceedings of the 2006 Conference*: MIT Press.

[4] Ando, R. K., Zhang, T., & Bartlett, P. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6, 1817–1853.

[5] Ando, R. R. & Zhang, T. (2007b). Two-view feature generation model for semi-supervised learning. In *Proceedings of the 24th international conference on Machine learning* (pp. 25–32).: ACM.

[6] Arnold, A., Nallapati, R., & Cohen, W. W. (2008). Exploiting feature hierarchy for transfer learning in named entity recognition. In *46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL:HLT)*.

[7] Atrey, P., Hossain, M., El Saddik, A., & Kankanhalli, M. (2010). Multimodal fusion for multimedia analysis: a survey. *Multimedia Systems*, 16, 345–379.

[8] Balcan, M.-f. & Blum, A. (2005). A pac-style model for learning from labeled and unlabeled data. In *In Proceedings of the 18th Annual Conference on Computational Learning Theory* (pp. 111–126).

[9] Balcan, M. F., Blum, A., & Yang, K. (2005). Co-training and expansion: Towards bridging theory and practice. *Advances in neural information processing systems*, (pp. 89–96).

[10] Balcan, M. F., Hanneke, S., & Vaughan, J. W. (2010). The true sample complexity of active learning. *Machine learning*, 80(2-3), 111–139.

[11] Barnard, K., Duygulu, P., Forsyth, D., De Freitas, N., Blei, D., & Jordan, M. (2003). Matching words and pictures. *The Journal of Machine Learning Research*, 3, 1107–1135.

[12] Belkin, M., Niyogi, P., & Sindhwani, V. (2006a). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 7, 2399–2434.

[13] Belkin, M., Niyogi, P., Sindhwani, V., & Bartlett, P. (2006b). Manifold regularization: A geometric framework for learning from examples. *Journal of Machine Learning Research*, 7, 2399–2434.

[14] Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. (2010). A theory of learning from different domains. *Machine Learning*, 79, 151–175.

[15] Ben-David, S., Blitzer, J., Crammer, K., & Pereira, F. (2007). Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems 20* Cambridge, MA: MIT Press.

[16] Bengio, Y. & Chapados, N. (2003). Extensions to metric based model selection. *The Journal of Machine Learning Research*, 3, 1209–1227.

[17] Berti-Equille, L., Dasu, T., & Srivastava, D. (2011). Discovery of complex glitch patterns:

A novel approach to quantitative data cleaning. In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on* (pp. 733–744).: IEEE.

[18] Bicke, S., Sawade, C., , & Scheffer, T. (2008). Transfer learning by distribution matching for targeted advertising. In *Proceedings of the Advances in Neural Information Processing Systems*.

[19] Bickel, S. (2006). Ecml-pkdd discovery challenge 2006 overview. In *Proc. ECML/PKDD Discovery Challenge Workshop*.

[20] Bickel, S., Brückner, M., & Scheffer, T. (2007). Discriminative learning for differing training and test distributions. In *Proc. of the 24th Int. Conf. on Machine Learning (ICML)* (pp. 81–88).

[21] Bickel, S. & Scheffer, T. (2004). Multi-view clustering. In *Proceedings of the IEEE international conference on data mining*.

[22] Bickel, S. & Scheffer, T. (2005). Estimation of mixture models using co-em. In *Proceedings of the European Conference on Machine Learning* (pp. 35–46).: Springer.

[23] Blaschko, M. B. & Lampert, C. H. (2008). Correlational spectral clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[24] Blei, D. & Jordan, M. (2003). Modeling annotated data. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval* (pp. 127–134).: ACM.

[25] Blum, A. & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory* (pp. 92–100).: ACM.

[26] Botev, Z., Grotowski, J., & Kroese, D. (2010). Kernel density estimation via diffusion. *The Annals of Statistics*, 38(5), 2916–2957.

[27] Bradley, D. & Bagnell, J. (2008). Differentiable sparse coding. In *Neural Information Processing Systems* (pp. 113–120).

[28] Brefeld, U., Büscher, C., & Scheffer, T. (2005). Multi-view discriminative sequential learning. *Proceedings of the European Conference on Machine Learning*, (pp. 60–71).

[29] Brefeld, U., Gärtner, T., Scheffer, T., & Wrobel, S. (2006). Efficient co-regularised least squares regression. In *Proceedings of the 23rd international conference on Machine learning* (pp. 137–144).

[30] Brefeld, U. & Scheffer, T. (2004). Co-EM support vector learning. In *Proceedings of the twenty-first international conference on Machine learning*: ACM.

[31] Candes, E. & Plan, Y. (2009). Matrix completion with noise. *Proceedings of the IEEE (submitted)*.

[Carlson et al.] Carlson, A., Cumby, C., Rosen, J., Rizzolo, N., & Roth, D. The snow learning architecture. *Software available at* `http://l2r.cs.uiuc.edu/~cogcomp/asoftware.php?skey=SNOW`.

[33] Cawley, G. & Talbot, N. (2006). Gene selection in cancer classification using sparse logistic regression with Bayesian regularization. *Bioinformatics*, 22(19), 2348.

[34] Cessie, S. L. & Houwelingen, J. C. V. (1992). Ridge estimators in logistic regression. *Applied Statistics*, 41(1), 191–201.

[35] Chang, C. & Lin, C. (2001a). Libsvm: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

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

[37] Chapelle, O., Schölkopf, B., & Zien, A. (2006a). *Semi-supervised learning*. MIT press Cambridge, MA.

[38] Chapelle, O., Schölkopf, B., & Zien, A., Eds. (2006b). *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

[39] Chapelle, O. & Zien, A. (2005). Semi-supervised classification by low density separation. In *Proceedings of the tenth international workshop on artificial intelligence and statistics*, volume 2005: Citeseer.

[40] Chaudhuri, K., Kakade, S. M., Livescu, K., & Sridharan, K. (2009). Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th annual international conference on machine learning* (pp. 129–136).: ACM.

[41] Chelba, C. & Acero, A. (2006). Adaptation of maximum entropy capitalizer: Little data can help a lot. *Computer Speech and Language*, 20(4), 382–399.

[42] Chen, J., Ji, S., Ceran, B., Li, Q., Wu, M., & Ye, J. (2008). Learning subspace kernels for classification. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 106–114).: ACM.

[43] Chen, M., Weinberger, K., & Chen, Y. (2011). Automatic feature decomposition for single view co-training. In L. Getoor & T. Scheffer (Eds.), *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11 (pp. 953–960). New York, NY, USA: ACM.

[44] Chen, N., Zhu, J., & Xing, E. P. (2010). Predictive subspace learning for multi-view data: a large margin approach. In *Advances in neural information processing systems 24*.

[45] Chin, K., DeVries, S., Fridlyand, J., Spellman, P., Roydasgupta, R., Kuo, W., Lapuk, A., Neve, R., Qian, Z., Ryder, T., Chen, F., Feiler, H., Tokuyasu, T., Kingsley, C., Dairkee, S., Meng, Z., Chew, K., Pinkel, D., Jain, A., Ljung, B., Esserman, L., Albertson, D., Waldman, F., & Gray, J. (2006). Genomic and transcriptional aberrations linked to breast cancer pathophysiologies. *Cancer Cell*, 10(6), 529–541.

[46] Christoudias, C. (2009). *Probabilistic models for multi-view semi-supervised learning and coding*. PhD thesis, Massachusetts Institute of Technology.

[47] Christoudias, C. M., Urtasun, R., & Darrell, T. (2008). Multi-view learning in the presence of view disagreement. In *Proceedings of the 24th conference on Uncertainty in Artifical Intelligence*.

[48] Cleuziou, G., Exbrayat, M., Martin, L., & Sublemontier, J. (2009). Cofkm: a centralized method for multiple-view clustering. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on* (pp. 752–757).: IEEE.

[49] Coates, A., Lee, H., & Ng, A. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[50] Collins, M. & Singer, Y. (1999). Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora* (pp. 189–196).

[51] Collobert, R. & Bengio, S. (2001). Svmtorch: Support vector machines for large-scale regression problems. *Journal of Machine Learning Research*, 21.

[52] Collobert, R., Sinz, F., Weston, J., Bottou, L., & Joachims, T. (2006). Large scale transductive svms. *Journal of Machine Learning Research*, 7.

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

[54] Culp, M., Michailidis, G., & Johnson, K. (2009). On multi-view learning with additive models. *The Annals of Applied Statistics*, 3(1), 292–318.

[55] Dai, W., Jin, O., Xue, G., Yang, Q., & Yu, Y. (2009). Eigentransfer: a unified framework for transfer learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*: ACM New York, NY, USA.

[56] Dai, W., Xue, G.-R., Yang, Q., & Yu, Y. (2007a). Co-clustering based classification for out-of-domain documents. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*.

[57] Dai, W., Yang, Q., Xue, G., & Yu, Y. (2007b). Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning* (pp. 200).: ACM.

[58] Dasgupta, S., Littman, M. L., & McAllester, D. (2002). PAC generalization bounds for co-training. In *Advances in neural information processing systems* (pp. 375–382).: MIT Press.

[59] Daumeé III, H. & Marcu, D. (2006). Domain adaptation for statistical classifers. *Journal of Artificial Intelligence Research*, 26, 101–126.

[60] de Sa, V., Gallagher, P., Lewis, J., & Malave, V. (2010). Multi-view kernel construction. *Machine Learning*, 79, 47–71.

[61] de Sa, V. R. (2005). Spectral clustering with two views. In *ICML workshop on learning with multiple views*.

[62] Dileep, A. D. & Sekhar, C. C. (2009). Representation and feature selection using multiple kernel learning. *International Joint Conference on Neural Networks*, (pp. 717–722).

[63] Elhamifar, E. & Vidal, R. (2009). Sparse subspace clustering. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2790–2797).

[64] Erhan, D., Bengio, Y., Courville, A., Manzagol, P., Vincent, P., & Bengio, S. (2010). Why does unsupervised pre-training help deep learning? *The Journal of Machine Learning Research*, 11, 625–660.

[65] Farquhar, J. D. R., Hardoon, D., Meng, H., Shawe-Taylor, J., & Szedmak, S. (2006). Two view learning: SVM-2K, theory and practice. *Advances in neural information processing systems*, 18, 355.

[66] Fei, H., Quanz, B., & Huan, J. (2010). Regularization and feature selection for networked features. In *Proceedings of the 19th ACM international conference on Information and knowledge management* (pp. 1893–1896).: ACM.

[67] Fort, G. & Lambert-Lacroix, S. (2005). Classification using partial least squares with penalized logistic regression. *Bioinformatics*, 21(7), 1104–1111.

[68] Ganchev, K., Graca, J., Blitzer, J., & Taskar, B. (2008). Multi-view learning over structured and non-identical outputs. In *Proceedings of The 24th Conference on Uncertainty in Artificial Intelligence*.

[69] Gao, J., Fan, W., Jiang, J., & Han, J. (2008). Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD conference on Knowledge Discovery and Data Mining*.

[Genkin et al.] Genkin, A., Lewis, D. D., & Madigan, D. *BBR: Bayesian Logistic Regression Software*. Software available at `http://www.stat.rutgers.edu/~madigan/BBR/`.

[71] Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*.

[72] Goeman, J., van de Geer, S., de Kort, F., & van Houwelingen, H. (2004). A global test for groups of genes: testing association with a clinical outcome.

[73] Goldman, S. & Zhou, Y. (2000). Enhancing supervised learning with unlabeled data. In *Proceedings of the 17th International Conference on Machine Learning*.

[74] Grant, M. & Boyd, S. (2008a). *CVX: Matlab software for disciplined convex programming*. Web page and software available at `http://stanford.edu/~boyd/cvx`.

[75] Grant, M. & Boyd, S. (2008b). Graph implementations for nonsmooth convex programs. *Lecture Notes in Control and Information Sciences*, 371, 95–110.

[76] Gretton, A., Borgwardt, K. M., Rasch, M., Scholkopf, B., & Smola, A. J. (2007). A kernel method for the two-sample-problem. In *Advances in NIPS 19*: MIT Press.

[77] Gupta, S. K., Phung, D., Adams, B., Tran, T., & Venkatesh, S. (2010). Nonnegative shared subspace learning and its application to social media retrieval. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10 (pp. 1169–1178). New York, NY, USA: ACM.

[78] Guyon, I., Saffari, A., Dror, G., & Cawley, G. (2010). Model selection: Beyond the bayesian/frequentist divide. *The Journal of Machine Learning Research*, 11, 61–87.

[79] H., W. W. Z. & Zhou (2008). On multi-view active learning and the combination with semi-supervised learning. In *Proceedings of the 25th international conference on Machine learning* (pp. 1152–1159).: ACM.

[80] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. (2009). The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1), 10–18.

[81] Hanneke, S. (2009). *Theoretical Foundations of Active Learning*. PhD thesis, Carnegie Mellon University.

[82] Hardoon, D. R., Szedmak, S., & Shawe-Taylor, J. (2004). Canonical correlation analysis: an overview with application to learning methods. *Neural Computation*, 16(12), 2639–2664.

[83] Harel, M. & Mannor, S. (2011). Learning from multiple outlooks. In L. Getoor & T. Scheffer (Eds.), *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11 (pp. 401–408). New York, NY, USA: ACM.

[84] Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag.

[85] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag.

[86] Haussler, D. (1999). Convolution kernels on discrete structures. *Technical Report UCSC-CRL099-10, Computer Science Department, UC Santa Cruz.*

[87] He, J. & Lawrence, R. (2011). A graph-based framework for multi-task multi-view learning. In L. Getoor & T. Scheffer (Eds.), *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11 (pp. 25–32). New York, NY, USA: ACM.

[88] Hinton, G., Osindero, S., & Teh, Y. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527–1554.

[89] Hinton, G. & Salakhutdinov, R. (2006a). Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504 – 507.

[90] Hinton, G. & Salakhutdinov, R. (2006b). Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504.

[91] Hoerl, A. E. & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1), 55–67.

[92] Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, 28(3/4), 321–377.

[93] Hou, C., Zhang, C., Wu, Y., & Nie, F. (2010). Multiple view semi-supervised dimensionality reduction. *Pattern Recognition*, 43(3), 720 – 730.

[94] Hsieh, C., Chang, K., Lin, C., Keerthi, S., & Sundararajan, S. (2008). A Dual Coordinate Descent Method for Large-scale Linear SVM. In *Proceedings of the Twenty Fifth International Conference on Machine Learning (ICML).*

[95] 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)* (pp. 549–552).

[96] Huang, J., Smola, A., Gretton, A., Borgwardt, K. M., & Schölkopf, B. (2006). Correcting sample selection bias by unlabeled data. In *Proceedings of Twentieth Annual Conference on Neural Information Processing Systems*.

[97] Iyengar, G. & Nock, H. (2003). Discriminative model fusion for semantic concept detection and annotation in video. In *Proceedings of the eleventh ACM international conference on Multimedia* (pp. 255–258).: ACM.

[98] Jacob, L., Hoffmann, B., Stoven, V., & Vert, J.-P. (2008). *Virtual screening of GPCRs: an in silico chemogenomics approach*. Technical Report HAL-00220396, French Center for Computational Biology.

[99] JL, F., M, M., S, M., K, S., & R., S. (2007). Genome scale enzyme-metabolite and drug-target interaction predictions using the signature molecular descriptor. *Bioinformatics*, 24(2), 225–33.

[100] Joachims, T. (1999). Transductive inference for text classification using support vector machines. In I. Bratko & S. Dzeroski (Eds.), *Proceedings of ICML-99, 16th International Conference on Machine Learning* (pp. 200–209).: Morgan Kaufmann Publishers.

[101] Joachims, T. (2006). Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 217–226).: ACM New York, NY, USA.

[102] Joachims, T., Cristianini, N., & Shawe-Taylor, J. (2001). Composite kernels for hypertext categorisation. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-* (pp. 250–257).: Citeseer.

[103] Judson, R., Houck, K., Kavlock, R., Knudsen, T., Martin, M., Mortensen, H., Reif, D., Rotroff, D., Shah, I., Richard, A., et al. (2010). In Vitro Screening of Environmental Chemicals for Targeted Testing Prioritization: The ToxCast Project. *Environmental health perspectives*, 118(4), 485–492.

[104] Kaariainen, M. (2006). Semi-supervised model selection based on cross-validation. In *International Joint Conference on Neural Networks* (pp. 1894 –1899).

[105] Kakade, S. & Foster, D. (2007a). Multi-view regression via canonical correlation analysis. In *Proceedings of the 20th annual conference on Learning theory* (pp. 82–96).: Springer-Verlag.

[106] Kakade, S. & Foster, D. (2007b). Multi-view regression via canonical correlation analysis. In *Learning Theory*, Lecture Notes in Computer Science (pp. 82–96).

[107] Kanehisa, M. & Goto, S. (2000). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28(1), 27.

[108] Kang, Y. & Choi, S. (2011). Restricted deep belief networks for multi-view learning. In *Proceedings of the ECML/PKDD 2011*.

[109] Kim, S. & Xing, E. P. (2008). Structured feature selection in high-dimensional space via block regularized regression. In *Proceedings of the 24th International Conference on Conference on Uncertainty in Artificial Intelligence*.

[110] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the fourteenth international joint conference on artificial intelligence* (pp. 1137–1143).: Morgan Kaufmann.

[111] Krishnapuram, B., Williams, D., Xue, Y., Hartemink, A., Carin, L., & Figueiredo, M. (2004). On semi-supervised classification. *Advances in neural information processing systems*, 17, 721–728.

[112] Lanckriet, G., De Bie, T., Cristianini, N., Jordan, M., & Noble, W. (2004). A statistical framework for genomic data fusion. *Bioinformatics*, 20(16), 2626.

[113] Lange, T., Braun, M., Roth, V., & Buhmann, J. (2002). Stability-based model selection. In *In Advances in Neural Information Processing Systems*.

[114] Larochelle, H., Erhan, D., & Bengio, Y. (2008). Zero-data learning of new tasks. In *AAAI*.

[115] Lawrence, N. (2004). Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in neural information processing systems 16*, volume 16 (pp. 329).: The MIT Press.

[116] Lee, H., Battle, A., Raina, R., & Ng, A. (2007). Efficient sparse coding algorithms. *Advances in neural information processing systems*, 19, 801.

[117] Lee, H., Raina, R., Teichman, A., & Ng, A. Y. (2009). Exponential family sparse coding with applications to self-taught learning. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence*.

[118] Li, G., Hoi, S. C. H., & Chang, K. (2010). Two-view transductive support vector machines. In *Proceedings of the SIAM International Conference on Data Mining*.

[119] Liang, L., Mandal, V., Lu, Y., & Kumar, D. (2008). Mcm-test: a fuzzy-set-theory-based approach to differential analysis of gene pathways. *BMC Bioinformatics*, 9(Suppl 6), S16.

[120] Liao, J. & Chin, K. (2007). Logistic regression for disease classification using microarray data: model selection in a large p and small n case. *Bioinformatics*, 23(15), 1945.

[121] Liao, X., Li, H., & Carin, L. (2007). Quadratically gated mixture of experts for incomplete data classification. In *Proceedings of the 24th International Conference on Machine learning*.

[122] Ling, X., Dai, W., Xue, G., Yang, Q., & Yu, Y. (2008). Spectral domain-transfer learning. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*: ACM New York, NY, USA.

[123] Long, B., Yu, P. S., & Zhang, Z. M. (2008). A general model for multiple view unsupervised learning. In *Proceedings of the 8th SIAM International Conference on Data Mining*.

[124] Madani, O., Pennock, D., & Flake, G. (2004). Co-validation: Using model disagreement on unlabeled data to validate classification algorithms. In *Proceedings of NIPS*: Citeseer.

[125] Marlin, B. M. (2008). *Missing data problems in machine learning*. PhD thesis, University of Toronto.

[126] Melville, P., Saar-Tsechansky, M., Provost, F., & Mooney, R. (2004). Active feature-value acquisition for classifier induction. In *IEEE International Conference on Data Mining* (pp. 483–486).: IEEE.

[127] Mercier, G., Berthault, N., Mary, J., Peyre, J., Antoniadis, A., Comet, J.-P., Cornuejols, A., Froidevaux, C., , & Dutreix, M. (2004). Biological detection of low radiation doses by combining results of two microarray analysis methods. *Nucleic Acids Research*, 32(1), e12.

[128] Mootha, V., Lindgren, C., Eriksson, K., Subramanian, A., Sihag, S., Lehar, J., Puigserver, P., Carlsson, E., Ridderstraale, M., Laurila, E., et al. (2003). PGC-1 α-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature Genetics*, 34(3), 267–273.

[129] Müller, C., Rapp, S., & Strube, M. (2002). Applying co-training to reference resolution. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (pp. 352–359).: Association for Computational Linguistics.

[130] Muslea, I. (2002). *Active learning with multiple views*. PhD thesis, University of Southern California.

[131] Muslea, I., Minton, S., & Knoblock, C. A. (2000). Selective sampling with redundant views. In *Proceedings of the fifteenth National Conference on Artificial Intelligence AAAI* (pp. 621–626).

[132] Muslea, I., Minton, S., & Knoblock, C. A. (2002a). Active + semi-supervised learning = robust multi-view learning. In *Proceedings of the 19th International Conference on Machine Learning* (pp. 435–442).

[133] Muslea, I., Minton, S., & Knoblock, C. A. (2002b). Adaptive view validation: A first step towards automatic view detection. In *Proceedings of the 19th International Conference on Machine Learning* (pp. 443–450).

[134] Muslea, I., Minton, S., & Knoblock, C. A. (2006). Active learning with multiple views. *Journal of Artificial Intelligence Research*, 27(1), 203–233.

[135] Muslea, I., Minton, S. N., & Knoblock, C. A. (2003). Active learning with strong and weak views: A case study on wrapper induction. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence* (pp. 415–420).

[136] Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. (2011). Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11 (pp. 689–696).: ACM.

[137] Nigam, K. & Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management* (pp. 86–93).

[138] Okuno, Y., Yang, J., Taneishi, K., Yabuuchi, H., & Tsujimoto, G. (2006(9)). GLIDA: GPCR-ligand database for chemical genomic drug discovery. *Nucleic Acids Res.*

[139] Olshausen, B. A. & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583), 607–609.

[140] Pan, S. J., Kwok, J. T., & YangPan, Q. (2008). Transfer learning via dimensionality reduction. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence* (pp. 677–682).

[141] Pan, S. J., Tsang, I. W., Kwok, J. T., & Yang, Q. (2011). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2), 199–210.

[142] Pan, S. J. & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345–1359.

[143] Pang, H., Lin, A., Holford, M., Enerson, B., Lu, B., Lawton, M., Floyd, E., & Zhao, H. (2006). Pathway analysis using random forests classification and regression. *Bioinformatics*, 22(16), 2028.

[144] Park, M. Y. & Hastie, T. (2008). Penalized logistic regression for detecting gene interactions. *Biostatistics*, 9(1), 30–50.

[145] Pérez-Cruz, F. (2008). Estimation of information theoretic measures for continuous random variables. In *Advances in Neural Information Processing Systems 21* (pp. 1257–1264).

[146] Poon, H. & Domingos, P. (2011). Sum-product networks: A new deep architecture. In *Proc. 12th Conf. on Uncertainty in Artificial Intelligence* (pp. 337–346).

[147] Quanz, B. & Huan, J. (2009a). Aligned graph classification with regularized logistic regression. In *Proc. 2009 SIAM International Conference on Data Mining*.

[148] Quanz, B. & Huan, J. (2009b). Large margin transductive transfer learning. In *Proceeding of the 18th ACM conference on Information and knowledge management* (pp. 1327–1336).: ACM.

[149] Quanz, B., Huan, J., & Mishra, M. (2011). Knowledge transfer with low-quality data: a feature extraction issue. In *Proceeding of the 27th International Conference on Data Engineering*.

[150] Quanz, B., huan, J., & Mishra, M. (2012). Knowledge transfer with low-quality data: a feature extraction issue. *IEEE Transactions on Knowledge and Data Engineering*, Accepted.

[151] Quanz, B. & Tsatsoulis, C. (2008). Determining object safety using a multiagent, collaborative system. In *Environment-Mediated Coordination in Self-Organizing and Self-Adaptive Systems (ECOSOA 2008) Workshop* Venice, Italy.

[152] Raina, R., Battle, A., Lee, H., Packer, B., & Ng, A. Y. (2007). Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning* (pp. 759–766). New York, NY, USA.

[153] Ranzato, M., Susskind, J., Mnih, V., & Hinton, G. (2011). On deep generative models with applications to recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (pp. 2857–2864).: IEEE.

[154] Rapaport, F., Zinovyev, A., Dutreix, M., Barillot, E., & Vert, J.-P. (2007). Classification of microarray data using gene networks. *BMC Bioinformatics*, 8, R35.

[155] Raskutti, B., Ferrá, H., & Kowalczyk, A. (2002). Combining clustering and co-training to enhance text classification using unlabelled data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 620–625).: ACM.

[156] Rasmussen, C. E. & Nickisch, H. (2010). GPML: Gaussian processes for machine learning toolbox. `http://www.gaussianprocess.org/gpml/code/matlab/doc/`.

[157] Rasmussen, C. E. & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. MIT Press.

[158] Rosenberg, D. & Bartlett, P. (2007). The rademacher complexity of co-regularized kernel classes. In *Proceedings of Artificial Intelligence & Statistics*.

[159] Rosenberg, D., Sindhwani, V., Bartlett, P., & Niyogi, P. (2009). Multi-view point cloud kernels for semisupervised learning. *Signal Processing Magazine, IEEE*, 26(5), 145 –150.

[160] Rosenberg, D. S. (2008). *Semi-supervised learning with multiple views*. PhD thesis, University of California, Berkely.

[161] Saar-Tsechansky, M., Melville, P., & Provost, F. (2009). Active feature-value acquisition. *Management Science*, 55(4), 664–684.

[162] Salakhutdinov, R. & Hinton, G. (2009). Deep Boltzmann machines. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 5 (pp. 448–455).

[163] Salzmann, M., Ek, C., Urtasun, R., & Darrell, T. (2010). Factorized orthogonal latent spaces. In *International Conference on Artificial Intelligence and Statistics, Sardinia, Italy*.

[164] Sarkar, A. (2001). Applying co-training methods to statistical parsing. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies* (pp. 1–8).: Association for Computational Linguistics.

[165] Satpal, S. & Sarawagi, S. (2007). Domain adaptation of conditional probability models via feature subsetting. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*.

[166] Schölkopf, B., Smola, A. J., & Müller, K.-R. (1999). Kernel principal component analysis. *Advances in kernel methods: support vector learning*, (pp. 327–352).

[167] Schuurmans, D. (1997). A new metric-based approach to model selection. In *In Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)* (pp. 552–558).

[168] Schuurmans, D. & Southey, F. (2002). Metric-based methods for adaptive model selection and regularization. *Machine Learning*, 48(1), 51–84.

[169] Schuurmans, D., Southey, F., Wilkinson, D., & Guo, Y. (2006). Metric-based approaches for semi-supervised regression and classification. In O. Chapelle, B. Schölkopf, & A. Zien (Eds.), *Semi-Supervised Learning* (pp. 421–451). The MIT Press.

[170] Settles, B. (2010). *Active Learning Literature Survey*. Technical Report 1648, Department of Computer Science, University of Wisconsin-Madison.

[171] Settles, B. (2011). From theories to queries: Active learning in practice. *JMLR Workshop and Conference Proceedings*, (pp. 1–18).

[172] Shawe-Taylor, J. & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge University Press.

[173] Shevade, S. & Keerthi, S. (2003). A simple and efficient algorithm for gene selection using sparse logistic regression.

[174] Shi, J. & Malik, J. (2000). Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (pp. 888–905).

[175] Shimodaira, H. (2000). Improving predictive inference under convariance shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(18), 227–244.

[176] Sindhwani, V., Chu, W., & Keerthi, S. (2007). Semi-supervised gaussian process classifiers. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence* (pp. 1059–1064).

[177] Sindhwani, V., Niyogi, P., & Belkin, M. (2005a). Beyond the point cloud: from transductive to semi-supervised learning. In *Proceedings of the 22nd international conference on Machine learning* (pp. 824–831).: ACM.

[178] Sindhwani, V., Niyogi, P., & Belkin, M. (2005b). A co-regularization approach to semi-supervised learning with multiple views. In *Workshop on Learning with Multiple Views, International Conference on Machine Learning*.

[179] 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* (pp. 976–983).: ACM.

[180] Smalter, A., Huan, J., & Lushington, G. (2008). Structure-based pattern mining for chemical compound classification. In *Proceedings of the 6th Asia Pacific Bioinformatics Conference*.

[181] Smalter Hall, A. (2011). *Genome-wide Protein-chemical Interaction Prediction*. PhD thesis, University of Kansas.

[182] Snoek, C., Worring, M., & Smeulders, A. (2005). Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM International Conference on Multimedia* (pp. 399–402).: ACM.

[183] Sokolov, A. & Ben-Hur, A. (2011). Multi-view prediction of protein function. In *ACM Conference on Bioinformatics, Computational Biology and Biomedicine*.

[184] Sridharan, K. & Kakade, S. M. (2008). An information theoretic framework for multi-view learning. In *Proceedings of the 21st Annual Conference on Learning Theory*.

[185] Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P. V., & Kawanabe, M. (2007). Direct importance estimation with model selection and its application to covariate shift adaptation. In *NIPS*.

[186] Szedmak, S. & Shawe-Taylor, J. (2007). Synthesis of maximum margin and multiview learning using unlabeled data. *Neurocomputing*, 70(7-9), 1254–1264.

[187] Talete srl (2007). *DRAGON (Software for Molecular Descriptor Caluclations)*. Talete srl, Milano, Italy. http://www.talete.mi.it/.

[188] Tseng, P. & Yun, S. (2009). A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117, 387–423.

[189] Vapnik, V. (1998a). *Statistical Learning Theory*. John Wiley.

[190] Vapnik, V. (1998b). *Statistical Learning Theory*. John Wiley and Sons.

[191] Wang, W. & Zhou, Z. H. (2007). Analyzing co-training style algorithms. In *Proceedings of the 18th European Conference on Machine Learning* (pp. 454–465).: Springer-Verlag New York Inc.

[192] Wang, W. & Zhou, Z. H. (2010a). Multi-view active learning in the non-realizable case. In *Neural Information Processing Systems*.

[193] Wang, W. & Zhou, Z. H. (2010b). A new analysis of co-training. In *Proceedings of the 27th international conference on Machine learning*.

[194] Wang, X., Pal, C., & McCallum, A. (2007). Generalized component analysis for text with heterogeneous attributes. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07 (pp. 794–803). New York, NY, USA: ACM.

[195] Wasseerman, L. (2004). *All of Statistics: A Concise Course in Statistical Inference*. Springer.

[196] Wenyuan Dai, Gui-Rong Xue, Q. Y. & Yu, Y. (2007). Co-clustering based classification for out-of-domain documents. In *Proceedings of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 210–219). San Jose, California, USA: ACM.

[197] Weston, J., Ratle, F., & Collobert, R. (2008). Deep learning via semi-supervised embedding. In *Proceedings of the 25th international conference on Machine learning* (pp. 1168–1175).: ACM.

[198] Wu, S., Zou, H., , & Yuan, M. (2008). Structured variable selection in support vector machines. *Electronic Journal of Statistics*, 2, 103–117.

[199] Wu, Y., Chang, E., Chang, K., & Smith, J. (2004). Optimal multimodal fusion for multimedia data analysis. In *Proceedings of the 12th annual ACM international conference on Multimedia* (pp. 572–579).: ACM.

[200] Wu, Y. & Liu, Y. (2007). Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association*, 102(479), 974–983.

[201] Xie, S., Fan, W., Peng, J., Verscheure, O., & Ren, J. (2009). Latent space domain transfer between high dimensional overlapping distributions. In *Proceedings of the 18th international conference on World wide web* (pp. 91–100).: ACM.

[202] Xing, E., Yan, R., & Hauptmann, A. (2005). Mining associated text and images with dual-

wing harmoniums. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI-2005)*.

[203] Xue, Y., Dunson, D., & Carin, L. (2007). The matrix stick-breaking process for flexible multi-task learning. In *ICML*.

[204] Yamazaki, K., Kawanabe, M., Watanabe, S., Sugiyama, M., & Müller, K. (2007). Asymptotic bayesian generalization error when training and test distributions are different. In *Proceedings of the 24th International Conference on Machine learning* (pp. 1079–1086).

[205] Yan, R. (2006). *Probabilistic Models for Combining Diverse Knowledge Sources in Multimedia Retrieval*. Technical report, In Ph.D Thesis.

[206] Yang, J., Liu, Y., Ping, E., & Hauptmann, A. (2007). Harmonium models for semantic video representation and classification. In *SIAM Conference on Data Mining* (pp. 1–12).: Citeseer.

[207] Ye, J., Chen, K., Wu, T., Li, J., Zhao, Z., Patel, R., Bae, M., Janardan, R., Liu, H., Alexander, G., et al. (2008). Heterogeneous data fusion for alzheimer's disease study. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1025–1033).: ACM.

[208] Yu, K., Zhang, T., & Gong, Y. (2009). Nonlinear learning using local coordinate coding. In *Advances in Neural Information Processing Systems 22*: Citeseer.

[209] Yu, S., Krishnapuram, B., Rosales, R., & Rao, R. B. (2011). Bayesian co-training. *Journal of Machine Learning Research*, 12, 2649–2680.

[210] Yu, S., Krishnapuram, B., Rosales, R., Steck, H., & Rao, R. B. (2008). Bayesian co-training. *Advances in neural information processing systems*, 20, 1665–1672.

[211] Yuan, M. & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68(1)(1-2), 49–67.

[212] Zhang, D., He, J., Liu, Y., Si, L., & Lawrence, R. (2011). Multi-view transfer learning with a large margin approach. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1208–1216).: ACM.

[213] Zhang, H., Chow, T., & Rahman, M. (2009). A new dual wing harmonium model for document retrieval. *Pattern Recognition*, 42(11), 2950–2960.

[214] Zhang, X. & Lee, W. S. (2007). Hyperparameter Learning for Graph Based Semi-supervised Learning Algorithms. In *Advances in Neural Information Processing Systems 19*.

[215] Zhao, P., Rocha, G., & Yu, B. (2006). *Grouped and hierarchical model selection through composite absolute penalties*. Technical report, Department of Statistics, University of California, Berkeley.

[216] Zheng, Z. & Padmanabhan, B. (2002). On active learning for data acquisition. In *IEEE International Conference on Data Mining* (pp. 562–569).: IEEE.

[217] Zhong, E., Fan, W., Peng, J., Zhang, K., Ren, J., Turaga, D., & Verscheure, O. (2009). Cross domain distribution adaptation via kernel mapping. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1027–1036).: ACM.

[218] Zhou, D. & Burges, C. J. C. (2007). Spectral clustering and transductive learning with multiple views. In *Proceedings of the 24th international conference on Machine learning* (pp. 1159–1166).: ACM.

[219] Zhou, N. & Zhu, J. (2007). *Group variable selection via a hierarchical lasso and its oracle property*. Technical report, Department of Statistics, University of Michigan.

[220] Zhou, Z., Zhan, D., & Yang, Q. (2007). Semi-supervised learning with very few labeled training examples. In *Twenty-Second AAAI Conference on Artificial Intelligence* (pp. 675–680).

[221] Zhou, Z. H. & Li, M. (2005). Tri-training: exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, (pp. 1529–1541).

[222] Zhou, Z.-H. & Li, M. (2007). Semi-supervised regression with co-training style algorithms. *IEEE Trans. on Knowl. and Data Eng.*, 19, 1479–1493.

[223] Zhu, J. & Hastie, T. (2004). Classification of gene microarrays by penalized logistic regression. *Biostatistics*, 5(3), 427–443.

[224] Zhu, X. (2008a). *Semi-Supervised Learning Literature Survey*. Technical report, Department of Computer Science, University of Wisconsin, Madison.

[225] Zhu, X. (2008b). *Semi-Supervised Learning Literature Survey*. Technical report, Department of Computer Science, University of Wisconsin, Madison.

[226] Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the twentieth international conference on Machine learning*.

[227] Zhu, X., Kandola, J., Lafferty, J., & Ghahramani, Z. (2006). Graph kernels by spectral transforms. In O. Chapelle, B. Scholkopf, & A. Zien (Eds.), *Semi-Supervised Learning*. Cambridge, MA: The MIT Press.

[228] Zhu, X., Khoshgoftaar, T. M., Davidson, I., & Zhang, S. (2007). Editorial: Special issue on mining low-quality data. *Knowledge and Information Systems*, 11, 131–6.