University of Tennessee, Knoxville

Trace: Tennessee Research and Creative Exchange

Doctoral Dissertations

Graduate School

12-2012

Extending Structural Learning Paradigms for High-Dimensional Machine Learning and Analysis

Christopher Todd Symons
University of Tennessee - Knoxville, csymons@utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Christopher Todd Symons entitled "Extending Structural Learning Paradigms for High-Dimensional Machine Learning and Analysis." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Computer Engineering.

Itamar Arel, Major Professor

We have read this dissertation and recommend its acceptance:

Hairong Qi, J. Wesley Hines, Stacy J. Prowell, Thomas E. Potok

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Extending Structural Learning Paradigms for High-Dimensional Machine Learning and Analysis

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Christopher Todd Symons

December 2012

Dedicated to Kai and Sena.

Acknowledgements

I would like to thank all of those who supported me throughout my time working on my dissertation. My friends and family have been a source of unlimited encouragement. I owe particular thanks to my advisor, Dr. Itamar Arel, without whose guidance and organizational skills I might never have completed my degree requirements. I also wish to thank my entire committee for their input and support. I am also grateful for the opportunity that my employer, the Oak Ridge National Laboratory, has given me through their Educational Assistance Program. This includes the tremendous support offered by my management at the lab, including Dr. Justin Beaver, Dr. Tom Potok, and Dr. Brian Worley. I would also like to express particular thanks to Dr. Stacy Prowell, who has offered me endless encouragement for over a decade. In addition, I would like to thank Dr. Raju Vatsavai and Dr. Goo Jun for their help and guidance with hyper-spectral satellite image data, Dr. Justin Beaver and Dr. Stacy Prowell for help and guidance in the field of network intrusion detection, and Dr. Karsten Steinhaeuser, Dr. Auroop Ganguly, and Dr. Chad Steed for guidance in the Climate domain.

The trick that one learns over time, a basic part of mathematical methodology, is to sidestep the equation and focus instead on the structure of the underlying physical process. -Richard Bellman

Abstract

Structure-based machine-learning techniques are frequently used in extensions of supervised learning, such as active, semi-supervised, multi-modal, and multi-task learning. A common step in many successful methods is a structure-discovery process that is made possible through the addition of new information, which can be user feedback, unlabeled data, data from similar tasks, alternate views of the problem, etc. Learning paradigms developed in the above-mentioned fields have led to some extremely flexible, scalable, and successful multivariate analysis approaches. This success and flexibility offer opportunities to expand the use of machine learning paradigms to more complex analyses. In particular, while information is often readily available concerning complex problems, the relationships among the information rarely follow the simple labeled-example-based setup that supervised learning is based upon. Even when it is possible to incorporate additional data in such forms, the result is often an explosion in the dimensionality of the input space, such that both sample complexity and computational complexity can limit real-world success. In this work, we review many of the latest structural learning approaches for dealing with sample complexity. We expand their use to generate new paradigms for combining some of these learning strategies to address more complex problem spaces. We overview extreme-scale data analysis problems where sample complexity is a much more limiting factor than computational complexity, and outline new structural-learning approaches for dealing jointly with both. We develop and demonstrate a method for dealing with sample complexity in complex systems that leads to a more

scalable algorithm than other approaches to large-scale multi-variate analysis. This new approach reflects the underlying problem structure more accurately by using interdependence to address sample complexity, rather than ignoring it for the sake of tractability.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Computational science is at the forefront of many efforts to extend scientific knowledge. As the problems being addressed become more complex, new computational paradigms must be designed to handle the complications that inevitably arise. Increased complexity can manifest in higher dimensionality, greater interdependence among variables, larger variety in data types, missing and incomplete data, more noise, etc. Typically, we will be dealing with all of these issues.

When casting the problem of predictive structure discovery as a computational analysis problem over experimental and/or simulation data, we are immediately confronted with well-known challenges related to data fusion and knowledge discovery in the analysis of high dimensional spaces with strong inter-dependencies among variables. The scientific community continues to struggle with this type of analysis in many important applications. In particular, while simulation approaches to scientific discovery are ideally suited to High Performance Computing (HPC), discovery of new model characteristics through analysis of the data remains a daunting challenge in the HPC community. Currently, analysis of extreme-scale data is typically performed on isolated portions of the data, ignoring interdependence for the sake of tractability. Although the curse of dimensionality is well known for its computational consequences, the more limiting factor in high-dimensional data is often due to its statistical consequences, which are also known as sample complexity. The sample complexity is related to the inability to account for the role of each of a large number of potentially influential variables given finite data.

New tools are required, but first and foremost, we also require new algorithmic approaches that are fundamentally different in design. They must deal with the sample complexity by taking advantage of information in the data that is being ignored and scale by design to perform analyses that are consistent with the nature of the problem, rather than being forced to make inaccurate independence assumptions for the sake of scalability. Maintaining a low dimensional representation by ignoring parts of the data (unless they are known a priori to be insignificant) does not solve the sample complexity side of the curse of dimensionality. In other words, the conclusions are just as unreliable for having ignored data, as they would be for not being able to account for it. We need methods that can systematically hypothesize interaction sub-structures among observed variables, observe correlations among the sub-structures, rank and identify contributing variables in the full-system context, and thereby capture the essence of the model underlying complex systems in the form of ultra-scale high-dimensional datasets.

In this work, we expand the basic foundations of the latest structural learning algorithms and explore their use in solving some of the most pressing problems in high-dimensional analysis in real world applications in which the relations between data elements fall outside of any single learning paradigm.

1.1 The curse of dimensionality: computational complexity vs. sample complexity

The *curse of dimensionality* (Donoho, 2000) is most widely known for its computational implications, perhaps because Richard Bellman, who is credited with coining the phrase, dealt with a problem in which the number of samples was not a limiting factor, since generating a sample would be the equivalent of evaluating a function in his domain (Bellman, 1961). However, as we attempt to use computational power to analyze more complex problems, the characteristic of the principle that often serves as the larger barrier to accurate analysis is *sample complexity*. Sample complexity is the side of the curse of dimensionality that essentially refers to the number of training examples required to allow effective statistical learning to take place. This is usually dependent upon the number of features, but the number of features should be thought of as the number of variables (complex or otherwise) that could play a role, as opposed to the number of variables recorded in the data. It can also be the case that the most useful features, perhaps the ones that represent the degrees of freedom in the problem space, are non-linear combinations of some of the recorded variables. In many large, complex applications, the sample complexity is the limiting factor, because although we may have extremely large sets of examples, the number is often small in relation to the actual dimensionality of the problem, even though we need it to be exponential in relation to the dimensionality. For example, even if we recorded everything of relevance, it could be the case that none of these recorded, *atomic*, variables is useful by themselves, but only when looked at as a conjunction with other variables. Even if the number of atomic variables, n, is reasonable, consider that the number of possible combinations to explore includes every non-empty member of the power set of these variables, which equates to $2^n - 1$. Thus, it is important to recognize how much we oversimplify analysis problems over complex systems when we cannot model them from first principles, because otherwise we are prone to overconfidence in our analyses.

Innate high dimensionality can result from natural complexity when there are an extreme number of factors that can be measured and that influence behavior in a system. Moreover, artificial high dimensionality can result from redundant and irrelevant factors being considered. As the problems computational science seeks to address grow more complex, we see both of these issues grow more prominent as natural complexity is exacerbated by lack of understanding of the system under study. Examples of importance include climate science, materials science, cyber security, etc.

In such domains, major challenges exist in data fusion and knowledge discovery in the analysis of high dimensional spaces composed of complex, inter-related structures and properties. Often there is information, whether recognized or not, that does not readily fit into current learning paradigms and statistical modeling techniques. We will focus on two goals in our work. One is the ability to learn to predict $y \in Y$ from examples (d-dimensional input vectors, x_i), while maximizing generalization performance on unseen data. In other words, machine-learning-based classification (e.g., $Y = \{0, 1\}$) and regression ($Y = \Re$) (Mitchell, 1997). The other is the ability to identify, and even construct, key influential inputs. While this can be thought of as feature selection, extraction, and/or construction (Blum and Langley, 1997; Kohavi and John, 1997; Guyon and Elisseeff, 2003; Liu and Yu, 2005), the application is often simply better understanding of influential interactions in a complex system. In the course of this work, we will focus on solving important real world problems that involve one or both of the following: 1) incorporating highly redundant information from another view or measurement modality of a problem, and 2) evaluating highly interdependent variables in a complex system. Both of these situations run straight into the sample complexity problems inherent in the curse of dimensionality.

1.2 Real world implementation issues

Unfortunately, data rarely comes in an easy to use form when addressing real applications. In this section we outline some of the most common problems and how we deal with them.

Much of the hard work in data analysis and machine learning comes during the data preparation phase. Observations can be numeric, categorical, or binary. Data can be recorded in different forms (text, images, etc.) using different numeric scales. Important observations may be missing, or even incorrect, for some examples. Some examples may belong to multiple classes simultaneously, and others may not belong to any known class. And that is just the tip of the iceberg!

Data normalization is common practice in data mining and machine learning, but the best way to do this is not always obvious. Many problem domains have different types of features and different scales at which they are measured. Since this can be problematic for some learning models, as standard practice, we convert categorical features to binary, and normalize all numeric data using a Softmax scaling approach, which is purported to retain the most information (Pyle, 1999).

Missing and incomplete data can be particularly problematic. The most common way of dealing with it is to impute missing values (Marwala, 2009). One of the advantages of the approaches to data integration discussed in this work is that they often do not require any special means for dealing with missing data. They allow the use of information that is normally not used, and this flexibility extends to the example level. For example, it is possible to alter a similarity score between two points based only on the data that is common between those points, thus negating the need for data imputation. Furthermore, in a task-based learning approach, when problems are constructed to consider common predictive structures, it is also possible to subdivide tasks to account for missing data.

Irrelevant and redundant data are particularly common in complex problem spaces. Therefore, the methods discussed in this work are designed to handle such information. It should be noted that simple methods for eliminating supposedly irrelevant features can be counter-productive, and thus we do not attempt to do so as a preprocessing step. For example, in some language processing tasks with sparse features, eliminating features that occur only once in the dataset can actually degrade performance (see Klein and Manning (2002) for an enlightening example of why this might occur).

1.3 Large data vs. complex data

It is important to note the difference between large data and complex data. There is a widespread and ever-growing effort to use ”big data” among businesses and government. Business intelligence tools have been commonplace for many years, and they are being used to a greater extent all the time. However, these tools are intended to work with clean, structured data, or else they help you evaluate the cleanliness of your data. They use standard statistics, which means they are not designed to deal with data representing ultrahigh-dimensional spaces and limited ground truth. Nor are they capable of incorporating data in ways that take advantage of the additional information we are focusing on in this work. They are primarily designed to analyze relatively simple relationships among and trends across cleanly measured variables collected in structured form. So, despite the fact that the work in the business world on ”big data” seems relevant to this proposal at first glance, it should be emphasized that such applications are almost completely unrelated to our work.

1.4 Main contributions

The major contributions of this work are outlined below.

1) Novel graph-building methods for graph-Laplacian-based semi-supervised learning techniques,
   a) Random subspace methods,
   b) Stochastic Discrimination graphs (Kleinberg, 1990; Irle and Kauschke, 2011),
   c) Theoretical justification,
   d) Demonstration of scalable graph-based learning methods,
2) Novel combined learning approach for budgeted, semi-supervised, multi-view (multi-modal) learning,
   a) Motivating applications for budgeted, multi-view learning
   b) Demonstrated effectiveness of knowledge carryover across views,
3) New, scalable framework for extreme-scale data analysis of complex systems data based on multi-task and semi-supervised learning,
   a) Automated construction methods for auxiliary tasks in complex systems,

b) Automated feature expansion methods for more extensive predictive structure exploration in complex systems,

c) Proof of principle for improved generalization ability,

d) Proof of principle of linear speedup,

4) Validation of approaches in multiple large-scale domains,

a) Cyber security: intrusion detection

b) Text processing

c) Climate

Chapter 2

Background

2.1 Moving beyond supervised learning

Supervised learning (Mitchell, 1997; Duda et al., 2001) is the most common form associated with the field of machine learning. It consists of having a human label information, and then having the computer learn to make the same judgments on new data. This simply requires one to define a set of classes (or a scoring mechanism in the case of regression) and a set of observables that constitute a sample (typically referred to as features). Then, in the most straightforward applications, based on examples classified by a human, the machine learns the significance of the features in determining the class to which an example belongs. There are too many forms of supervised learning to mention here, but two of the more popular forms that form the basis for much of the field are Support Vector Machines (SVMs) (Vapnik, 1998), a nonparametric form of learning that seeks to maximize the margin between classes, and Graphical Models (Bishop, 2006), a form of learning that seeks to constrain the search space for a solution by using links between random variables to assert dependence assumptions among them. While supervised learning represents the foundations of machine learning, the present and the future belong to adaptive learning methods, such as semi-supervised learning, active learning, multi-task learning, transfer learning, and reinforcement learning. These and similar subfields have arisen due to a need to adapt machine learning to real problem areas.

2.2 Semi-supervised learning

Semi-supervised learning (Chapelle et al., 2006) is generally defined as any learning method that uses both labeled and unlabeled data during the model discovery process. Methods for incorporating the unlabeled information can vary. Some methods include the use of data-dependent priors and low-density separation, but the most common approach is graph-based (Chapelle et al., 2006; Johnson and Zhang, 2008). Graph-based methods are typically designed based on the assumption that the data naturally occur on an underlying manifold, such that the true degrees of freedom of the problem can be discovered using unlabeled data. In other words, the most common approaches are non-linear forms of dimensionality reduction where unlabeled data is used to find a lower-dimensional space that is good for classification (or regression). In essence, the intent is to find structure in the ambient space that can be exploited to constrain the search for a good hypothesis. Some common forms of this type of nonlinear dimensionality reduction include Isomap (Tenenbaum et al., 2000), Locally-Linear Embedding (LLE) (Roweis and Saul, 2000), Laplacian Eigenmaps (LEM) (Belkin and Niyogi, 2003), Diffusion Maps (DM) (Nadler et al., 2005), Semidefinite Embedding (Weinberger and Saul, 2006), etc. A more comprehensive list can be found in Lin and Zha (2008).

A central construct in many of these methods is the graph Laplacian from spectral graph theory (Chung, 1997). A graph is constructed to represent a manifold (or densely populated region of interest in the ambient space), and the graph Laplacian facilitates the discovery of a low-dimensional space that is smooth with respect to this graph. In semi-supervised learning the goal is to augment learning through the use of unlabeled samples, so the unlabeled data is used to find a low-dimensional space on which learning via the labels can be more effective. The normalized graph Laplacian is a matrix defined as follows:

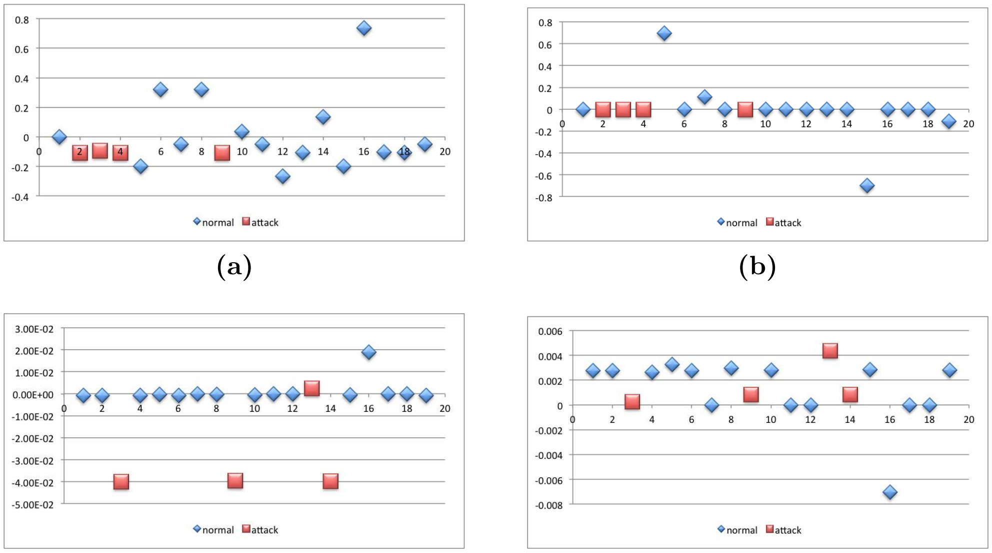
$$\mathcal{L}(u,v) = \begin{cases} 1, & \text{if } u = v \text{ and } d_v \neq 0 \\ \frac{-1}{\sqrt{d_u d_v}}, & \text{if } u \text{ and } v \text{ are adjacent} \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

where u and v are vertices in the graph, d is the degree (number of incident edges) of a vertex, and adjacency refers to a neighboring connection in the graph. The unnormalized form given below is also commonly used:

$$L(u,v) = \begin{cases} d_v, & \text{if } u = v \\ -1, & \text{if } u \text{ and } v \text{ are adjacent} \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

Using the eigenvalues and associated eigenvectors of these positive, semi-definite, symmetric matrices provides a method for discovering dimensions that are smooth with respect to the graph that defines it. If the graph varies smoothly with respect to the target problem (i.e. examples from different classes or clusters are rarely linked, similar examples from the perspective of the target problem are linked, etc.), then it can be used to represent a manifold. The Laplacian of the graph can then be used to find a space that roughly represents that manifold. The eigenvector associated with the smallest non-zero eigenvalue is smoothest with respect to the graph, such that points connected in the graph will be close together in the dimension defined by said eigenvector. This smoothness with respect to eigenvector-defined dimensions decreases as you progress to the larger eigenvalues.

Another useful property of the eigensystem is the fact that the number of zero-value eigenvalues is equal to the number of connected components in the graph. In addition, an eigenvector will not involve more than one component of the graph. Thus, after counting the number of connected components, which is an $O(n)$ operation, you



(a) **(b)**

(c) **(d)**

Figure 2.1: Eigenvector values of the labeled data from the graph Laplacian based on nearest neighbors. 2.1a and 2.1b are the values in the first and second dimensions, respectively, using a graph based on labeled points only. 2.1c and 2.1d are the values in the first and second dimensions, respectively, when using a graph of the labeled and unlabeled data. In this case, the addition of the unlabeled data provides a transformation that allows linear separation of the two classes.

need to retain at least that many dimensions in a new space in order to distinguish between all points after they are mapped.

Typically, semi-supervised learning methods based on the graph Laplacian separate the manifold discovery process from the learning process in such a way that the manifold discovery is completely unsupervised. There is a general recognition that the labels can be used as constraints when building the graph, but in semi-supervised learning, such an approach would touch only a small portion of the graph. In (Goldberg et al., 2007), a dissimilarity measure is used to alter the graph. However, the method either requires ground truth dissimilarity, in which the only parts of the graph that are affected are those for which labels are available, or dissimilarity based on some domain specific features that are manually constructed to enforce disparity.

As an example of a real nonlinear transformation using this method, Figure 2.1 shows the values of the first two non-zero eigenvectors for the labeled points during training of a classifier using the Kyoto2006+ intrusion detection dataset (Song et al., 2011). This is a two-class classification problem, where the classes represent attack behaviors and normal behaviors in a network. The semi-supervised classifier was built using 19 labeled and 3022 unlabeled examples sampled from a single day's worth of data. On the test set from Kishimoto et al. (2011), this classifier achieves an AUC score of 98.5, with a recall of 71.4% and a false positive rate of 0.02%. For comparison, Kishimoto et al. (2011) use a multi-classifier approach that uses over 10 million training examples and achieves a recall of 80.9% with a false positive rate of 5.9%.

There are many other approaches to semi-supervised learning. One that is particularly noteworthy for its ability to use the labels in a more robust manner is the predictive structure framework of Ando and Zhang (2005). Their approach is based on multi-task learning and attempts to find structure that overlaps many prediction problems that are formulated in such a way that they always have labels. In other words, many of the tasks they use are simply predicting things that are observable at all times, such as input variables. An interesting thing to note about the experiments in Ando and Zhang (2005) is that many of the formulated tasks are constructed with the use of the labels, and it is these tasks that seem to help the most with the more complex (noisy) problems, particularly if the number of available labels was non-trivial. One drawback to the approach is that the framework, as described, requires a different kind of domain expertise in terms of the construction of the formulated tasks.

2.3 Active learning

Active learning, which can also be referred to as selective sampling (Cohn et al., 1994; Freund et al., 1997; Dasgupta, 2005; Dasgupta et al., 2005; Balcan et al., 2008; Druck et al., 2009; Settles, 2009), is another method that can be used to affect the sample complexity of a problem. The objective of active learning is most commonly described as either achievement of similar performance using a substantially reduced amount of training data or improvement of generalization performance on a fixed-size training set. Its viability has been demonstrated in the literature via noticeable improvements in generalization performance (Cohn et al., 1994; Schohn and Cohn, 2000; Abe et al., 2006) and/or reductions in annotation requirements (Sassano, 2002; Brinker, 2003; Shen et al., 2004). More recently, theoretical evidence has been provided for its effectiveness (Freund et al., 1997; Dasgupta, 2005; Balcan et al., 2008).

In the active-learning literature, many measures have been utilized to actively select the most valuable samples for use in training, including uncertainty (Schohn and Cohn, 2000; Tong and Koller, 2002), variance in prediction from query-by-committee (Freund et al., 1997), and risk minimization (Zhu, 2003). Uncertainty-based measures typically utilize some means of evaluating model confidence in order to assign significance to a sample. Thus, this type of measure is usually model-dependent. However, using uncertainty alone is not advisable because it often ignores the distribution of the available data. In other words, there is the danger of selecting outliers and degrading generalization performance due to overfitting of data that does not represent the real distribution. To emphasize this point, it should be noted that most machine learning methods are built on the assumption that the data is *independently and identically distributed* (*i.i.d.*), such that the data on which they will be applied comes from the same distribution as the data on which they were trained. However, this assumption can be drastically violated by some selective sampling approaches.

For this reason, several methodologies have emerged that take into account the distribution of the unlabeled data. Freund et al. (1997) apply a query-by- committee algorithm to selective sampling on certain concepts that are dense-in-themselves, and prove an exponential improvement in the sample complexity. McCallum and Nigam (1998) apply active learning with Expectation Maximization on a text classification task. That work employs information concerning the representativeness of a document, which allows consideration of the data distribution.

In many practical cases, active learning is performed in batch. In other words, instead of a single example being chosen at each round of active learning, examples are selected as a batch in order to limit retraining costs. Thus, some methods that attempt to choose a set of diverse examples have been proposed. Brinker (2003) and Shen et al. (2004) seek to include a diverse sample by using geometric distance computable in the Euclidean space of Support Vector Machines (SVMs). Sassano (2002) reports that diverse batch selection is possible with a two-pool method when it is applied to Japanese word segmentation.

It should be emphasized that efforts to incorporate the data distribution in sample selection procedures have repeatedly been shown to be important for optimally increasing generalization performance, and it has been empirically demonstrated that such efforts result in improved outcomes (Cohn et al., 1994; Schohn and Cohn, 2000; Freund et al., 1997; Zhu, 2003). In addition, some specific formal problems have been shown to have strong theoretical guarantees in this regard (Freund et al., 1997; Dasgupta et al., 2005).

More recent work has extended active learning beyond the notion of selective sampling of examples. In particular, Druck et al. (2009) use generalized expectation criteria (Mann and McCallum, 2008) to allow labeling of features instead of examples.

2.4 Multi-view and multi-modal learning

The idea of multi-view learning gained a lot of traction in the machine-learning community with the development of co-training (Blum and Mitchell, 1998), which was one of the first well-known, semi-supervised learning approaches. The problem on which it was tested involved two views of a web-page classification problem. The first view was constructed from the words in the content of the page, while the second was constructed of the words underlined in the hyperlinks to the page. The approach was based on label-propagation. Labeled samples in one view were used to classify examples that would comprise training data for the second view, and the approach would iterate until all samples were labeled. The idea was that the two classifiers should agree, and if they did, then one could be confident in the resulting models. However, this approach has also been used to demonstrate the risk of error propagation when propagating labels.

Multi-view learning has since received more attention as a problem in its own right (Foster et al., 2009). In particular, its relevance to addressing problems with multiple measurement modalities is becoming clear (Singh et al., 2009; Symons and Arel, 2011). Methods of graph-based semi-supervised learning have clear applicability here as well (Sindhwani et al., 2005a; Sindhwani and Rosenberg, 2008).

2.5 Multi-task and task-transfer learning

Multi-task learning (Caruana, 1997; Ben-David and Schuller, 2003; Bakker and Heskes, 2003; Xue et al., 2007) is a machine-learning paradigm that uses the similarities among two or more learning tasks to improve the generalization performance of each of the individual models. Multi-task learning is often used interchangeably with the idea of task-transfer learning (Taylor et al., 2008). However, task-transfer learning is more commonly thought of as being incremental in nature, such that one is applying previously learned knowledge the way a human would, whereas multi-task learning usually refers to a more specific subfield of machine learning where tasks are learned jointly, typically through some type of joint optimization. Recent work has provided theoretical guarantees (Ando and Zhang, 2005; Ben-David and Borbely, 2008) for certain notions of task-relatedness.

Various methods exist, but the general approach typically centers around finding structures that are useful (predictive) in multiple tasks, such as in Ando and Zhang (2005). Such approaches can provide more samples upon which to judge feature relevance. In Xue et al. (2007) a Dirichlet process prior is used to judge which tasks are similar enough that they should be learned together. They outline a *symmetric* joint learning approach, which is more along the lines of traditional multi-task learning, and an *asymmetric* learning approach where the prior is learned from previous tasks, such that the method falls more squarely under task-transfer learning. Bakker and Heskes (2003) use neural networks in which some model parameters are shared among all tasks while other parameters are unique to a task. Task clustering is performed to find the shared parameters.

2.6 Common threads

This chapter highlights a great deal of overlap across the learning paradigms mentioned above. The use of unlabeled data is ubiquitous, and a common thread is the use of additional information to find structure in the feature space. Typically, the predictive nature of the structure is considered, but not always. The goal is to find a new hypothesis space in which to search for an good model, but ideally, the new space should be smaller, while containing better choice of models. If one can indeed use additional information to find such a hypothesis space, either through feature expansion, feature selection, non-linear dimensionality reduction, etc., then a better model should result.

Chapter 3

Robust graph-based semi-supervised learning

The frontiers of machine learning research continue to expand based on requirements identified through new applications. As the potential of computational learning is recognized by a more diverse set of users with broader and more complex sets of issues to analyze, machine learning practitioners are in a position of constantly adapting algorithms to the peculiarities of new domains. The more labor involved in applying algorithms to new problems, the less likely that the advantages they offer will ever be realized. Thus, the historical trend has been one in which classification algorithms that are less sophisticated, but better understood, are applied despite their limitations. Unless new methods are made extremely robust to variations in application, they often remain a niche product used by only a few, regardless of any potential they might have to enhance our data analysis capabilities across a broad set of applications. In this chapter, we outline new methods for enhancing the applicability of graph-based semi-supervised learning to a more complex set of domains. This work covers and expands upon work published in Symons et al. (2012).

3.1 Understanding assumptions for more effective use of information

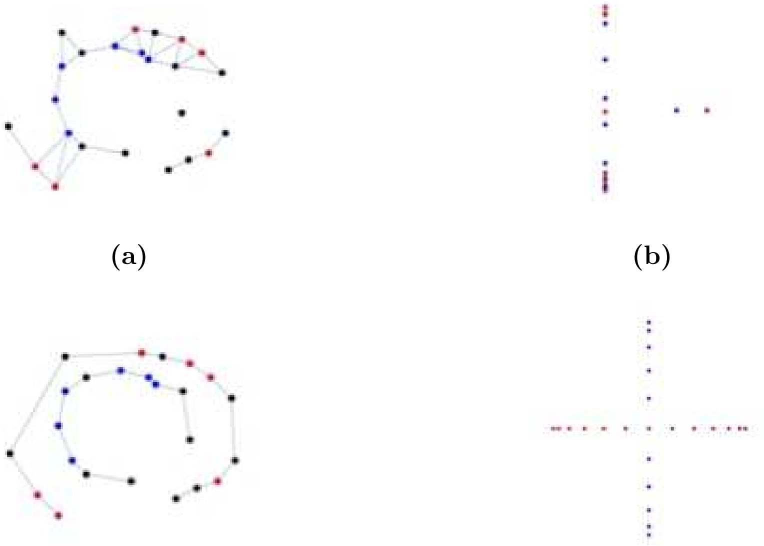
Many popular forms of semi-supervised learning rely on graph-based methods for regularization or spectral dimensionality reduction. Some common forms of nonlinear dimensionality reduction include Isomap (Tenenbaum et al., 2000), Locally-Linear Embedding (LLE) (Roweis and Saul, 2000), Laplacian Eigenmaps (LEM) (Belkin and Niyogi, 2003), Diffusion Maps (DM) (Nadler et al., 2005), Semidefinite Embedding (Weinberger and Saul, 2006), etc. A more comprehensive list can be found in Lin and Zha (2008). A central construct in many of these methods is the graph Laplacian. A graph is constructed to represent a manifold (or densely populated region of interest in the ambient space), and the graph Laplacian facilitates the discovery of a low-dimensional space that is smooth with respect to this graph. In semi-supervised learning the goal is to augment learning through the use of unlabeled samples, so the unlabeled data is used to find a low-dimensional space on which learning via the labels can be more effective.

As emphasized in Goldberg et al. (2008), normalized-output algorithms, such as LLE, LEM, and DM, do not handle noise well, and should not be applied arbitrarily, and there is a need for improvements that are robust. Goldberg et al. (2009) recognize the potential problem of having data that resides on multiple manifolds and offer some methods for applying semi-supervised manifold learning in such cases. However, while the methodology adds some robustness in such multi-manifold cases, the manifold description is made without the use of the labels so that high levels of noise can still hide the manifolds. Furthermore, it is quite possible that many of these discoverable manifolds are not relevant to the target problem.

A consistent theme among most graph-based semi-supervised learning approaches is that the unlabeled data is used to discover the manifold (or the decision space in general, as in the case of a cluster-like assumption), and the labeled data is used to learn a model in this new space. We present a simple method for adding robustness to these methods by taking a different view of the semi-supervised aspect of these algorithms. We turn the focus to the fact that these methods can be used to find functions that are smooth *with respect to the graph that we construct.* From this perspective, it makes sense to use the labeled data to guide the graph construction process in order to avoid close connections (which will be preserved in the reduced space) that are created due to noise in the feature set. To see why this is problematic, see the synthetic examples in Figure 3.1. The figures show the results of Laplacian-Eigenmap dimensionality reduction on each of the two graphs. It is not hard to imagine that when the ambient space is noisy, nearest neighbors built without regard to label information will often not represent a manifold of interest, as in Figure 3.1a, and thus the targeted dimensionality reduction would not be useful, as in Figure 3.1b.

However, in order to make good use of the labeled data, it is necessary to go beyond using them as constraints. It is also necessary to be cognizant of the fact that there likely will be within-class clusters, so that we must avoid strongly linking members of the same class if they are from very different subsets of that class. Doing so would violate the assumptions of local-distance preservation (whether geodesic or other) on which the methods are based. In practice, using the labels to guide the graph construction can confound the search for a relevant manifold if the labels are applied carelessly, while conservative methods of using the labels have little effect. The approach presented here takes care to preserve relevant local structure by using random subspaces as an edge weighting mechanism. Our goal is to find a method that allows us to incorporate expert domain knowledge via the labels in a way that can find relevant but subtle differences between samples. Therefore, we also discuss more direct methods for biasing the graph and demonstrate the difficulty of finding a robust approach.

The focus of this chapter is on the graph-construction process in this area of semi-supervised learning and how it can be improved to find a better decision space. The principle intuition is the same for any method that uses a graph constructed from data points to represent a manifold that one hopes to discover.



(a)

(b)

(c)

(d)

Figure 3.1: Each of the graphs 3.1a and 3.1c is intended to represent a manifold or manifolds in a given two-class problem (red and blue). We assume that black points are unlabeled, and that the manifolds in 3.1c are ones that are relevant to the target problem. In 3.1b and 3.1d, we plot the points according to the coordinates generated by the first two non-zero eigenvectors resulting in the eigensystem of the Laplacian matrix of the graphs in 3.1a and 3.1c, respectively. To demonstrate the effect in terms of the target problem, in plots 3.1b and 3.1d, we color each point according to it's actual manifold in 3.1c. In other words, all of the points on the outer semi-circle are colored red, and all of the points on the inner semi-circle are colored blue. It is obvious, that if 3.1c does indeed represent the true manifolds, then using proximity alone, without regard to label information, can be problematic.

3.2 Graph modifications

Typically, semi-supervised learning methods based on the graph Laplacian separate the manifold discovery process from the learning process in such a way that the manifold discovery is completely unsupervised. There is a general recognition that the labels can be used as constraints when building the graph, but in semi-supervised learning, such an approach would touch only a small portion of the graph. In Goldberg et al. (2007), a dissimilarity measure is used to alter the graph. However, the method either requires ground truth dissimilarity, in which the only parts of the graph that are affected are those for which labels are available, or dissimilarity based on some domain specific features that are manually constructed to enforce disparity.

The approach to graph building in this paper seeks to utilize the small number of labels assumed to be available to influence the edge construction in a manner that allows concept-specific structure to be encoded in the graph. We imagine that this approach can be less than optimal for extremely clean feature spaces that already meet the assumption that nearby points are in the same class. However, this assumption is rarely valid in practice, unless the feature space has already been heavily engineered. We are primarily concerned with finding good semi-supervised solutions for newer domains that lack a good understanding of the noise and how to eliminate it in preprocessing. Moreover, we focus on problems where labeled data is expensive and unlabeled data is abundant.

As in the Augmented PAC Model described in Balcan and Blum (2006) we approach the problem from the viewpoint that there should be a notion of compatibility, χ, between the hypothesis and the data distribution (as estimated using the unlabeled examples). Graph-based manifold learning methods assume that the target concept passes through a dense portion of the ambient space, and that this manifold can be discovered using the unlabeled data. In data sets where the feature space has been refined (e.g. image recognition data that has been preprocessed, centered, scaled, etc.), the ability to find the manifold is well-demonstrated experimentally.

Unfortunately, most applications do not have such a nice, clean ambient space, and yet, the decision space will almost always have an innate dimensionality that is much lower than that of the ambient space. If we then make the further (rather uncontroversial) assumption that there will be features whose similarity across examples has nothing to do with the target problem (either because they are noisy or they represent an observation that is not noise per se, but is not relevant to the target concept), then these features should ideally not be used to construct the classifier's representation of the manifold. So, with respect to the Augmented PAC Model, our compatibility matches their notion of compatibility with a data distribution over edges, such that two adjacent vertices should have a common label. However, intuitively, improving the chance that two edges will in fact share a label, should also provide greater justification for constraining the search to match this graph-imposed compatibility constraint.

In semi-supervised learning, the amount of labeled data is typically small. This raises several issues when trying to incorporate useful information from the labels at an early stage; i.e. when the dimensionality is still large. For example, feature selection is made that much more difficult, and therefore, attempts at using feature selection to eliminate noise are not particularly helpful. Furthermore, applying the labels directly through the use of a classifier is not advisable, since many classes contain subclasses or clusters, such that enforcing links between points based on labels alone is unwise. For example, creating an edge between two very dissimilar points can create a graph that violates the Riemannian assumptions that underpin the intuition of the Laplacian Eigenmap approach. We would like to have some sense that the end result will preserve some semblance of geodesic distance (even if the true distances are obscured by noise in the ambient space). Even if using a classifier for label propagation was a good idea, the high dimensionality would make it difficult to learn an accurate classifier at this stage.

3.2.1 Random Subspace Graphs

We first utilize a method based on random subspace selection (Bryll et al., 2003; Skurichina and Duin, 2001) in order to allow the labels to influence the graph construction in a robust way. The approach is simple in that we randomly select many feature splits and use them to train simple classifiers, each of which represents one subspace. Random subspace selection is sometimes used as a method for finding different views for ensemble learning. In such cases the hope is that the views will be independent. We do not necessarily have the same inclination here, which is important, because it could be difficult to ensure independence due to label sparsity. In ensemble learning, the independence is a necessary condition to ensure improved accuracy. However, we care very little about accuracy in that sense, because we are not looking to make a real classification. In fact, we simply want to place examples together in such a way as to provide some sense of smoothness according to our target, and therefore, if all of the classifiers were completely accurate, we would expect to end up ignoring any in-class clusters, and thus we would still have difficulty finding a good manifold.

What we do want is an ability to link together examples that share subsets of features that are useful in determining a classification in our target problem. Using the classifications of many classifiers that represent *good* hypotheses from different subspaces allows us to find a more delicate similarity that represents our target concept. If two examples share feature subsets that are useful according to our labeled data, then we want them to be classified together. If they both lack good predictive features in particular subspaces, then they will often be misclassified together, which is what we want. Another advantage to using random subspaces is that we can expect the classifiers to more effectively utilize the labeled data based on the fact that random subspace selection reduces the dimensionality of the feature space on which each classifier learns.

In the method that we employ in our experiments, we first perform one hundred random feature splits, resulting in two hundred classifiers that are trained on the labeled data based on the subset of features that represent their hypothesis space. We then construct a nearest neighbor graph based on cosine similarity while weighting the similarity scores based on the percentage of shared classifications using the random subspaces. We then build the unnormalized graph Laplacian. Using an unweighted graph not only performs well in practice, but we also expect it to be more robust to noise.

There are several parameters here that can be chosen rather arbitrarily. Finding optimal ways to choose these parameters (i.e. model selection) forms a good basis for improving the method, but that lies outside of the scope of this paper. Therefore, for our experiments we chose these parameters based on suggestions from the literature whenever possible. We retained the six nearest neighbors for most of our experiments, but we used eight nearest neighbors for the text categorization to mirror the same choice as that made in Belkin and Niyogi (2004). Similarly, unlike some methods, the use of Laplacian Eigenmaps does not automatically suggest the size of the new space. Therefore, we typically retained a basis size that was twenty percent of the number of labels used, once again to reflect the suggestions from Belkin and Niyogi (2004), although we explore this parameter in one set of experiments.

The graph construction algorithm is shown in Algorithm 1, where $w_{u,v}$ is the weight on the edge between vertices u and v, n is the size of the dataset including both labeled and unlabeled samples, and m is the dimensionality of the feature set.

3.2.2 Stochastic Discrimination graphs

Stochastic Discrimination

Stochastic Discrimination (SD) (Kleinberg, 1990, 2000b; Irle and Kauschke, 2011) is an ensemble method of classifier construction, in which so-called *weak* classifiers are combined to make a higher-level model. SD differs in significant ways from standard

Algorithm 1 Random Subspace Graph Construction

```
Input: data $\{(x_i, y_i)\}_{i=1}^{l}$, $\{x_i\}_{i=l+1}^{n}\}$, $numNeighbors := k$, $numFeatureSplits := s$
for $i = 1$ to $s$ do
    Randomly split feature set into two equal parts
    Train linear classifier on each part
    Classify each sample point using the classifiers
end for
for $u = 1$ to $n$ do
    for $v = 1$ to $n$ do
        $w_{u,v} = \frac{\sum_{i=1}^{m} u_i v_i}{\sqrt{\sum_{i=1}^{m} u_i u_i}\sqrt{\sum_{i=1}^{m} v_i v_i}}$ × (percent of time classified together)
    end for
end for
Retain $k$ nearest neighbors; create Laplacian matrix $L(u, v)$
```

methods of combining classifiers. Most notably, it is well known for its ability to support complex models without overfitting.

Stochastic Discrimination can be described most naturally as a binary classification scheme. Thus, while it is possible to build multi-class SD models, we will restrict our explanation here to the binary case. As described in Kleinberg (2000b), while most ensemble methods generate weak classifiers that are in some sense experts on the problem in that they will all agree on at least some of the easy points, it is important that this not happen in SD. While Stochastic Discrimination is a random subspace method in the most general use of the term, the weak models produced by the method are very different from those generated by other random subspace methods.

Selection of the weak learners is guided very strictly by three overarching principles: *generalization*, *uniformity*, and *enrichment*. In order for a model to have the ability to generalize, weak models must cover enough space to capture points outside of the training data. In other words, the weak models must apply to test points, such that standard generalization assumptions apply. A weak model must also have at least some discriminatory power, even if its error rate is close to fifty percent. Thus, an enriched model is one that contains a greater fraction of the labeled points from one class than from the other class. This does not simply mean that it has more of one class than the other; rather it means that the percentage of all points of class 1 covered by the weak model is greater than the percentage of all class 2 points that it covers. The amount of enrichment that one requires a new weak learner to have can be set using a parameter, β, that defines the minimum difference between the coverage percentages of the two classes. Finally, the algorithm seeks to ensure uniformity of coverage of points. This uniformity is class-specific, such that we won't add a new model to the ensemble even if it is enriched, unless the average coverage of points of each class is less than the average coverage for that class so far (plus some constant λ).

The following definitions are taken from Kleinberg (2000b):

Definition 1. Enrichment: A subset of the feature space (i.e., a weak classfier) $\mathcal{M}$ of F is said to be enriched with respect to classes C_1 and C_2 if

$inf\{|Pr(M|C_1) - Pr(M|C_2)|\ |M \in \mathcal{M}\} > 0$

Definition 2. Uniformity: A subset of the feature space (i.e., a weak classfier) $\mathcal{M}$ of F is said to be uniform with respect to classes C_1 and C_2 if for every point, p, in either C_1 or C_2, and every nonempty subset of $\mathcal{M}$ of the form $\mathcal{M}_{x,y}$, $Pr_F(p \in M | M \in \mathcal{M}_{x,y})$ is equal to x if p is a member of C_1, and is equal to y if p is a member of C_2.

The idea is to generate an ensemble model consisting of a wide variety that subsets of the feature space, such that those subsets (classifiers) cover points evenly, cover enough space to provide generalization ability, and cover a disproportionate number of points from one class, C_1, or the other, C_2.

To turn these ensembles into classifiers, a new test point is evaluated based on the subsets that cover it, as well as those that do not. Each subset contributes the following score to the point:

$$X_{(C_1,C_2)}(p, S) = \left(\frac{1_S(p) - Pr(S|C_2)}{Pr(S|C_1) - Pr(S|C_2)} \right), \tag{3.1}$$

where 1_S is the indicator function of the set S. In other words $1_S(p) \mapsto 1$ for points $p \in S$, and $1_S \mapsto 0$ for points $p \notin S$.

Then, points are classified using the following sum:

$$Y^t_{(C_1,C_2)} = \frac{\left(\sum_{k=1}^{t} X^k_{(C_1,C_2)}\right)}{t}, \tag{3.2}$$

where t is the number of classifiers.

It is possible in this fashion to cause points of class C_1 to have a mean of 1.0 and points of class C_2 to have a mean of 0. New points can then be classified as belonging to class C_1 when $Y^t_{(C_1,C_2)} > 0.5$. By the Central Limit Theorem, as t approaches infinity, the variance of the probability density function of points of class C_1 and the variance of the probability density function of points of class C_2 both approach zero. This fact helps explain the resistance to overfitting as the number of weak classifiers is increased.

The SD algorithm as described in Kleinberg (2000b,a) can be implemented in a variety of ways. In particular, the performance is heavily dependent upon how the stream of weak classifiers is generated and which classifiers are retained (e.g., the choice of beta; How enriched do retained classifiers have to be?). One significant choice we make in our implementation is that we grow weak classifiers around neighboring points rather than choosing the expansion points at random. The motivation for this is the fact that we want to use SD to generate a psuedometric that is both target-based and local-proximity-based.

Graph construction using SD

From a more theoretical standpoint, the characteristics we want to require in our graph construction approach are resistance to overfitting, since we won't have much labeled data, and lack of correlation between ensemble members to ensure a globally applicable psuedometric. Among ensemble classification methods, there are two prominent approaches that have both of these properties. The first, is the well-known AdaBoost algorithm (Schapire et al., 1997; Schapire and Freund, 2012). The second is Stochastic Discrimination (Kleinberg, 2000b,a; Irle and Kauschke, 2011). AdaBoost ensures that ensemble-members' errors are not correlated by adding more weight in the next round to examples on which the current ensemble makes mistakes or is unsure. While AdaBoost modifies the training set from the example point of view, SD modifies it from the feature point of view. Thus, SD is a natural fit for what we want to do, while AdaBoost is not.

As in the more general random subspace method used in Algorithm 1, we can use Stochastic Discrimination to generate a task-relevant psuedometric. Kleinberg discusses the point that weak classifiers generated via SD are not classifiers in the traditional sense of the word. Similarly, we dont want classifiers in the traditional sense of the word either. Just as SD depends on the weak learners being error prone, we do too. In other words, if each SD learner was highly accurate, we would likely be joining together many points that, while sharing the same class, are not close in the sense of where they lie on the manifold that we want to discover.

Owing to the over-fitting resistance of SD, we are able to benefit from additional weak classifiers as they help lower the overall bound on the generalization error without any real risk of hurting performance. In addition, because the weak classifiers are error prone, we can obtain a fine-grained pseudometric simply by using the scores (Equation 3.1) generated by the weak classifiers covering any given two points. This pseudometric captures local proximity due to the way we create our stream of classifiers, and any bound on the generalization error that applies to the SD algorithm applies to our graph edges as well, with a sufficient number of weak classifiers and a sufficient number of unlabeled points.

The graph construction algorithm is shown in Algorithm 2.

3.3 Theoretical analysis

The framework described in this section can be considered to rely on a notion of compatibility, χ, as described in Balcan and Blum (2006); Balcan (2008). The notion

Algorithm 2 SD Graph Construction

```
Input: data $\{(x_i, y_i)\}_{i=1}^{l}$, $\{x_i\}_{i=l+1}^{n}\}$, $numNeighbors := k$, $numWeakClassifiers := c$, $\beta$, $\lambda$, $minPoints := q$
Train SD classifier: $SD(c, \beta, \lambda, q)$
for $u = 1$ to $n$ do
  for $v = 1$ to $n$ do
    $w_{u,v} = \sum_{i=1}^{c} \begin{cases} \frac{1_{S_i}(u) - Pr(S_i|C_2)}{Pr(S_i|C_1) - Pr(S_i|C_2)}, & \text{if } 1_{S_i}(u) = 1_{S_i}(v) \\ 0, & \text{otherwise} \end{cases}$
  end for
end for
Retain $k$ nearest neighbors; create Laplacian matrix $L(u, v)$
```

of compatibility is based on finding a model that has a low *unlabeled error rate*. In the case of a graph regularizer, this can indicate that the function being learned *agrees with the graph* and would not label two connected nodes with different class labels. Of course, if the graph incorrectly connects examples from different classes, then the target function itself does not have an unlabeled error rate of zero, even if some hypotheses do.

In Balcan (2008), various sample complexity bounds are provided. In some cases an assumption is made that the target function's unlabeled error rate is low (essentially zero), and in other cases the bounds depend on the unlabeled error of c^*, the true target function. For example, Theorem 2.3.2 provides a sample complexity bound in the realizable case ($c^* \in C$) that depends upon the unlabeled error of the target, c^*. A graph constructed over noisy samples is likely to have many ”errors.” Therefore, the first assumption is too simplistic for many real-world situations. Using unlabeled data alone, the target function's unlabeled error cannot be bounded at all, since it is entirely possible that similarity in the ambient feature space does not reflect similarity in terms of the target concept at all. In other words, the number of mistakes in the notion of compatibility itself (the graph) cannot be bound while ignoring all information concerning the target concept. Although labeled and unlabeled error are of different types, it should still be possible to use supervised-learning bounds on generalization error to provide a bound on the unlabeled error rate of c^*, meaning that use of label information in the construction of the graph can bound this error with respect to the target, allowing bounds to be applied to the overall sample complexity.

First, let's look at the definition of *notion of compatibility* given in Definition 2.2.1 from Balcan (2008), which we restate here, in slightly modified form, for completeness.

Definition 3. A notion of compatibility is a function $\chi : C \times X \mapsto [0, 1]$ where we define $\chi(f, D) = \mathbf{E}_{x\sim D}[\chi(f, x)]$. Given a sample S, we define $\chi(f, S)$ to be the empirical average of χ over the sample.

C is the *hypothesis space* from which the hypothesis, f, is chosen, and X is the *instance space*, from which the distribution, D, is drawn. For our purposes, we will actually assume that $\chi(f, D) = \mathbf{E}_{x\sim D}[\chi(f, x_i, x_j)]$, so we are looking at the expectation over pairs of examples (edges in the graph). Note that the *unlabeled error rate* is simply a measure of incompatibility between a hypothesis, f, and the distribution, D; i.e. $1 - \chi(f, D)$, or $1 - \chi(f, S)$, for a given sample.

Now, consider Theorem 2.3.2 from Balcan (2008), which we restate verbatim here for completeness.

Theorem 4. *If $c^* \in C$ and $err_{unl}(c*) = t$, then m_u unlabeled examples and m_l labeled examples are sufficient to learn to error ϵ with probability $1 - \delta$, for*

$$m_u = \frac{2}{\epsilon^2}\left[ln|C| + ln\frac{4}{\delta}\right] \text{ and } m_l = \frac{1}{\epsilon}\left[ln|C_{D,\mathcal{X}}(t + 2\epsilon)| + ln\frac{2}{\delta}\right].$$

In particular, with probability at least $1-\delta$, the $f \in C$ that optimizes $\hat{err}_{unl}(f)$ subject to $\hat{err}(f) = 0$ has $err(f) \leq \epsilon$.

Alternatively, given the above number of unlabeled examples m_u, for any number of labeled examples m_l, with probability at least $1-\delta$, the $f \in C$ that optimizes $\hat{err}_{unl}(f)$ subject to $\hat{err}(f) = 0$ has

$$err(f) \leq \frac{1}{m_l}\left[ln|C_{D,\mathcal{X}}(err_{unl}(C^*) + 2\epsilon)| + ln\frac{2}{\delta}\right].$$

Next, we need the following definition from Kleinberg (2000a):

Definition 5. An m-class problem in supervised learning, presented as two finite sequences $\mathbf{E} = (E_1, E_2, \ldots, E_m)$ and $\mathbf{T} = (T_1, T_2, \ldots, T_m)$ of classes in a finite feature space (intuitively, all examples and the training examples, respectively), is said to be solvable if there exists a collection $\mathbf{M}$ of subsets of the feature space such that $\mathbf{T}$ is $\mathbf{M}$-representative of $\mathbf{E}$, and such that $\mathbf{M}$ is $\mathbf{T}$-enriched and $\mathbf{T}$-uniform.

Note that *enrichment* and *uniformity* are defined in Section 3.2.2

Now, consider Theorem 1 from Kleinberg (2000a), which we also restate essentially verbatim here for completeness.

Theorem 6. *There exists an algorithm $\mathcal{A}$ with the following property: given any solvable problem, $\mathbf{E}$, $\mathbf{T}$, in supervised learning, if $\mathbf{M}$ is a collection of subsets of the feature space, such that $\mathbf{T}$ is $\mathbf{M}$-representative of $\mathbf{E}$, and if $\mathbf{M}$ is $\mathbf{T}$-enriched and $\mathbf{T}$-uniform, then given any desired upper bound u on the error rate, $\mathcal{A}$ will output, within time proporational to $\frac{1}{u}$ and inversely proportional to the square of $e(\mathbf{T}, \mathbf{M})$ (the $\mathbf{T}$-enrichment degree of $\mathbf{M}$), a classifier whose expected error rate on $\mathbf{E}$ is less than u.*

The algorithm $\mathcal{A}$ builds classifiers by sampling, with replacement, from the set $\mathbf{M}$, and then combining the "weak classifiers" in the resulting samples. We reduce n-class problems to n-many two-class problems; given a training pair (T_1, T_2) for any such two-class problem, a sample S of size t produces the classifier which assigns any given example q to class 1 if

$$\frac{1}{t}\sum_{S\in\mathbf{S}} \frac{1_S(q) - Pr(S|T_2)}{Pr(S|T_1) - Pr(S|T_2)} > 0.5, \tag{3.3}$$

(where $1_S(q)$ is the indicator function of the set S).

Note that the phrase $\mathbf{M}$-representative in the above theorem, just means that the set of all examples, E_i, of class i is indistinguishable from the set of training examples, T_i, for that class when using the sets in $\mathbf{M}$.

Now, we can combine the two theorems, by building a Stochastic Discrimination graph using Algorithm 2, such that vertices concur with the SD classifier, which allows us to bound the error on edges; i.e. the *unlabeled error rate*. If we can use

Theorem 6 to impose an unlabeled error rate on our semi-supervised algorithm, then the unlabeled error rate, t, of the target function in Theorem 4 can be defined, and thus we can bound the generalization error in terms of the number of unlabeled examples m_u and the number of labeled examples m_l. Note that we are considering this in the context of binary classification for simplicity.

Theorem 7. *If $c^* \in C$ and we define $err_{unl}(c*)$ to be $1 - \chi(f, S)$, where $\chi(f, S) = \boldsymbol{E}_{x \sim S}[\chi(f, x_i, x_j)]$ and S represents pairs of samples defined by a graph constructed using Stochastic Discrimination with expected error $< t$, then $err_{unl}(c*) \leq 2t - 2t^2$ and m_u unlabeled examples and m_l labeled examples are sufficient to learn to error ϵ with probability $1 - \delta$, for*

$$m_u = \frac{2}{\epsilon^2}\left[ln|C| + ln\frac{4}{\delta}\right] \text{ and } m_l = \frac{1}{\epsilon}\left[ln|C_{D,\mathcal{X}}(2t - 2t^2 + 2\epsilon)| + ln\frac{2}{\delta}\right].$$

In particular, with probability at least $1-\delta$, the $f \in C$ that optimizes $\hat{err}_{unl}(f)$ subject to $\hat{err}(f) = 0$ has $err(f) \leq \epsilon$.

Proof. Recall that we defined the unlabeled error rate, $err_{unl}(c*)$, over pairs of samples, and that these pairs were selected (joined) according to the SD algorithm with expected error $< t$, which is possible by Theorem 6 from Kleinberg (2000a). Then, in the binary case, $err_{unl}(c*)$ depends on the number of pairs having only one vertex misclassified, since if both vertices are misclassified, then it does not increase $err_{unl}(c*)$. Therefore, it follows that if the error on the individual vertices is $< t$, then the error on the pairs is $err_{unl}(c*) \leq (1-t)t + t(1-t) = 2t - 2t^2$. The remainder of the proof follows directly from Theorom 4 from Balcan (2008). □

3.4 Graph-based classifiers

3.4.1 Laplacian Eigenmaps

We are focusing on the effects of altering the graph, so we use two different graph based classifiers. The first uses a Laplacian Eigenmap (Belkin and Niyogi, 2003) for dimensionality reduction, followed by construction of a simple linear classifier in the new space in the same manner as in Belkin and Niyogi (2004), in which the coefficients for the new dimensions are set by minimizing the sum of squared error on the labeled data. In other words, the weights of our new dimensions are given by the vector **a** in the following:

$$a = (EE^T)^{-1}Ec \tag{3.4}$$

where c is a vector representing the class labels, the entries of E are $\lambda_k v_{i,k}$, i is the index of the labeled point in the matrix, and k is the index in the new low-dimensional space; i.e. the k-th eigenvalue and eigenvector provide the mapping into the new space for labeled point i. The number of connected components in the graph is determined in order to eliminate the zero-valued eigenvalues, and then the mapping starts with the next eigenfunction.

3.4.2 Out-of-sample extension

The Laplacian Eigenmap approach is inherently transductive, meaning that it only creates a mapping for an unlabeled example if it was part of the set used for graph construction. This means that applying a method transductively would involve solving the eigenvalue problem all over again for any new point or set of points, which would be impractical for most purposes. For an out-of-sample extension that allows efficient application to new points, we utilize the Nystrom Formula as described in Ouimet and Bengio (2005). The method has been shown (and for the most part verified via our own experiments) to provide inductive classification results with no significant difference in accuracy from the transductive application. It simply uses the Laplacian matrix as a data-dependent Kernel function K_D in the following formula in order to map a new point into each dimension k of the new decision space:

$$f_k(x) = \frac{\sqrt{n}}{\lambda_k} \sum_{i=1}^{n} v_{ik} K_D(x, x_i) \tag{3.5}$$

where n is the size of the original dataset, and (λ_k, v_k) are the k-th eigenvalue and eigenvector.

3.4.3 Laplacian RLS/Bayesian Kernel Model

The next semi-supervised model that we focus on is interesting from multiple perspectives. In fact, it is possible to arrive at the same functional form for this model based on two completely different derivations. In other words, this model represents both the Laplacian Regularized Least Squares (Laplacian RLS) model in Belkin et al. (2006) and the Bayesian Kernel Model in Liang et al. (2007a,b); Pillai et al. (2007) with a Dirichlet process prior.

In the case of the Laplacian RLS (Belkin et al., 2006), we are using unlabeled data as a graph-based regularization term. This essentially means that the algorithm penalizes models that assign points that are adjacent in the graph to different classes.

In the case of the Bayesian Kernel Model (Liang et al., 2007a,b; Pillai et al., 2007), a model is estimated by selecting from among functions in the reproducing kernel Hilbert spaces (RKHS) induced by the chosen kernel. We would like to assume that we have a smooth function that we want to represent. We can look at our kernel as data that fall on a smooth manifold; i.e. that points in the original space actually vary along a dense manifold that cuts through that space.

We will see that these assumptions can also lead us to the same functional form as the Laplacian RLS. The derivation can be found in Liang et al. (2007a,b); Pillai et al. (2007). The relevant Bayesian kernel derivation is based on integral operators. The form that is used in Liang et al. (2007a) is the following:

$$f(x) = \int k(x,u)d\gamma(u) = \int k(x,u)w(u)dF(u) \tag{3.6}$$

F is the unknown distribution of the kernel knots, u, where a knot is a data point on the manifold. In essence this means that F can be set to correspond to the marginal distribution of the data, X.

Given a fixed sample from an uncertain distribution F in the Dirichlet process (DP) model, the posterior is the following Dirichlet process Liang et al. (2007a):

$$F|X_n \sim DP(\alpha + n, F_n), F_n = (\alpha F_0 + \sum_{i=1}^{n} \delta_{x_i})/(\alpha + n) \tag{3.7}$$

In Liang et al. (2007a), we see that if we want to predict the value of a new point, x, based on our sample from F, i.e. based on our labeled and unlabeled training data, then we want the following:

$$E[f|X_n] = a_n \int k(x,u)w(u)dF_0(u) + n^{-1}(1 - a_n) \sum_{i=1}^{n} w(x_i)k(x, x_i) \tag{3.8}$$

where $a_n = \alpha/(\alpha + n)$.

Then, assuming an uninformative prior, they take the limit $\alpha \to 0$ to get the following representer form:

$$\hat{f}_n(x) = \sum_{i=1}^{n} w_i k(x, x_i) \tag{3.9}$$

Thus, according to Liang et al. (2007a,b); Pillai et al. (2007), when the uncertainty about the probability distribution function for the data, X, is expressed using a Dirichlet process prior, then the function f can be approximated by the following formula over labeled and unlabeled examples.

$$\hat{f}(x) = \sum_{i=1}^{n} w_{n,i} K(x, x_i) + \sum_{i=1}^{n_m} w_{n+n_m,n+i} K(x, x_i^m) \tag{3.10}$$

This results in the exact same functional form as that derived in Belkin et al. (2006). And in fact, the graph Laplacian over the observed data can then approximate the Laplacian on the manifold by solving the following.

$$\hat{f}(x) = \text{argmin}_{f \in \mathcal{H}_K} \left[\frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i) + \gamma_A \|f\|_K^2 + \frac{\gamma_I}{(n+n_m)^2} f^T L f \right] \tag{3.11}$$

where L is the Laplacian derived from the data and $f = \{f(x_1), ..., f(x_n), f(x_1^m), ..., f(x_{n_m}^m)\}$. γ_A and γ_I are parameters that control the amount of regularization in the ambient space and intrinsic space, respectively.

3.4.4 Laplacian RLS Model Implementation

So, once again, we can use a method of semi-supervised learning using the graph Laplacian. In our experiments, we use the unnormalized form of the graph Laplacian here as well (Equation 2.2).

The output function that is learned is the following:

$$f(x) = \sum_{i=1}^{l+u} \alpha_i K(x_i, x), \tag{3.12}$$

where K is the $(l+u) \times (l+u)$ Gram matrix over labeled and unlabeled points, and α is the following learned coefficient vector:

$$\alpha = (JK + \gamma_A l I + \frac{\gamma_I l}{(l+u)^2} LK)^{-1} Y, \tag{3.13}$$

with L being the Laplacian matrix described above, I being the $(l+u) \times (l+u)$ identity matrix, J being the $(l+u) \times (l+u)$ diagonal matrix with the first l diagonal entries equal to 1 and the rest of the entries equal to 0, and Y being the $(l+u)$ label vector, $Y = [y_1, ..., y_l, 0, ..., 0]$. See Belkin et al. (2006) for details.

This method does have two parameters that control the amount of regularization. For all of our experiments, we use the following parameters, as suggested for manifold regularization in Belkin et al. (2006): $\gamma_A l = 0.005$, $\frac{\gamma_I l}{(l+u)^2} = 0.045$.

3.5 Experiments

A variety of experiments were constructed to demonstrate the effect of altering the graph in the manner proposed. The experiments cover several different problem sets with a variety of parameters in an effort to evaluate the robustness of the proposed methods. For comparison purposes, we use the standard Laplacian Eigenmap (LEM) (Belkin and Niyogi, 2004) and the standard Laplacian RLS (Belkin et al., 2006), where the graph is constructed using unlabeled data only. We use cosine similarity as our distance measure, since it outperformed Euclidean distance in all of our experiments. We denote the approach with the random-subspace augmented graph as LB-LEM and LB-LapRLS in the experimental results. LB stands for label-biased, since we are essentially choosing a bias for our graph-based learning algorithms based on the labeled data. We refer to the combination of the SD-graph-construction method with the LapRLS as SD-LapRLS.

3.5.1 Brain-computer interface

The Brain-Computer Interface (BCI) problem comes from a collection of electroencephalography (EEG) recordings using 39 probes. The goal is to determine whether the human subject was concentrating on moving their right or left hand during the monitoring process. This is an extremely noisy dataset, and one in which the use of the unlabeled data alone is very unsuccessful in discovering a good predictive space. More details can be found in Chapelle et al. (2006). Note that the BCI dataset was identified in Chapelle et al. (2006) as one in which it is very difficult to improve over the supervised baseline obtained using an SVM (error: 34.31%; AUC: 71.17%).

The experiments in Table 6.1 were conducted using the method described in Belkin and Niyogi (2004), with the only difference being in the construction of the graph. For each of the methods described, we ran 10 experiments in which 100 labeled samples were randomly selected and the remaining 300 samples were used as unlabeled data. The unlabeled data were then used as the transductive test set. In addition, in

Table 3.1: Average error of Laplacian-Eigenmap-based classifiers on BCI data.

Graph construction method	Average Error
Unlabeled data graph (LEM)	0.4856
Random subspace label-biased graph (LB-LEM)	**0.3086**
Top 10 feature label-biased graph	0.4596
Top 20 feature label-biased graph	0.4673
Top 30 feature label-biased graph	0.4746

Table 3.2: Average error of Laplacian RLS classifiers on BCI data.

Model Building Condtions	Average Error	AUC
Standard LapRLS	0.3244	0.7431
Standard LapRLS*	0.3136	0.7483
LB-LapRLS	**0.2697**	**0.8083**
SD-LapRLS	0.2750	0.7894

*LapRLS results in Chapelle et al. (2006), obtained using model selection, a normalized graph Laplacian, and an RBF base kernel; which was the best result among all 11 semi-supervised algorithms tested in the benchmark.

order to demonstrate that some simple methods for using the labeled data in the graph construction can have significant robustness problems, we ran a third version of experiments with a feature-selection-based graph construction process. In this approach, we select the top features (out of 117 total) as ranked using a combination of mutual information and fisher criterion, as in Dhir and Lee (2009). Table 3.2 shows a comparison of results across the 12 data splits from the benchmark set in Chapelle et al. (2006), where the LapRLS performed the best out of all methods in the benchmark. We see that our simple approach can improve even these results. The SD-LapRLS was built with $\beta = 0.05$, $\lambda = 5$, and using 1000 weak classifiers. Although the SD-graph results are very good, model selection was required to obtain them, and the more general random subspace approach performs better, while having the advantage of being trivially parallelizable.

3.5.2 Text categorization

Text categorization is a common high-dimensional classification task that does not seem to fit the manifold assumption. Text data is generally thought to be cluster-like instead. Therefore, it is an interesting test for manifold-based techniques, and one in which we might assume that nonlinear manifold learning methods might perform poorly. The task is a binary classification task from the well-known 20 newsgroup data. The two categories tested here are the atheism and the religion newsgroups. The data was preprocessed to remove headers and stopwords, and the terms were stemmed using the Porter stemming algorithm. Finally, the stems were used as features with values determined by their TF-IDF scores. This task had 1424 documents. Ten random experiments were carried out in which selections of one thousand of the documents were used for the graph, and the remaining 424 data points were used for the out-of-sample tests. Eight nearest neighbors were used in the graph construction.

The advantage one gains over the purely unsupervised graph construction by using the labels to bias the graph should increase with the number of labeled data available. Table 6.2 demonstrates this advantage on the text categorization task. While this also shows that there is some risk to using this method with very few labels, it is more likely the ratio of the number of labeled data to the size of the ambient feature space that leads to a requirement for a larger labeled set. The number of dimensions retained is kept at 20% of the number of labels, as suggested in Belkin and Niyogi (2004). Also note that the semi-supervised approaches clearly outperform the purely supervised linear SVM (which is typically very good for text categorization).

As expected, the accuracy of the classifier rises as the percentage of graph edges that join two examples from the same class increases. This is significant because it confirms that the hypotheses under consideration in the learning process are indeed being restricted in a way that matches our notion of compatibility, χ, as discussed above.

Table 3.3: Binary text categorization

		Avg. Error Rates	
Algorithm	Labels	transductive	out-of-sample
LEM	100	0.2253	0.2120
	200	0.1954	0.1882
	300	0.1849	0.1724
LB-LEM	100	0.2951	0.2538
	200	0.2064	0.1861
	300	0.1730	0.1649
Linear SVM	100	0.3047	0.3009
	200	0.2268	0.2300
	300	0.1870	0.1913

Table 3.4: Average edge correctness.

	LEM	LB-LEM		
		100	200	300
Accuracy	75.5	77.3	81.9	85.0

3.5.3 Dimensionality reduction in hyperspectral image analysis

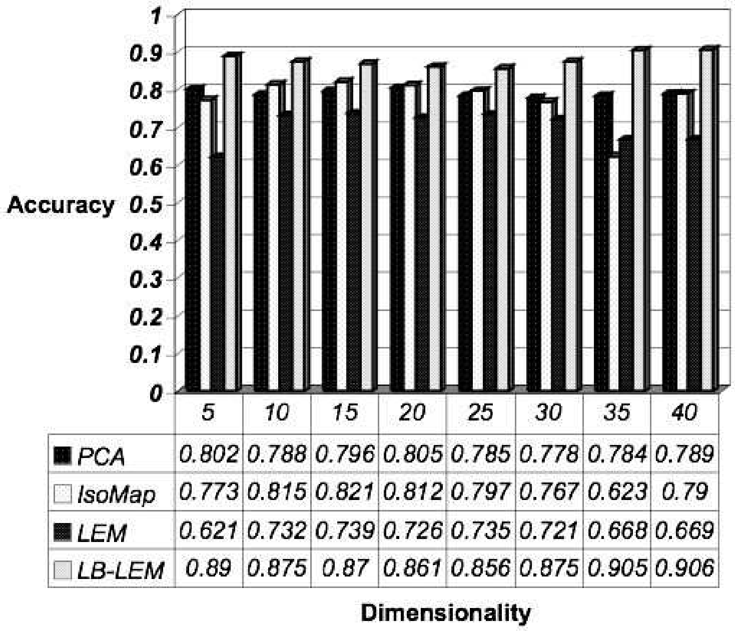
Land cover classification by hyperspectral image (HSI) data analysis has become an important part of remote sensing research in recent years Landgrebe (2002). Compared to conventional multi-spectral images where each pixel usually contains tens of bands, pixels in hyperspectral images usually consist of more than a hundred spectral bands, providing fine-resolution spectral information. Classification techniques for this multiclass application need to handle high-dimensional, high-resolution data. Obtaining ground truth is another challenge, since HSI can cover very large areas and it is not usually possible to obtain highly accurate class labels for all locations in the image.

A Hyperion hyperspectral image taken from Okavango Delta, Botswana in May 2001 is used for the experiments. The acquired data originally consisted of 242 bands, but only 145 bands are retained after preprocessing. The area used for the experiments has 1476×256 pixels with 30m spatial resolution. We used two spatially disjoint class maps from the same geographical region, and there are 9 classes in total. The training set consists of 1580 labeled instances, and the test set has 1434 instances.

For these experiments, we treat the problem as one of purely transductive learning. In other words, we use the test points to influence the dimensionality reduction. The transductive approach is practical in this case since we want to classify unlabeled portions from the same image or geographic region. For classification based on the dimensionality reduction, we employed a maximum-likelihood classifier (MLC) with Gaussian distribution, where the class-conditional distribution of each class is modeled as a multi-variate Gaussian: $p(\mathbf{x}|y_i) \sim \mathcal{N}(\mu, \Sigma_i)$. The mean and covariance of each distribution are measured by maximum-likelihood estimation. Given a test data point, the MLC outputs the class label with maximum posterior probability, $y = \arg\max_i P(y_i|\mathbf{x}) \propto \arg\max_i p(\mathbf{x}|y_i)P(y_i)$. The prior probability distribution $P(y_i)$ is measured emipirically from the training set.

In this case, we compare effects of the dimensionality reduction performed by our method (LB-LEM) to that of the standard Laplacian Eigenmap method (Belkin and Niyogi, 2003), as well as dimensionality reduction performed via an alternative non-linear dimensionality reduction method, ISOMAP (Tenenbaum et al., 2000), and the standard linear principle component analysis (PCA).

As shown in Figure 3.2, the modifications in the graph have a very positive effect on the classifier. In addition, it appears that the use of the domain knowledge through the labels incorporates some robustness into the selection of the appropriate dimensionality, as shown in Figure 3.3.



	5	10	15	20	25	30	35	40
■ PCA	0.802	0.788	0.796	0.805	0.785	0.778	0.784	0.789
□ IsoMap	0.773	0.815	0.821	0.812	0.797	0.767	0.623	0.79
■ LEM	0.621	0.732	0.739	0.726	0.735	0.721	0.668	0.669
▨ LB-LEM	0.89	0.875	0.87	0.861	0.856	0.875	0.905	0.906

Figure 3.2: Classification accuracy on Botswana data using dimensionality reduction with ML classifier.

3.6 Summary of label-guided graph construction

We have outlined robust methods for biasing the graph construction in relevant semi-supervised learning methods and demonstrated their effectiveness. There are many obvious ways in which to optimize these general approaches above and beyond their basic descriptions, but the focus here has been on determining a general approach to providing robustness that does not require a lot of effort for manually incorporating domain knowledge. Since we utilize the labels more effectively as a source of domain knowledge, we can notice clear improvements when the feature set is not extensively hand engineered and is therefore noisy with respect to the target problem. These improvements are realized without requiring any significant increase in effort for domain adaptation.

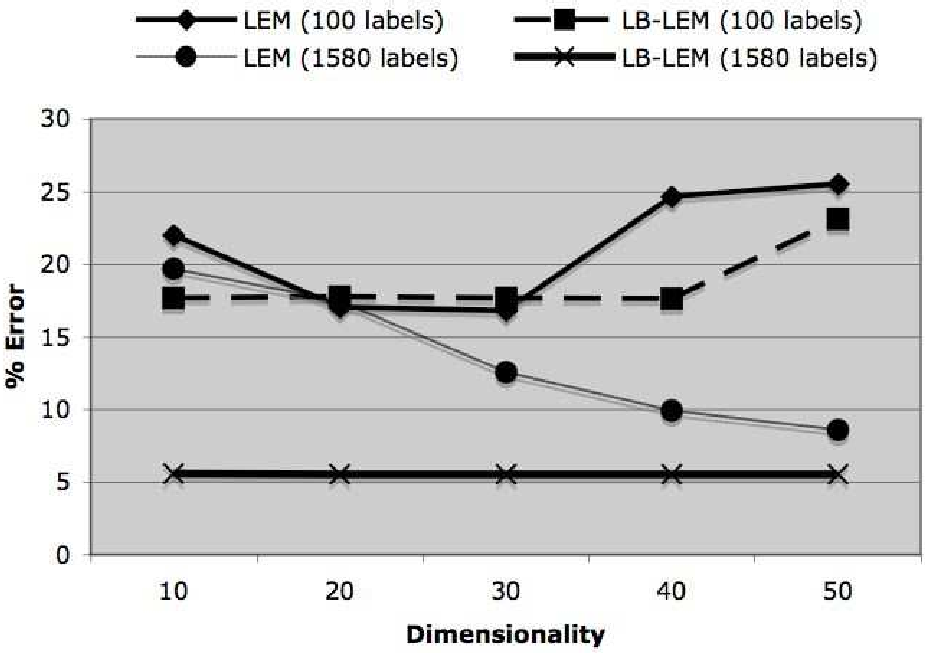


Figure 3.3: Change in classification error on Botswana data based on number of dimensions retained.

Chapter 4

Graph-based multi-modal learning

4.1 Traditional data fusion

Traditional data fusion methods (Nakamura et al., 2007; Corona et al., 2009) involve dealing with data in stages, creating ensembles of learners, or creating complex features by combining observations from different modalities. Unfortunately, while the real power of fusing information may be found in combining features directly, such an approach must deal with the sample complexity problem as the number of features being considered increases. All of the above approaches are inadequate when one wishes to carry over information from modalities that will not be available when the model is applied. This chapter discusses such applications and explores new methods of combining data to get around this issue. Some of the work in this section was published in Symons and Arel (2011).

4.2 Budgeted learning: Addressing missing modalities and limited ground truth

Budgeted learning under constraints on both the amount of labeled information and the availability of features at test time pertains to a large number of real world problems. Ideas from multi-view learning, semi-supervised learning, and even active learning have applicability, but a common framework whose assumptions fit these problem spaces is non-trivial to construct. We leverage ideas from these fields based on graph regularizers to construct a robust framework for learning from labeled and unlabeled samples in multiple views that are non-independent and include features that are inaccessible at the time the model would need to be applied. We describe example applications that fit this scenario, and we provide experimental results to demonstrate the effectiveness of knowledge carryover from training-only views.

As learning algorithms are applied to more complex applications, relevant information can be found in a wider variety of forms, and the relationships between these information sources are often quite complex. The assumptions that underly most learning algorithms do not readily or realistically permit the incorporation of many of the data sources that are available, despite an implicit understanding that useful information exists in these sources. When multiple information sources are available, they are often partially redundant, highly interdependent, and full of noise (and other information that is irrelevant to the problem under study). In this paper, we are focused on a framework whose assumptions match this reality, as well as the reality that labeled information is usually sparse. Most significantly, we are interested in a framework that can also leverage information in scenarios where many features that would be useful for learning a model are not available when the resulting model will be applied.

As with constraints on labels, there are many practical limitations on the acquisition of potentially useful features. A key difference in the case of feature acquisition is that the same constraints often don't pertain to the training samples. This difference provides an opportunity to allow features that are impractical in an applied setting to nevertheless add value during the model-building process. Unfortunately, there are few machine learning frameworks built on assumptions that allow effective utilization of features that are only available at training time. In this paper we formulate a knowledge carryover framework for the budgeted learning scenario with constraints on features and labels. The approach is based on multi-view and semi-supervised learning methods that use graph-encoded regularization. Our main contributions are the following: (1) We propose and provide justification for a methodology for ensuring that changes in the graph regularizer using alternate views are performed in a manner that is target-concept specific, allowing value to be obtained from noisy views; and (2) We demonstrate how this general set-up can be used to effectively improve models by leveraging features unavailable at test time.

4.2.1 Scaling issues in multi-view learning

Large-scale, multiple-evidence learning is a growing area of importance for machine learning practitioners. Some of the increased focus on large-scale multi-view problems might be attributed to the general rise of semi-supervised learning, which has demonstrated significant gains in performance when augmenting supervised learning with large amounts of unlabeled data. The major impetus of much progress in semi-supervised learning came out of work on a natural multi-view problem Blum and Mitchell (1998). Moreover, as the use of semi-supervised techniques has become widespread, the data that practitioners look to incorporate is often larger and has more complex inter-relations, often including potentially separable self-sufficient views. Although the field is nascent, one of the more prominent areas of study in multi-view, semi-supervised learning involves graph-based methods Sindhwani et al. (2005a); Sindhwani and Rosenberg (2008). Unfortunately, as seen in Chapter 3, most of these methods are non-parametric or semi-parametric Orbanz and Teh (2010), which can complicate large-scale learning, due to the fact that the model's complexity is data-dependent. Another complicating factor involved in processing many natural multi-view datasets is the inconsistent availability of all views, in which case an application might require techniques for handling missing data Marwala (2009) or might be viewed as a budgeted learning problem Greiner et al. (2002); Cesa-Bianchi et al. (2009).

Examples of natural multiple evidence problems abound. For example, in the case of web page analysis, we often have both text and images, or video and audio. In medical domains, particularly as the number of multiple-modality clinical studies increases, we often have several interdependent, partially redundant views of a condition, characterized by multiple imaging technologies, textual descriptions of symptoms, blood assays, etc. Even in fields as well studied as satellite image analysis, the current ubiquity of information collection often means that we now have multiple imaging methods, ground sensor data, etc. covering a given location and time. Unfortunately, one inconvenient reality that all of these applications share is the high likelihood that an inconsistent set of views will be available for different examples. In fact, often only a subset of views or a single view will be available when applying a decision model learned across multiple evidential views. For example, while many training examples may contain text and images, we may not have both when we wish to index a new web page. In medical studies it has become increasingly common to study multiple diagnostics in an attempt to find a single modality test that outperforms the others or is most cost-effective. In other words, the search for a multi-modality diagnostic is often only a secondary goal when cost is a driving factor. In the case of satellite image analysis, ground sensor data might be available for many training samples, but when applying a model to alternate localities, this secondary evidence is often missing. See Figure 4.1 for a conceptualization of two such applications.

Thus, a large number of multiple evidence learning scenarios can benefit from knowledge carryover. In this case, we view the problem as one of budgeted learning, but unlike the scenario in Greiner et al. (2002); Cesa-Bianchi et al. (2009), we usually do not have control over which features or views are available at test time. Thus, standard data fusion techniques are not appropriate, since we want knowledge learned from alternate modalities to benefit a single-modality model. In addition, we are usually constrained by the number of labeled examples, such that most large-scale multi-view problems fall into the domain of semi-supervised learning as well. In

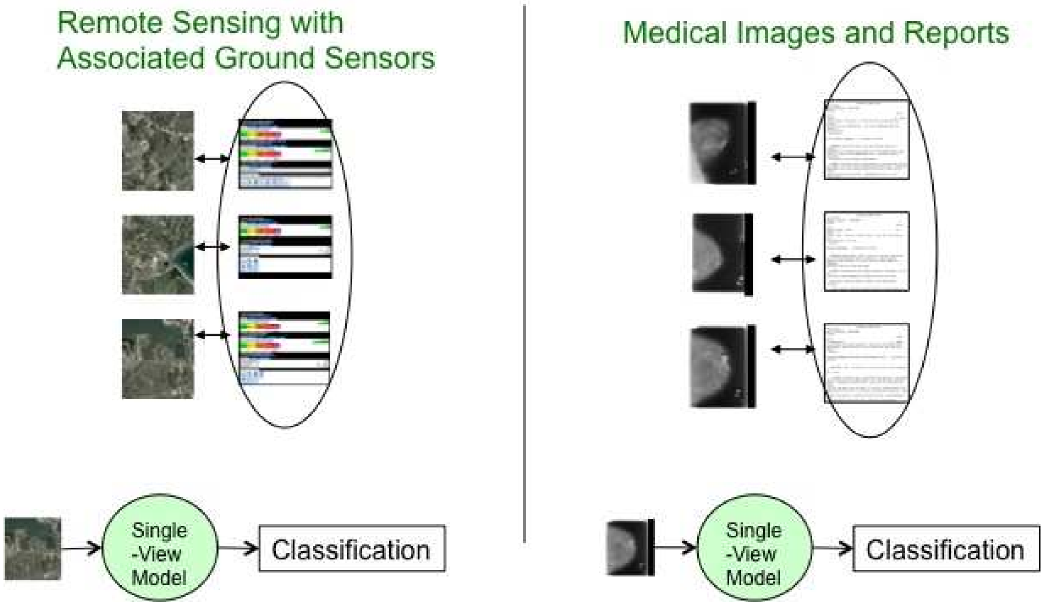

Remote Sensing with Associated Ground Sensors
Single -View Model
Classification
Medical Images and Reports
Single -View Model
Classification

Figure 4.1: Conceptualization of multi-view budgeted learning.

these cases, we are often particularly concerned with the *curse of dimensionality.* Some traditional data fusion techniques applied to multiple evidence learning would rely on exploration of combinations of features from different views. In addition to increasing the dimensionality relative to a static number of labeled training samples, these methods cannot benefit from alternate views unless they are available when the model is applied. Ensemble fusion techniques that build separate models for each view cannot take advantage of the links between modalities provided by the examples, and these methods also cannot benefit from training-only evidence.

The benefits of the approach we will outline are many, including the following:

- It is possible to utilize powerful features and views (self-sufficient feature subsets) to improve classifiers that will not have access to such features. This systematically addresses the real-world issue of training-only views, a little explored subfield of machine learning.

- It is a naturally semi-supervised method that is able to take advantage of large amounts of unlabeled data.

- It is possible to fully leverage incomplete data (features that are missing from some of the training examples) without data imputation.

- The method is applicable to a broad range of learning algorithms that use graphs for regularization or dimensionality reduction.

- It is a particularly forgiving method when the relevance or importance of many features is not clearly known.

4.2.2 Multi-view regularization

As discussed above, constraints imposed on feature acquisition come from a variety of sources and can be found in many real-world applications. Often these problems fit naturally into a multi-view setting, in which feature sets can be reasonably partitioned into disjoint, cohesive sets that are somewhat redundant. For example, in a medical study where a battery of tests is performed, and results from all of these procedures are available for subjects of the study (the training set), an applied diagnostic that is dependent upon all of these tests would be prohibitively expensive. Therefore, the standard approach is to independently consider the separate tests to find one procedure that seems to be the best diagnostic, while inter-related, useful information from the rest of the study typically remains unused. The framework presented here allows such training-only observations to effectively improve a model that operates on a feature set that does not include these observations.

We view this as a budgeted, multi-view learning problem. Since most applications that fit this scenario will also have few labeled and many unlabeled examples, we also treat the problem as semi-supervised. We assume that we have l labeled examples, $\{(x_i, y_i)\}_{i=1}^{l}$ and u unlabeled examples, $\{x_i\}_{i=l+1}^{l+u}$. In addition, we have j distinct views of each example, $x_i = (x_i^1, x_i^2, ..., x_i^j)$. For simplicity, we will assume $y \in \{+1, -1\}$, with multi-class classifiers being constructed of multiple one-vs-all binary classifiers. To facilitate discussion, we will refer to a feature set available at both training and test time as a *primary view*, and we will refer to a feature set available in training only as a *secondary view*. For example, satellite image features could be a primary view used for classification, and corresponding ground-sensor features could be a secondary view not available when the model would be applied but that encapsulates useful information about some or all of the training examples.

Related work on semi-supervised, multi-view regularization (Sindhwani et al., 2005a; Sindhwani and Rosenberg, 2008) employs two regularizers, but the regularization using the unlabeled examples is purely unsupervised. The result is that even though a tradeoff can be made and the unlabeled regularization can be de-emphasized, in cases where a feature set is particularly noisy with regard to the target concept, it is difficult to obtain benefit from such a view, particularly if the labeled set is small. In contrast, we demonstrate how changes in the construction of the regularizer that are informed by the labeled information, as outlined in Chapter 3, provide benefit consistently. In particular, it becomes safer to include such information, since it is unlikely to make the regularization less effective than single-view, semi-supervised regularization.

4.2.3 Random subspace multi-view smoothing

The framework we employ is one based on semi-supervised learning using graph-based regularization across separate views. When attempting to use secondary views in the construction of a graph regularizer, one must be cognizant of the potential for the secondary information to be much noisier than the primary information. Therefore, just as in Chapter 3, we need a method to alter the graph that attempts to use only the concept-relevant information from the secondary views. In the general setup we are addressing, the only information we have about the target concept is from the labels.

We utilize the method based on random subspace selection from Chapter 3 in order to allow the views to influence the graph construction in a robust way that reflects information about the target problem. We use the classifications of many classifiers from different subspaces and various views to find a target-relevant similarity. Although one must specify the number of neighbors to retain and the number of splits to use in the graph construction (see Algorithm 3), the method seems to be rather robust with regard to these choices. In the method that we employ in our experiments, for each view, we perform $s = 100$ random feature splits, resulting in two hundred classifiers that are trained on the labeled data based on the subset of features that represent their hypothesis space. We then construct a nearest neighbor graph based on cosine similarity in the primary view while weighting the similarity scores based on the percentage of shared classifications using theses random subspace models. We then build the unnormalized graph Laplacian.

Algorithm 3 Multi-View Random Subspace Graph Construction

```
Input: data $\{(x_i, y_i)\}_{i=1}^{l}$, $\{x_i\}_{i=l+1}^{n}$ in each view, $numNeighbors := k$, $numFeatureSplits := s$
for each view $v_j$ do
  for $i = 1$ to $s$ do
    Randomly split feature set into two equal parts
    Train linear classifier on each part
    Classify each sample point using the classifiers
  end for
end for
for $u = 1$ to $n$ do
  for $v = 1$ to $n$ do
    $w_{u,v} = \frac{\sum_{i=1}^{m} u_i v_i}{\sqrt{\sum_{i=1}^{m} u_i u_i}\sqrt{\sum_{i=1}^{m} v_i v_i}} \times$ (percent of time classified together)
  end for
end for
Retain $k$ nearest neighbors; create Laplacian matrix $L(u, v)$
```

Based on common practice from related literature, we set the number of nearest neighbors, k, equal to 8. We test the effect of the graph changes using the two different base algorithms that use the Laplacian matrix of the graph described in Chapter 3. Once again, the first learning algorithm is the Laplacian Eigenmap (LEM) approach described in Belkin and Niyogi (2004), and the second is the Laplacian Regularized Least Squares (LapRLS) approach described in Belkin et al. (2006).

4.3 Transductive Laplacian eigenmap classifier

Again, as described in Belkin and Niyogi (2004), we construct a linear classifier in a new space that allows transductive classification. The coefficients for the new dimensions are set by minimizing the sum of squared error on the labeled data. In other words, the coefficient vector **a** is obtained using the following equation:

$$a = (EE^T)^{-1}Ec \tag{4.1}$$

where c is a vector representing the class labels and the entries of E are the eigenfunctions of the Laplacian matrix, $\lambda_k v_{i,k}$; i is the index of the labeled point in the matrix, and k is the index in the new low-dimensional space. The mapping starts with the eigenvector associated with the first non-zero eigenvalue, and includes as many eigenvectors as the number of dimensions desired.

4.3.1 LEM Out-of-Sample Extension

For our out-of-sample extension, we again use the Nystrom Formula as described in Ouimet and Bengio (2005). It employs the Laplacian matrix as a data-dependent Kernel function K_D in the following formula in order to map a new point into each dimension k of the new decision space:

$$f_k(x) = \frac{\sqrt{n}}{\lambda_k} \sum_{i=1}^{n} v_{ik} K_D(x, x_i) \tag{4.2}$$

where n is the size of the original dataset, and (λ_k, v_k) are the k-th eigenvalue and eigenvector.

4.3.2 Multi-View Laplacian Regularized Least Squares

Note that the Laplacian Regularized Least Squares (LapRLS) (Belkin et al., 2006) algorithm uses two regularizers, including the graph Laplacian. However, our modifications to the regularization only affect the unlabeled-data regularization through the Laplacian matrix. We use the graph construction method described in Algorithm 3 above to produce the multi-view-derived Laplacian matrix that is used here. In this case the output function that is learned is the following:

$$f(x) = \sum_{i=1}^{l+u} \alpha_i K(x_i, x), \tag{4.3}$$

where K is the $(l+u) \times (l+u)$ Gram matrix over labeled and unlabeled points, and α is the following learned coefficient vector:

$$\alpha = (JK + \gamma_A l I + \frac{\gamma_I l}{(l+u)^2} LK)^{-1} Y, \tag{4.4}$$

with L being the Laplacian matrix described above, I being the $(l+u) \times (l+u)$ identity matrix, J being the $(l+u) \times (l+u)$ diagonal matrix with the first l diagonal entries equal to 1 and the rest of the entries equal to 0, and Y being the $(l+u)$ label vector, $Y = [y_1, ..., y_l, 0, ..., 0]$.

The modifications we employ are all during the graph construction phase. This means that we can train a LapRLS learner using a primary view in a straightforward manner since the secondary view information is encoded into the regularization term, $\frac{\gamma_I l}{(l+u)^2} LK$, via the matrix, L. While the LapRLS avoids the need to select the number of dimensions as in the Laplacian Eigenmap approach, it does have its own parameters that control the effect of the unlabeled data. For all of our LapRLS experiments, we use the following parameters, as suggested for manifold regularization in Belkin et al. (2006): $\gamma_A l = 0.005$, $\frac{\gamma_I l}{(l+u)^2} = 0.045$.

4.4 Theoretical discussion

The theoretical analysis in Section 3.3 can be applied to graph construction in multiple views without modification. In other words, since compatibility is defined as an expectation over samples, in the multi-view setting the same theoretical arguments hold if the graph encoded notion of compatibility is derived from alternate views in addition to unlabeled data.

4.5 Experimental Results

In order to demonstrate the effectiveness of the approach, we utilize the Multiple Features Dataset, which is available through the UCI Machine Learning Repository (Asuncion and Newman, 2007). This dataset is an image recognition task over 2000 handwritten digits. It is a 10-class problem containing 200 examples of each digit. Each example is composed of 6 distinct features sets, which we use as 6 separate views; one primary and five secondary. The views are the following: 240 pixel averages in 2×3 windows (Pix); 216 profile correlations (Fac); 76 Fourier coefficients of the digit shapes (Fou); 64 Karhunen-Loéve features (Kar); 47 Zernike moments (Zer); and 6 morphological features (Mor). Additional information on the dataset can be found in Van Breukelen et al. (1998). Each of our experimental results is an averaged error measurement over 10 random splits of the data into a semi-supervised training set of 100 labeled examples and 900 unlabeled examples and a separate test set of 1000 examples for inductive testing. Table 4.1 shows a comparison of single view learning methods. When the training and test features are the same, it indicates that a single view was used; i.e. either the pixel features (Pix) or all features combined into a single feature set. In the case of the Multi-View Laplacian Eigenmap (MV-LEM), the views are treated separately, with pixel features being the only ones used for inductive testing and the other views providing information through the graph construction process as described above. It is interesting to note that the best performance is obtained with a model that only has access to the pixel values at test time and that the standard Laplacian Eigenmap approach does not improve with straightforward addition of the other features. The best performance is obtained by a careful approach that recognizes the potentially redundant information across multiple views and the

Table 4.1: High-level classifier comparison: A linear SVM is trained on a single view consisting of just pixel features or all features combined into a single set. The same two sets are used for a LEM classifier, including a version that uses random subspace smoothing based on all features as a single set. This is compared to the method in this paper (MV-LEM), using all features for training, but only pixel features at test time. Transductive results are provided in addition to the inductive results on the set-aside test set.

	Features Used		Average Error Rate	
Classifier	Training	Testing	Transductive	Inductive
Linear SVM	Pix	Pix	.099	.098
Linear SVM	All	All	.081	.083
Laplacian Eigenmap	Pix	Pix	.083	.068
Laplacian Eigenmap	All	All	.085	.068
Single-View RS Laplacian Eigenmap	All	All	.111	.081
Multiview Laplacian Eigenmap	All	Pix	.073	.057

hard realities one must face when attempting to include more features without the ability to increase the size of the labeled data.

Table 4.2 and Table 4.3 compare the effect of different graph construction methods using the LEM learner and the LapRLS learner. The smoothing of the graph takes one of the following three forms: no smoothing; random subspace smoothing via the primary view, or the cumulative effect of random subspace smoothing via all views. In this case, regardless of the method and regardless of the primary view, the addition of information from the other views during training always provides a significant level of improvement to the classifier that operates on the primary view only. We do not include the morphological features as a primary view, since the 6 features are not sufficient to generate useful models for this 10 class problem.

Table 4.2: Knowledge carryover comparisons using Laplacian Eigenmaps. Classification is always performed using a single primary view. Each classifier uses the graph Laplacian for dimensionality reduction, retaining 40 eigenfunctions. Graph construction uses no smoothing (None), random subspace smoothing based on the primary view only (Prime RS), or the cumulative effect of random subspace smoothing using all of the views (Both RS).

	Features Used		Average Error Rate	
Graph Smoothing	Training	Testing	Transductive	Inductive
None	Pix	Pix	.083	.068
Prime RS	Pix	Pix	.095	.059
Both RS	All	Pix	.064	.054
None	Fac	Fac	.136	.124
Prime RS	Fac	Fac	.122	.101
Both RS	All	Fac	.072	.087
None	Fou	Fou	.307	.301
Prime RS	Fou	Fou	.316	.308
Both RS	All	Fou	.063	.227
None	Kar	Kar	.128	.114
Prime RS	Kar	Kar	.122	.092
Both RS	All	Kar	.065	.076
None	Zer	Zer	.276	.256
Prime RS	Zer	Zer	.276	.256
Both RS	All	Zer	.064	.203

4.6 Summary of multi-vew, budgeted learning results

In this chapter, we have demonstrated the use of principles from multi-view and semi-supervised learning for budgeted learning in the face of realistic constraints on the availability of both features and labels. Our experiments clearly show consistent improvement when the only difference in training is the use of secondary views (not available at test time) to modify the Laplacian matrix used for regularization or dimensionality reduction based on the approach outlined in this chapter. The general

Table 4.3: Knowledge carryover comparisons using LapRLS. Classification is always performed using a single primary view. Graph construction uses random subspace smoothing based on the primary view only (Prime RS) or the cumulative effect of random subspace smoothing using all of the views (Both RS).

	Features Used		Average Error Rate	
Graph Smoothing	Training	Testing	Transductive	Inductive
Prime RS	Pix	Pix	.185	.213
Both RS	All	Pix	.162	.195
Prime RS	Fac	Fac	.113	.107
Both RS	All	Fac	.070	.078
Prime RS	Fou	Fou	.337	.349
Both RS	All	Fou	.318	.332
Prime RS	Kar	Kar	.180	.181
Both RS	All	Kar	.168	.172
Prime RS	Zer	Zer	.331	.329
Both RS	All	Zer	.304	.307

framework is one that also supports the insertion of expert knowledge via feature-based active learning. For example, it would be possible to use such an approach to filter the features used in the secondary views.

Chapter 5

Multi-task, semi-supervised learning for large-scale analysis of complex systems

5.1 Barriers in complex systems analysis

Coordinated knowledge discovery across diverse data at very large scales is extremely difficult. When the number of potentially predictive or important variables is at such scales, the severe ratio of potentially useful observations to known outcomes (i.e. the sample complexity side of the curse of dimensionality) is even more of a problem than lack of computational power. Ultrascale datasets offer some hope of accounting for the importance of complex variables, but there are no general techniques or libraries that can effectively utilize such large-scale data for knowledge discovery. This chapter discusses initial work to fill the need for such data analysis tools by engineering a set of solutions around an analytical approach that thrives in the presence of inter-related large-scale data.

There is often an overwhelming number of atomic observables/measurements to begin with in large systems. Despite this, linear and non-linear feature combinations, observations that encode temporal (and in many cases spatial) dependencies, etc. are all very important, such that atomic variables often mean almost nothing, while more complex features are quite revealing (take temporal behavior patterns in cyber security for example). For example, in climate data analysis, the combinatorics of teleconnections alone is enough to necessitate ultrascale computing. Current techniques are completely inadequate for the analysis of predictive correlation in such scenarios, and therefore most analyses focus on correlations within a subset of known phenomena, and the ability to discover the unknown is severely handicapped.

We have built a set of core algorithms that can use large-scale coordinated predictive analysis across seemingly inter-related portions of the data in order to isolate noise from potentially predictive observations. Coordinated knowledge discovery is enabled by analyzing observables and outcomes as sets of prediction tasks, allowing use of all of the data for the filtering of noise and the discovery of potentially important observations. The general basis for the framework comes out of statistical machine learning with certain guarantees on the generalization ability achievable based on the application of the uncovered observations in a predictive setting (see Ando and Zhang (2005)). In essence, this is an ideal framework for isolation of noise and ranking of important patterns with solid theoretical foundations. Moreover, it is very amenable to improvement through automatic problem construction/exploration, it is designed to take advantage of inter-relations across large-scale analyses, and it naturally provides a scalable analysis scheme.

The advantages that allow parallelization and ultrascale concurrent processing are the following: 1) The initial learning phase allows a natural division of the data into separate problems spaces. 2) The subspaces searched by each problem can be set based on (loose) correlation to each analysis task. Therefore, these spaces will overlap, but not be all encompassing, allowing intelligent division of the data. 3) The phase that is potentially difficult to parallelize is one in which hypothesis spaces are compared at large scale. However, the objective is to discover observations that are representative of effective decision spaces, and in this setting it turns out that linear analysis techniques like Principle Component Analysis (PCA) are appropriate. In other words, while PCA is not good at finding discriminative structures in a single problem analysis, it is very useful for finding them here, since it can be used to find observations that represent good discriminators across many of the individual analyses (at least enough to rise above a noise threshold). The ability to use techniques like PCA is important, because it can be exactly translated into a particular ”summation form” that techniques like Map-Reduce can exploit for essentially linear scalability on multi-core platforms (Chu et al., 2006). Exploration of non-linearities can be conducted via the feature construction for the individual problems, which allows a seamless fit with the parallel analysis structure presented here.

Current, even state-of-the-art, analyses in fields that contain ultrascale data are mostly limited to the exploration of a select, small subset of variables in combination or even to univariate analyses in many cases. This is not something that is exclusive to those that do not have access to high-performance computing (HPC). In fact, it is well represented by HPC users for the simple reason that the curse of dimensionality is particularly harsh on most attempts to consider the complex inter-actions in the systems under study. Climate data is represented by ultrascale temporal and spatial datasets such that even the latest large-scale analyses are restricted to several variables known to be of interest a priori. This is clearly inadequate for providing the understanding that is required for better prediction in climate extremes, and other approaches are needed (Ganguly, 2009). In the case of cyber security, modern methods not only ignore most complex variables and temporal aspects, but those that attempt to analyze data from many different subsystems do so in a piecemeal approach that approaches fusion as a method for joining decisions that ignore the inter-relations among the variables involved at lower levels (Corona et al., 2009).

There are many software packages that attempt to scale traditional analysis techniques (e.g Matlab, R, Colt, etc. all have parallel versions). But parallelization of traditional methods that do not attempt to or cannot jointly consider the interactions in such systems is not the kind of scalability that is required for the type of scientific discovery that is most desired from ultrascale datasets. Many large-scale data packages use visualization as a key component to isolate portions of the data for further exploration, and while visualization approaches are useful, they become more difficult and less meaningful with larger and larger datasets. Often, they end up being a guided approach to ignoring large portions of the data simply because all of the inter-actions cannot be jointly considered, despite the understanding that they might actually be relevant.

In addition to not being set up to allow straightforward application by the scientific community, modern, scalable algorithms are not scalable to extreme levels. For example, the scalable approach in Aliferas et al. (2010) is polynomial, but on the order of the size of the conditioning set. Even small conditioning set sizes would be intractable at ultrascale, unless the algorithm is wrapped in a parallel learning framework, such as the one we propose to implement here. And such is the flexibility of the approach we propose that this is possible; i.e. it is fairly flexible with respect to the specific approach used for the underlying predictive analysis, so that it becomes possible to wrap such algorithms in our basic approach. On a related note, the same paper (Aliferas et al., 2010) gives a relevant exploration of the relation between feature selection and causal discovery through a review of much of the recent, relevant literature from statistics and machine learning. The review supports the importance of approaches such as those represented here to scientific discovery and the crucial role of predictive targets in the analytic process.

For the most part, data analysis techniques have traditionally been designed around the study of several variables in relation to a single target. Many modern scientific datasets (particularly those that require ultrascale data processing) are characterized by the presence of huge numbers of inter-related target functions of interest. Several subsets of variables can contribute to understanding emerging local phenomenon that is suggestive of discovery goals. Analyzing targets in isolation is a suspect approach at best under these conditions. Current software packages are not designed to take advantage of the information contained in the interactivity of large systems. The analysis method in this chapter is designed to systematically hypothesize interaction sub-structures among observed variables, observe correlations among the sub-structures, rank and identify contributing variables, and thereby capture the essence of the model underlying complex systems generating very large datasets. We seek to step away from the kind of approaches to large-scale data analysis that have so far been attempted in the HPC community. The approach is strongly motivated by the very datasets to which we envision applying it, and it offers a clear path to extreme scalability.

5.2 General approach: using interaction information

We take advantage of the fact that a large percentage of data exploration relates to prediction. In other words, it is vital to understand which subsets of variables in which combinations influence or generate certain phenomena. This type of analysis is incredibly common, and can be defined in a very generic manner.

In the following, a very minimal example is given in order to give a high-level view of the base analytical approach. At the most basic level, the approach can support a user in the analysis of some phenomenon of interest. The initial analysis treats this phenomenon as a core prediction problem. The ground truth that consists of the values of the variable specified as the target problem is used to build additional, related prediction problems using other variables. More specifically, a shallow correlation analysis is used to find other variables in the dataset that seem to correlate (at least loosely, and the level can be user defined) with our target. We use these correlated variables to construct more prediction problems with each of them as the corresponding target. Note that we can safely assume that at least some of these other phenomena are related (perhaps in a very complex way) to the target. If that is not a valid assumption, then there may be no need to study the dataset all together as a large-scale data problem. We consider that variables (atomic or complex) either correlate due to random factors or they correlate for a reason. We want to make the distinction between the two possibilities. The variables and patterns and their perceived predictive values are analyzed across a large set of prediction tasks that seem to correlate in the same system. The output of this analysis is used to inform the user of interesting variables and patterns thereof that relate to their target phenomenon (having ruled out or re-ranked many based on perceived value across the entire system).

Consider the following example, which is an oversimplification of the analysis that describes the intuitive principles of the approach. Assume there are two sets of variables, the first correlates strongly with only your target apart from random correlations here and there with other relevant phenomena, and the second set of variables correlates strongly with your target and strongly or weakly with a number of other somewhat related phenomena in your system (more so than random correlation would likely account for). What is the probability that the first is noise relative to the probability that the second is? As long as your system is somewhat inter-connected, bringing in additional information to evaluate variables can add substantial value. If the joint analysis is extremely large across many factors in a dataset, a reasonable ratio of examples to potentially predictive observations can be maintained even when dramatically expanding the set of variables under consideration. Figures 5.1 and 5.2 show this contrast pictorially when predicting storm severity based on climate variables.

Most of the loosely correlated data need not be relevant to the target problem; as long as some of the coordinated problems are relevant, coordinated learning allows noise to be more reasonably distinguished from predictive variables. There is very strong evidence in Ando and Zhang (2005) that given a large enough amount of information, this general approach to incorporating extra information, even in moderately inter-connected problem spaces, can find pattern sets that generalize (are good predictors on data that was not in the initial analysis) better than feature sets

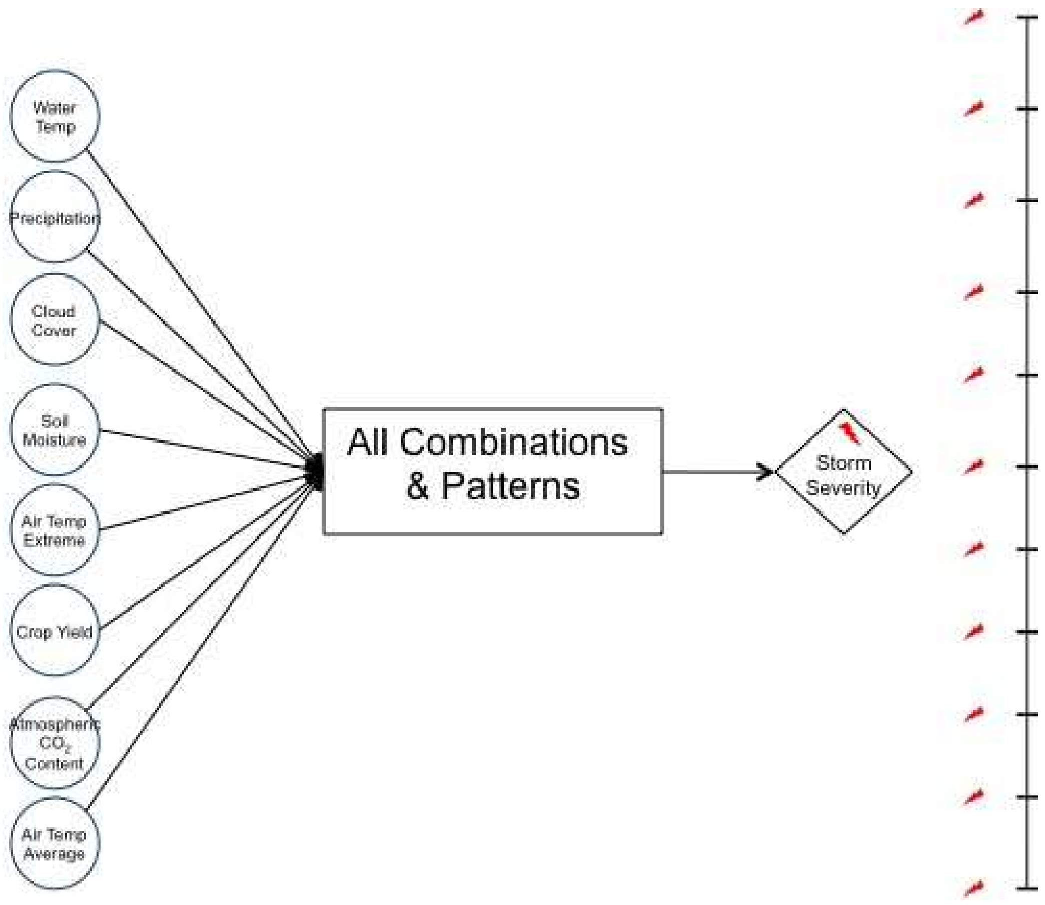


Figure 5.1: Learning the predictive value of variables against a single outcome variable.

hand-engineered by domain experts over decades of study. Another important factor pertaining to real-world data is the fact that data imputation is not necessary to incorporate variables with missing values. For example, as many predictive problems as necessary can be created by sub-setting the data into groups that have all values available in that particular auxiliary problem, and thus it is possible to learn from any relevant data that is available, and combine the knowledge from each of the sub-problems at the joint analysis phase.

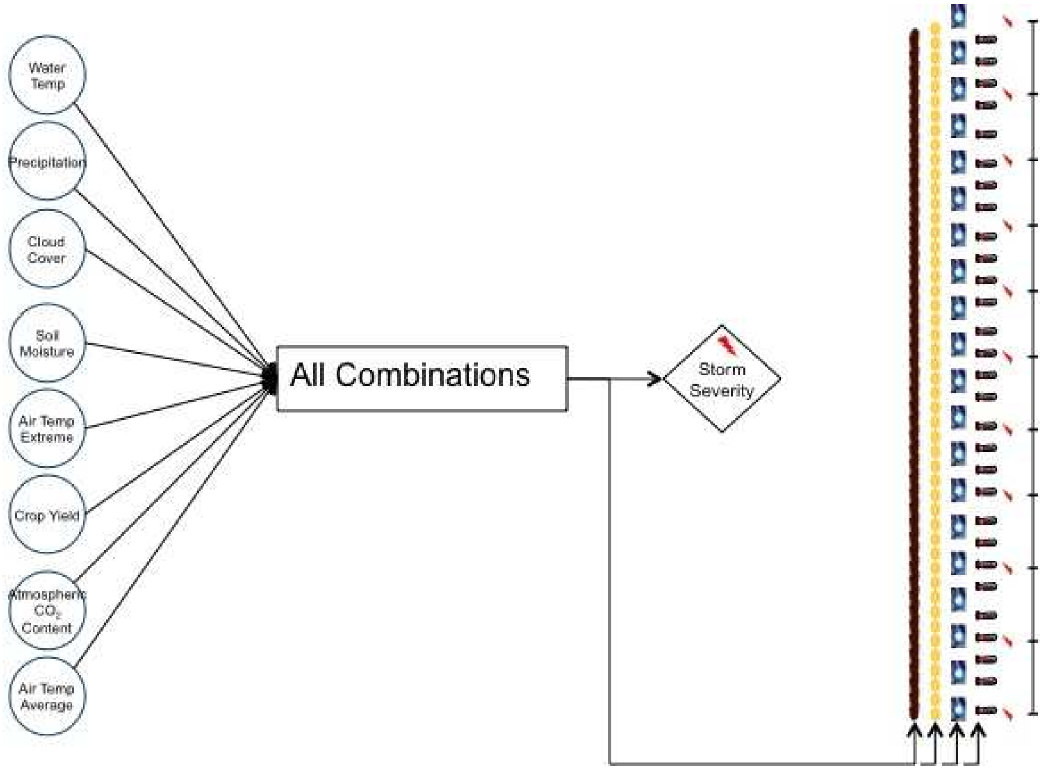


Figure 5.2: Learning the predictive value of variables across many outcomes in an interconnected system.

5.2.1 Task selection for multi-task learning

The framework we developed takes a naturally parallelizable approach to multi-task learning (Ando and Zhang, 2005) and modifies it for complex systems analysis, where automated auxiliary task creation is naturally motivated. In both theory and practice it has been shown that joint learning across multiple *related* tasks can provide benefit to generalization performance. The definition of *relatedness*, however, is important. The scenarios we hope to address are in the context of complex systems, and therefore, relatedness of two variables (the tasks of predicting two different random variables) can be thought of as being affected by or affecting many common variables. In Ben-David and Borbely (2008), it was shown that when multiple tasks are $\mathcal{F}$-related, then multi-task learning can improve performance on each task. More specifically, the authors show that the sample complexity of each task will have a smaller upper bound when learning jointly from related tasks. So, what is this notion of relatedness, and does it apply to the joint prediction of multiple variables in complex systems?

The following definition is taken verbatim from Ben-David and Borbely (2008):

Definition 8. For a measure space $(\mathcal{X}, \mathcal{A})$, where $\mathcal{X}$ denotes a domain set, and $\mathcal{A}$ is a σ-algebra of its subsets, we discuss probability distributions, P over $\mathcal{X} \times \{0, 1\}$, for which the P-measurable sets are the σ-algebra generated by the sets of the form $A \times B$, for $A \in \mathcal{A}$ and $B \subseteq \{0, 1\}$.

- For a function, $f : \mathcal{X} \rightarrow \mathcal{X}$, let $f[\mathcal{A}]$ be $\{A \subseteq \mathcal{X} : f^{-1}(A) \in \mathcal{A}$, and let $f[P]$ be the probability distribution over $\mathcal{X} \times \{0, 1\}$ defined by having the probability distribution $f[P]$ assign to a set $T \subseteq \mathcal{X} \times \{0, 1\}$, the probability $f[P](T) = P(\{(f(x), b)|(x, b) \in T)\})$.

Let $\mathcal{F}$ be a set of transformations $f : \mathcal{X} \rightarrow \mathcal{X}$, and let P_1, P_2 be probability distributions over $\mathcal{X} \times \{0, 1\}$.

- We say that P_1, P_2 are $\mathcal{F}$-related if there exists some $f \in \mathcal{F}$ such that $P_1 = f[P_2]$ or $P_2 = f[P_1]$.
- We say that two samples are $\mathcal{F}$-related if they are samples from $\mathcal{F}$-related distributions.

By this definition, one must know or define a set a transformations that constitute $\mathcal{F}$. Therefore, there is a lot of flexibility in the definition to capture various relationships between tasks. However, it should be expected that if the family of functions is allowed to grow too large, then complexity of joint learning increases since one must be able to account for all possible transformations. Leaving this increase in complexity aside for a moment, let's consider how inter-relatedness in complex systems factors into this definition of *relatedness*. If we limit our family of functions to linear transformations, then any prediction task that has a strong linear correlation with our target task is technically $\mathcal{F}$-related, according to the above definition. The trick in Ben-David and Borbely (2008) is that the transformations among tasks must be known in order to create a working algorithm. Thus, as long as we stick with simple correlations, we should be able to ensure benefit from the joint learning.

Also note that if what we care about is the discovery of the shared structure among all tasks, then we may only need to show that the average error among all tasks is reduced. With this goal in mind, a proof in the context of structural learning is given in Ando and Zhang (2005). While structure discovery is the main goal of our work, we would ideally be able to improve the generalization error bound for each task, since we often have a primary task, and additional tasks that we learn from are not always of independent interest.

Note that using labeled data to find potential joint learning tasks is more problematic when the labeled set size is small, as in most semi-supervised learning problems. While this is not always a problem in many complex systems, such as climate data analysis, where the number of observed target-variable outcomes is often large, it can still cause problems due to the high dimensionality.

As one might surmise based on the above discussion of relatedness, the most difficult step to achieving success in this framework is the task selection step. A similar issue is expressed in Blitzer et al. (2006), where the authors use the basic ASO approach for transfer learning, which they refer to as Structural Correspondence Learning (SCL). In this case the choice of joint tasks involves the selection of so-called *pivot* features. In the case of transfer learning the goal is to find common structures that are predictive of similar items in the two domains. For example, Blitzer et al. (2006) are attempting to build models for part-of-speech tagging. Thus, the tasks are selected such that they can learn structures that indicate nouns, etc.

The choice of pivot features in SCL is very similar to our task-selection goal, and many of the difficulties pointed out in Blitzer et al. (2006) can be seen in structural learning for complex systems. For example, features need to occur often enough that predictive capability of co-occurring structures can be estimated accurately. In addition, if there isn't variety in the choice of their pivot features, they may not learn structures that can make important distinctions that they care about. In SCL, the extra information comes from the unlabeled information in the new domain, as well as from the unlabeled information in the original domain. In complex systems, the extra information comes from several sources. First, we have the ability to look at data where important values are not recorded rather than relying on data imputation. Second, the approach allows us to combine knowledge from data where our target variable is not recorded, but where related variables are. And finally, we can benefit from noise reduction even across data where our target variable is consistently recorded when the variable measurements are noisy.

We investigated the use of the following variable correlation measures to aid in automatic task creation, where new tasks are created to predict the highly ranked variables:

1) The combination of Fisher criterion score and Mutual Information described in Dhir and Lee (2009).
2) Kraskov mutual information (Kraskov et al., 2004).
3) Spearman rank correlation.
4) Feature weighting based on construction of a linear discriminative learning method.
5) Maximal Information Coefficient (Reshef et al., 2011).

In general, the best results were achieved with the Spearman rank correlation scores. For example, while the Maximal Information Coefficient (see Section 5.3.3) revealed more powerful relationships, it was nontrivial to create new problems that captured the proper transformation between variables.

5.3 Scalable multivariate analysis framework

Once tasks have been chosen for joint learning, we can step through the large-scale learning approach in detail.

Our framework relies on the Alternating Structure Optimization (ASO) algorithm of Ando and Zhang (2005). The approach was designed to support semi-supervised learning through a multi-task learning framework, in which so-called *auxiliary tasks* that are present in unlabeled data can assist in the discovery of predictive structures relevant to the target problem. In the original work, the notion was for an expert to define relevant tasks for which the labels were always available. In addition, the goal was to choose auxiliary tasks that would be relevant to the target problem.

The construction of learning tasks can be varied, and in Ando and Zhang (2005) there are a two main themes in the construction. The first is to follow an expert-derived task construction approach, and the second is to follow the inspiration of co-training Blum and Mitchell (1998) to propagate labels. However, unlike in co-training, error propagation is not very problematic, since the goal is to learn predictive structures from the propagated labels, as opposed to learning a final model from them.

Algorithm 4 Alternating Structure Optimization (Ando and Zhang, 2005)

```
Input: $\{(X_i^l, Y_i^l)\}$, $(l = 1, \ldots, m)$
Parameters: $h$ and $\lambda_1, \ldots, \lambda_m$
Output: $h \times p$ matrix $\theta$
Initialize $u_l = 0$, $(l = 1, \ldots, m)$ and $\theta$
iterate
for $l = 1$ to $m$ do
    Fix $\theta$ and $v_l = \theta u_l$, and solve for $w_l$:
    $w_l = \text{argmin}_{w_l} \left[ \frac{1}{n_l} \sum_{i=1}^{n_l} L(w_l^T X_i^l + (v_l^T \theta) X_i^l Y_i^l) + \lambda_l \|w_l\|_2^2 \right]$
    Let $u_l = w_l + \theta^T v_l$
end for
Compute the SVD of $U = \left[\sqrt{\lambda_1} u_1, \ldots, \sqrt{\lambda_m} u_m\right]$
$U = V_1 D V_2^T$
$\theta :=$ first $h$ rows of $V_1^T$
until convergence
```

When following Algorithm 4, the bulk of the computational time (after task construction) is spent on the learning of the individual tasks. Although we initially thought that the SVD computation would take a long time, this turned out to not be the case in relative terms. However, for scales at which the SVD does become problematic, scalable solutions are available, see below. Fortunately, the task ranking and construction is trivially parallelizable, as is the work comprising the set of individual model training events. Thus, the framework scales nearly linearly, and should continue do so to much larger scales than we attempted. It can be observed that the algorithm should perform better as both the unlabeled data, u, and the number of tasks, m, grow. In other words, the more related tasks that are available, and the more labels available for each of those related tasks, the more likely that the structure discovery process will be able to avoid making spurious connections among potentially predictive variables based on noise.

5.3.1 Internal learners

As proposed, the method only requires that the internal learners be parametric, so that parameter weights can be compared to find common structure. As in Ando and Zhang (2005), our implementation relies on Stochastic Gradient Descent (SGD) (Zhang, 2004) to quickly learn weight-based models. In the case of regression, $Y \in \mathcal{R}$, we use the Huber loss, and in the binary case, $Y \in \{-1, +1\}$, we use the modified Huber loss. The Huber loss is less sensitive to outliers than the standard square loss for regression problems.

- Huber loss:

$$\phi(h(x), y) = \begin{cases} (h(x) - y)^2, & \text{if } |h(x) - y| \leq 1 \\ 2|h(x) - y| - 1, & \text{otherwise} \end{cases} \tag{5.1}$$

- Modified Huber loss:

$$\phi(h(x), y) = \begin{cases} max(0, 1 - h(x)y)^2, & \text{if } h(x)y \geq -1 \\ -4h(x)y, & \text{otherwise} \end{cases} \tag{5.2}$$

5.3.2 Feature set expansion

One of the goals of using the system's inter-relatedness is to increase the ground truth such that the roles of a larger number of potentially predictive variables can be more reliably estimated. Therefore, it makes sense to find efficient ways to select higher-order variables for exploration. While there is a large body of work on deep-learning (Bengio, 2009; Arel et al., 2010) that does this in an unsupervised manner, we focus once again on relevance to the target problem as a means for constraining our search for useful variables.

In order to take full advantage of the additional information on which features' predictive power can be assessed, we have the capability to construct higher-order features that we believe should be considered. While we occasionally do this in a brute force approach, we also use measures from Maximal Information-based Nonparametric Exploration (MINE) statistics Reshef et al. (2011) to explore feature construction based on combining Maximal Information Content (MIC) with other MINE statistics that arise from the characteristic matrix. For example, the Maximum Edge Value (MEV) measures how close a relationship is to being a function, such that when a high MIC score showing dependence between two variables is combined with a low MEV score indicating the relationship is not function-like, the value of the information may be hard to capture without creating a higher-order feature from the variables (see below).

5.3.3 MINE statistics

Maximal Information-based Nonparametric Exploration (MINE) statistics (Reshef et al., 2011) are a set of techniques that allow the discovery of a large variety of relationships between variable pairs. These statistics are generated from a matrix of mutual information scores generated by binning the data points into a variety of grids of various sizes. The primary MINE statistic is known as the Maximal Information Coefficient (MIC) score. This score is equivalent to the largest mutual information score that arises from any of the grids into which the samples can be placed. Unlike a mutual information score, the maximal information coefficient is comparable across datasets, since the score is normalized to fall between 0 and 1, inclusive.

In addition to the MIC score, there are three other MINE statistics available: the Maximum Edge Value (MEV); the Maximum Asymmetry Score (MAS), and the Minimum Cell Number (MCN). The Maximum Edge Value measures how function-like a relationship is. In other words, if the relationship is not a functional mapping, then this score will be low. The Maximum Asymmetry Score measures the deviation from monotonicity of the relationship. The value will be low if the relationship is highly monotonic. Finally, the Minimum Cell Number measures the complexity of the relationship by looking at the number of cells required to reach the MIC score. The MEV and MAS scores will fall between 0 and 1, inclusive, just like the MIC score. However, the MCN value is not normalized. Note that MINE statistics are calculated between pairs of variables. Therefore, once again applying the algorithm to assess multiple pairs is trivially parallelizable.

5.3.4 Parallel Processing

All code for this project was written in Java, which supports threading across multi-core CPUs by using the native threads provided by the underlying operating system. To ensure that any computationally intensive section uses the available cores, we use the Java ExecutorService class as follows:

```
private static final int NTHREADS = Runtime.getRuntime().
    availableProcessors();
private ExecutorService exec = Executors.newFixedThreadPool(NTHREADS);
```

Then, sections that should use different threads can be parallelized using the *execute* method. For example, the following code would send individual learning tasks to their own threads to populate the matrix U in Algorithm 4.

```
exec.execute(new Runnable(){
        public void run() {
                ASO_SGD stochasticGD = new ASO_SGD(currentProblem);
                Hyperplane hPlane = stochasticGD.optimizeAsInZhang05();
                Iterator<Feature> fIter = hPlane.featureIterator();
                while(fIter.hasNext()){
                        Feature f = fIter.next();
                        uMatrix.set(iIndex, featureFactory.getID(f),
    hPlane.featureScore(f)*Math.sqrt(lambda));
                }
        }
});
```

In addition, to Java threads, we use the Mahout machine learning library, an Apache Software Foundation project that overlays machine learning algorithms on top of Apache's Hadoop distributed computation library. For, example, the following code allows Singular Value Decomposition of the above-mentioned U matrix to be computed.

```
org.apache.mahout.math.SingularValueDecomposition svd = new org.apache.
    mahout.math.SingularValueDecomposition(uMatrix);
```

However, we use the Parallel Colt library for routine calculation of the SVD, since it constituted only a negligible amount of the computation time in our experiments.

```
DenseDoubleSingularValueDecomposition svd = new
    DenseDoubleSingularValueDecomposition(uMatrix, true, true);
```

5.3.5 Scalability Results

It turns out that nearly all of the computation time for our complex systems analysis is taken by two stages. The first consists of the selection of tasks, and the second comprises the training of the models for the chosen tasks. Fortunately, both of the heavy computation stages are trivially parallelizable. In our tests, we achieve superlinear speedup of these code sections on multicore machines. See Figure 5.1 for timing results.

Table 5.1: Parallel ASO: Training time speedup of heavy computation phases for the Multi-Task Learning Framework. 2 × 2.93 GHz 6-core Intel Xeon processors with hyper-threading and 16GB 1333 MHz DDR3 RAM.

Phase	# Tasks	Serial Run Time	Parallel Run Time	Speedup
Task Construction	100	263.9min	11.3min	23.4×
Task Training	100	19.3min	1.5min	12.9×

Chapter 6

Major application areas

The overarching theme of this work is the goal of integrating information to address the curse of dimensionality in large-scale learning. All of the research was driven by actual problems in which large amounts of potentially informative data is available in forms that do not allow simple integration through standard learning or statistical analysis approaches. This section describes a few of the major applications addressed in this work, and describes many results not presented to this point in the thesis.

6.1 Network intrusion detection

Cyber security is one of the major areas where large scale learning approaches appear to hold great promise. In this section we focus on the area of cyber security known as network intrusion detection (Lippmann et al., 2000; Mell et al., 2003; Sommer and Paxson, 2010) and expand on work published in Symons and Beaver (2012). In this domain, labeled data specific to the network of interest must be obtained for any new deployment. In the past, the potential costs of obtaining large amounts of such data have stymied efforts to apply machine learning algorithms to network intrusion detection. However, we are able to show that by using large amounts of unlabeled data, certain algorithms are able to produce dramatic results using very little labeled data.

Learning systems are becoming increasingly pervasive in network intrusion detection research and practice. Although there are many examples of the potential promise of these methods, a variety of factors combine to make both research advances and practical deployment of machine learning systems difficult in the intrusion detection domain. Many of these factors are related to the lack of available labeled data on operational networks. Attacks captured in the wild are rarely made available, and even then, they cannot be directly leveraged to learn models that will be applied to different networks. The critical nature of the *i.i.d.* assumption (that all points used in training and to which the model will be applied are pulled independently and identically from the same distribution) underlying almost all machine learning methods, is often neglected due to the belief that training a model *in situ* (in this context we mean where it will be deployed) is too costly or impractical. Prior results on synthetic datasets have utilized hundreds of thousands or millions of training examples. Even previous semi-supervised learning experiments in this domain have utilized thousands of labeled examples. Obtaining examples in a new network is typically considered too costly to support models that require such large numbers of labeled events, particularly when they aren't guaranteed to dramatically outperform alternate methods.

Since guarantees on the generalization performance of machine learning approaches are based on theoretical error bounds, they do not apply if the assumptions of the method do not match the reality of its utilization. In other words, when deploying a learning system in an environment with its own idiosyncrasies, and keying off of network statistics and other variables that are intertwined with the noise peculiar to the network, the standard assumptions upon which the learning algorithm depends need to be recognized. In nearly all cases, the theoretical performance guarantees depend on the *i.i.d.* assumption. In practice, this means that effective learning systems would ideally always be trained *in situ*, or using examples from the network environment in which they are deployed. Therefore, discriminative learning based on known attack data is potentially limited by the cost inherent in identifying and/or generating real attacks in a new environment. In addition, there is some cost involved in ensuring that normal traffic in an existing network is indeed innocuous. There are often model-specific assumptions that must be recognized as well, and many model types are not well suited to the intrusion detection domain.

Given recent improvements in semi-supervised learning, a practical question that arises is the following: Can we learn an effective intrusion-detection model using a small number of labeled examples? If so, the costs associated with training in situ become less prohibitive. Having a penetration testing team perform attacks on a network for a few hours as opposed to weeks or months, or manually verifying dozens of network transactions as opposed to tens of thousands, becomes a much more manageable requirement to place on an organization for proper instrumentation of a system.

In this section, we show that when the learning algorithm has several implicit assumptions that match the data generation process in the network intrusion detection domain, generalization performance based on very few labels can dramatically outperform existing defense methods. We justify the use of the LapRLS model (Belkin et al., 2006) both theoretically and experimentally and use the algorithm to provide strong evidence on data derived from large-scale operational network data (Song et al., 2011) that we can indeed build very effective models using a small number of labeled examples. In addition to comparing it to several other supervised and semi-supervised models, we compare our results to published results using sophisticated anomaly detection methods and to the output of a signature-based intrusion detection system (IDS) applied to the same data. An ability to generalize very effectively based on few observations is confirmed, demonstrating clear potential to augment current IDS tools with very realistic training requirements in a way that can potentially provide strong alerting coverage against unknown attacks with trivial false positive rates.

The experimental analysis uses data from Kyoto University that was recently made available to the public (see Song et al. (2011)). While this is a carefully curated, valuable new resource, it has limitations, and therefore, we only claim to provide strong evidence that real tools with these characteristics are currently viable. Our experiments are carefully designed to demonstrate true generalization performance on *unknown* attacks using minimal training sets, and the results on the operational data show that we can catch nearly all previously unseen attacks with a false positive rate that is an order of magnitude lower than any of the alternatives (including the signature IDS, which cannot identify previously unseen attacks).

6.1.1 Machine Learning in Intrusion Detection

Most operational network intrusion detections systems rely on very specific rules, or *signatures*, to identify potentially malicious traffic. Human experts generate the signatures after they have extensively analyzed an attack and determined the attack's indicative bit patterns and conditions. While signatures are effective at identifying a specific instance of an attack, developing them is a time-consuming and manually intensive process, during which the network remains vulnerable. Furthermore, simple variants of the attack on which the signature is based will often not trigger the signature pattern. As the frequency and diversity of attack attempts rise, organizations are finding it increasingly difficult to keep pace in developing the raw number of signatures required. A different process for attack analysis is necessary if computer network defense systems are to remain effective.

The intrusion detection research community has responded to the problem of signature development latency by exploring machine-learning methods capable of learning the discriminating characteristics of malicious traffic from exemplar network transaction data. The collective works cover a broad range of techniques and are applied in various architectures in order to propose an optimum approach to network traffic classification. See Dua and Du (2011); Tsai et al. (2009) for reviews of the field. Despite this significant body of work, machine-learning approaches, and in particular supervised learning systems, are sporadically deployed compared with the less sophisticated signature-based approaches. We attribute this phenomenon to both a low confidence in the reported performance of machine-learning-based intrusion detectors, and a poor understanding of how to operationally field them.

Contrast with Anomaly Detection

Outside of the machine learning community, the phrase *anomaly detection* Chandola et al. (2009) is often used interchangeably with and regularly confused with machine learning. For people who understand all of the subfields that lie within these domains, this is not an issue, but a large portion of the community involved in cyber security is unaware of differences. One early contributing factor to the intermixing of these terms is due to a tendency to classify unsupervised learning, or clustering, as a form of machine learning. Another factor is the use of machine learning techniques to solve anomaly detection problems, e.g. where clusters are taken as ground truth for learning classification models. However, we submit that there is a fundamental distinction between the two areas of study based on theoretical foundations of machine learning that have generalization performance as a critical concept. Thus, while anomaly detection can be any mechanism that looks for unusual patterns, machine learning looks to generalize an *expert-defined* distinction. Therefore, on a fundamental level, it is perfectly natural in machine learning to build a model purposely designed to distinguish between malicious and benign network traffic and have optimal performance on previously unseen events. On the other hand, such a problem definition has very little connection with a general definition of anomaly detection, since attacks aren't necessarily anomalous and normal behaviors often are.

In light of the above, it is important to point out that this paper is written in a general context that differentiates learning from anomaly detection, in particular, based on the use of classification labels and the emphasis on generalization performance in machine learning. Thus, real ground-truth labels are a necessary component to the model building process that we hope to address. The downside to using labels is the cost of obtaining them, while the upside is the ability to steer a model in a desired direction. In addition, when we talk about using unlabeled data to augment label information via semi-supervised learning, we do so based on a notion of compatibility (see section 3.3) that uses concepts like regularization to achieve better generalization performance on a classification task defined by the labeled data.

In Sommer and Paxson (2010), the authors provide a good summary of the challenges inherent in the application of machine learning to intrusion detection, but the problem being addressed is still defined to be "outlier detection." In other words, one of the points being argued is that since the problem being solved is anomaly detection, machine learning techniques, which operate well on notions of similarity, are challenged. Our view is that normal traffic can often be completely different from anything previously seen on the network. We also contend that previously unseen attacks may not necessarily appear to be anomalous in the originally defined feature space, yet have distinguishing characteristics such that they are more similar to known attacks than normal traffic. If these assumptions are reasonably accurate, then outlier detection is not the problem we want to solve. Instead, we operate on the assumption that an expert-derived feature space can capture information that allows previously unseen attacks, whether anomalous or not, to be identified as sharing certain distinguishing characteristics with known attacks. The generalization performance we observe when detecting previously unknown attacks on operational data (see section 6.1.2) offers strong evidence that new attacks do indeed resemble known attacks in ways that allow them to be distinguished from normal traffic, even on data where anomaly-detection and signature-based systems struggle to reliably discriminate.

The problem becomes one of finding the right view through which the desired distinction can be seen. Therefore, our approach is to solve a classification problem where experts have provided a small number of ground truth labels on the target network. Our goal is to show the power of using a model whose assumptions very closely match the data generation process in this domain (see section 3.4.3). We use the availability of the labeled operational data in the Kyoto2006+ dataset Song et al. (2011) to help demonstrate that the label requirements for a machine learner can be made small enough, using current methods, to realistically deploy effective learning-based intrusion detectors.

Data Limitations

The lack of confidence that exists in academic evaluations of machine-learning network intrusion detectors can be traced to a shortage of publicly available data. Organizations typically keep their network intrusion data hidden to prevent publicizing any vulnerability. As a result, the majority of academic studies present results that explore a singular approach tailored to a specific environment, and are difficult to verify or validate more generally. A significant gap in the literature that applies machine-learning techniques to the network intrusion detection problem is the absence of a relevant network intrusion data set that can be used as a basis for comparison. While the 1999 KDD cup ”classification task” data Lippmann et al. (2000) provided an initial surge of interest in machine-learning-based intrusion detection, the background traffic was simulated and the data no longer accurately represents modern network traffic. The lack of other relevant, public labeled data sets has severely limited the exploration of machine learning methods in network intrusion detection. The release of the Kyoto2006+ dataset Song et al. (2011), which captures metric sets associated with real operational network flows, is therefore a very promising step toward more accessible research in this area. We summarize some of the most relevant characteristics of this dataset in section 6.1.2.

Semi-Supervised Intrusion Detection

Semi-supervised learning has begun to be explored in intrusion detection. In Chen et al. (2008), the authors explore the use of transductive spectral methods and Gaussian random fields on the 1999 KDD Cup dataset Lippmann et al. (2000). The transductive approach achieves the best results in that study, but unfortunately the use of transductive methods is not practical in real-time systems. Although it is probably safe to assume that an out-of-sample extension would not suffer a major drop in performance, the results only support incremental improvement over supervised methods, and the authors lament that they do not reach a level of performance that would be valuable in practice. For example, although the test setup is different, the anomaly detection techniques used in Abe et al. (2006) appear to achieve significantly better results on the same dataset.

In Lane (2006), semi-supervised concepts are explored, but in a very untraditional manner, involving partially observable markov decision processes (POMDPs), an area of reinforcement learning that suffers from scalability issues Roy et al. (2005). The method is intended to be a proposed framework in which to place intrusion detection. As such, it has merit, but it does not make significant strides toward practical usage. Mao et al. Mao et al. (2009) take an interesting approach based on multi-view, semi-supervised learning and active learning, which requires an interactive process with the user, and apply it to the KDD Cup data. They show improvement over their baseline, which is single view learning without active learning, but the amount of labeled data used is still very high and the number of false positive alerts remains in an unusable range.

Laplacian RLS for Intrusion Detection

As mentioned above, the main semi-supervised model that we focus on in the cyber domain is the Laplacian RLS. The application of this model in this domain is interesting from multiple perspectives. Recall from 3 that it is possible to arrive at the same functional form for this model based both the Laplacian Regularized Least Squares (Laplacian RLS) model in Belkin et al. (2006) and the Bayesian Kernel Model in Liang et al. (2007a,b); Pillai et al. (2007) with a Dirichlet process prior. This is relevant to our discussion because we hope to use models whose assumptions more realistically match the realities of the data. And we can argue that both of these derivations are in tune with how we hope to shape (or avoid shaping) the model.

In the case of the Laplacian RLS Belkin et al. (2006), we are using unlabeled data as a graph-based regularization term, which essentially means that we can use as much unlabeled data as we want to penalize models that would assign points among the unlabeled data that are extremely close together in our expert inspired feature space as belonging to different classes. This makes sense as long as the features are relatively important to the problem domain. We believe this to be true a priori due to the fact that they were derived by experts.

Now, reconsider the case of the Bayesian Kernel Model Liang et al. (2007a,b); Pillai et al. (2007), with a Dirichlet process prior. In network intrusion detection, each event may be generated based on its own random process, and we don't want to restrict the form of each of these processes. In addition, we don't want to restrict the possible number of processes that could be generating the events we observe. This is a typical case in which a Dirichlet process prior might be used. It also makes sense because while we hope that the target function we are trying to learn lies in a dense region that cuts through the original feature space, it also allows us to represent each event as being generated by its own random process.

6.1.2 Experimental Results

We use the Kyoto2006+ dataset Song et al. (2011) for all of the experiments in this section. The dataset covers nearly three years of network traffic through the end of 2008 over a collection of both honeypots and regular servers that are operationally deployed at Kyoto University. The data is provided in the form of observations and statistical features that characterize terminated connections. We only use the first 14 features since any system would have access to the information required to construct these features, whereas the additional features are unlikely to be available. The fact that the features are pre-calculated allows for more accurate comparison of different model types, but it unfortunately restricts the possible features to those provided.

Before using the data, we convert categorical features to binary, and normalize all numeric data using a Softmax scaling approach (with $r = 1$), which is purported to retain the most information Pyle (1999).

Unfortunately, the dataset does not provide information on specific attack types. Therefore, we are unable to take advantage of a cost-sensitive learning scheme, and we are unable to determine how well we are doing with regard to differentiating attack types or prioritizing alerts. Moreover, there is a good deal of suspected labeling error. Even though the number of errors is likely tiny compared to the size of the dataset, this is an important point (see Song et al. (2011) for a detailed description of the dataset).

The dataset essentially represents a two-class classification problem, where the classes represent malicious traffic and non-malicious traffic in a network. There is a distinction made between *known* and *unknown* attack types, which we leverage in some of our experiments to test the ability to generalize knowledge to previously unseen attacks. *Unknown* attacks are defined as those that were not flagged by the signature IDS, but for which the Ashula tool detected shellcodes. The only packet information available to our models is the number of bytes sent by the source and destination.

Comparative Analysis

All tests in this section are performed across the test data used in Kishimoto et al. (2011), which comprises 12 days of traffic pulled from the last six months of 2008. Table 6.1 shows the initial results of training two supervised learners from the Minorthird library Cohen (2004), a linear Support Vector Machine (SVM) and a maximum entropy learner, using a full day's labeled data from January 1, 2008. For comparison, we also display the alerting results from the intrusion detection system (IDS) that are included in the dataset, and we list the results from Kishimoto et al. (2011), which employed an anomaly detection approach using multiple classifiers trained over 10 million training examples.

Table 6.1: Reported IDS results, multi-classifier anomaly detection results, and results of using all (111,589) labeled examples from Jan. 1, 2008 for supervised learning. Testing is performed across the same test data as in Kishimoto et al. (2011), which comprises 12 days of traffic pulled from the last six months of 2008. *The signature IDS alerts are recorded in the dataset. Anomaly detection results are from Kishimoto et al. (2011).

Classifier	Recall	False Positive Rate	AUC Score
Signature IDS*	0.09004	0.01619	N/A
Anomaly Detection	0.8093	0.0590	N/A
Maximum Entropy	0.77292	0.02059	0.72044
Linear SVM	0.98952	0.03528	0.96295

Next, we compare the semi-supervised learners to the supervised learners using very small labeled datasets. The semi-supervised learners are the Laplacian Eigenmap (LEM) and the Laplacian Regularized Least Squares (RLS) algorithms described above. The results are shown in Table 6.2. Subsets of 100 labeled examples and approximately 3000 unlabeled examples from Jan. 1, 2008 are used for training, and testing is performed across the same test data as above. There were 111,589 examples (terminated connections) on January 1, 2008. The classification results are averaged over 10 random selection of the labeled data. We first randomly select 100 examples as our labeled training set and retain the rest as unlabeled examples for use by the semi-supervised learners. However, we also remove redundancy through an approximate similarity measure by hashing the examples based on label value, binary feature values, and 10% ranges of the normalized numeric feature values. This leaves an average of 56.6 labeled examples per experiment, with a high of 69 and a low of 19. It also preserves approximately 3000 unlabeled examples per experiment. We report the average recall, false positive rate, and area under the ROC curve, which is a plot of the tradeoff between false positive rate and recall as the decision threshold of the binary classifier is varied (i.e. the AUC score, see Flach et al. (2011) for an interesting discussion of this measure). Keep in mind that we purposely restricted

Table 6.2: Classifier comparison using small training sets of fewer than 100 labeled examples and approximately 3000 unlabeled examples from Jan. 1, 2008. Testing is performed across the same test data as in Song et al. (2011), which comprises 12 days of traffic pulled from the last six months of 2008. Results are averaged over 10 random selections of labeled examples.

Classifier	Recall	False Positive Rate	AUC Score
Maximum Entropy	0.77292	0.02059	0.72044
Linear SVM	0.96354	0.03029	0.94802
Laplacian Eigenmap	0.64112	0.08715	0.75926
Laplacian RLS	0.89144	0.02667	**0.98651**

the number of labeled examples to an extreme in order to demonstrate the viability of training such models in their deployment environments.

Training on Known to Catch Unknown

Of particular interest is the ability to catch previously unobserved and unknown attacks after training on a small or reasonable number of known attack types. Because the Kyoto2006+ dataset Song et al. (2011) differentiates between known and unknown attacks, we can test this ability directly. In Table 6.3, we examine the ability of the Laplacian RLS learner to catch unknown attacks after being trained on normal traffic and known attacks only. The setup is the same as before using data from Jan. 1, 2008, such that the results are averaged over 10 random selections of the labeled data. Each set has 100 labeled data points total to begin with, thus after eliminating redundancy, we observe a combined total of under 70 labeled examples (combined number of normal and known-attack terminated connections) for each classifier, with as few as 19 labeled examples. Once again, there are approximately 3000 unlabeled examples per experiment. We also count how often the IDS results recorded in the Kyoto2006+ dataset alerted on the data with normal and unknown attacks only. There are a total of 398 unknown attacks that occur during the 12 days in the test set.

Table 6.3: Alerting on unknown attacks. The Laplacian RLS classifiers were trained on subsets of fewer than 70 labeled data comprising only known attacks and known normals. *The signature IDS alerts are recorded in the dataset Song et al. (2011).

Classifier	Recall	| False Positive Rate|	AUC Score
Signature IDS*	0.00000	0.01619	N/A
Laplacian RLS	0.99975	0.02538	0.99987

Table 6.4: Performance of the individual classifiers (randomly selected training sets). All classifiers require less tradeoff between precision and recall than classifier 1. Therefore, they can all conceivably be tuned to achieve the same results: 178 or fewer false positives, while alerting on 397 out of 398 unknown attacks. *The signature IDS alerts are recorded in the dataset Song et al. (2011).

Classifier	|Training Data|	Recall	|False Neg|	|False Pos|	AUC
Signature IDS*	N/A	0.00000	398	13,074	N/A
Laplacian RLS 1	**19**	**0.99749**	**1**	**178**	**0.99968**
Laplacian RLS 2	57	1.0	0	14,753	**0.99993**
Laplacian RLS 3	58	1.0	0	28,498	**0.99992**
Laplacian RLS 4	60	1.0	0	28,498	**0.99970**
Laplacian RLS 5	64	1.0	0	25,621	**0.99993**
Laplacian RLS 6	65	1.0	0	17,456	**0.99993**
Laplacian RLS 7	59	1.0	0	18,278	**0.99986**
Laplacian RLS 8	69	1.0	0	28,498	**0.99986**
Laplacian RLS 9	57	1.0	0	28,498	**0.99995**
Laplacian RLS 10	58	1.0	0	14,707	**0.99995**

Table 6.5: Alerting on unknown attacks. The Laplacian RLS classifiers were trained on subsets of fewer than 70 labeled data comprising only known attacks and known normals, and they were built using automatic threshold-finding functions intended to reduce false positive alerts.. *The signature IDS alerts are recorded in the dataset Song et al. (2011).

Classifier	Recall	| False Positive Rate|	AUC Score
Signature IDS*	0.00000	0.01619	N/A
Laplacian RLS	0.99749	0.00166	0.99987

If we look more closely at the individual results, the real promise of the Laplacian RLS, and potentially other semi-supervised methods whose assumptions match the domain, shines through. In Table 6.4 we provide the results of each of the 10 runs in order to demonstrate how low the number of false positives can be bounded. The first run has the lowest AUC score of 0.99968, but has the lowest false positive rate of 0.00022 (out of 808,108 normal events). It is also the only classifier to have a recall of less than 100%, but it still catches 99.75% of the unknown attacks. The binary Laplacian RLS model uses a threshold, so the AUC score indicates how much tradeoff needs to occur between precision and recall. Therefore, since the model that catches 397 unknown attacks, while missing only one, only has 178 false positive alerts and yet has the lowest AUC score, all of the other models should be tunable to allow them to miss a single attack while keeping their false positive number at 178 or lower, as well, since they require less of a tradeoff than the first model.

Given the AUC scores in Table 6.4, it makes sense to add an automatic threshold selection routine to the training step in order to obtain better performance. Table 6.5 and Table 6.6 show the results of the Laplacian RLS classifiers when the thresholds are tweaked during training (on training data) to eliminate false positives. In this case, we used a method whereby we rank all labeled training data by the score assigned by the model, and then we attempt to find a threshold that will guarantee a maximum false positive rate of 0.00000001 on the training data with the hope that this will transfer to the test data. We find the distance between this discovered threshold and

Table 6.6: Performance of the individual classifiers (randomly selected training sets) when using an automatic threshold-finding function during training. This function is intended to raise the threshold to avoid false positive alerts, but it only uses training-data to find the threshold. *The signature IDS alerts are recorded in the dataset Song et al. (2011).

Classifier	|Training Data|	Recall	|False Neg|	|False Pos|	AUC
Signature IDS*	N/A	0.00000	398	13,074	N/A
Laplacian RLS 1	19	**0.99749**	**1**	**164**	**0.99968**
Laplacian RLS 2	57	**0.99749**	**1**	**173**	**0.99993**
Laplacian RLS 3	58	**0.99749**	**1**	**676**	**0.99992**
Laplacian RLS 4	60	**0.99749**	**1**	**9807**	**0.99970**
Laplacian RLS 5	64	**0.99749**	**1**	**166**	**0.99993**
Laplacian RLS 6	65	**0.99749**	**1**	**167**	**0.99993**
Laplacian RLS 7	59	**0.99749**	**1**	**1779**	**0.99986**
Laplacian RLS 8	69	**0.99749**	**1**	**166**	**0.99986**
Laplacian RLS 9	57	**0.99749**	**1**	**203**	**0.99995**
Laplacian RLS 10	58	**0.99749**	**1**	**151**	**0.99995**

the maximum score of 1, multiple it by 0.75, and add it to the old threshold to obtain a new one. Unfortunately, our choice of 0.75 is rather arbitrary, so despite the fact that the threshold is set on the training data, it is likely that such a method would need to be tweaked manually in practice based on the number of false positives that a user could tolerate. However, it is clear that these models are very powerful methods of finding unknown attacks, and it is equally clear that if the intention is to find previously unseen attacks, then these methods hold great promise for the defense of large networks. As mentioned above, the optimal threshold for each of these learners should guarantee fewer than 178 false positives for any of the the classifiers. Thus, the improvements shown in Table 6.6 can be improved upon as well. Therefore, future work will include better methods of automatic threshold generation, which is a particular challenge when the size of the training data is limited to realistic numbers as in this paper.

6.1.3 Scaling graph-based learners

While graph based learning methods are particularly useful for encoding alternate information into the learning process, they are not inherently scalable. In this section, we explore a couple of ways in which these methods can be adapted to large-scale problems. Straightforward parallel processing is possible, and we provide some results using such an approach, but we also demonstrate that a linear version of learning can be quite powerful when the graph is only used for regularization.

6.1.4 Linear Laplacian Regularized Least Squares

The Linear Laplacian Regularized Least Squares (Linear LapRLS) algorithm (Sindhwani et al., 2005b) is a very scalable approach for problems with sparse feature spaces, meaning that most values are zero, but it is also a viable replacement for the standard LapRLS when the number of examples is extremely large. The formulation is similar to the LapRLS (Belkin et al., 2006) with the graph Laplacian still serving as a regularization term based on the unlabeled points. However, instead of using the Gram matrix as a regularization term for the labeled points, the algorithm uses the standard L2 regularization of the coefficient vector. In other words, it penalizes models that disagree with the graph, and it penalizes models with large weights.

Where standard manifold regularization attempts to minimize Equation 3.11, linear manifold regularization attempts to minimize the following function (see Sindhwani et al. (2005b)):

$$(w^*, b^*) = \underset{w,b}{\operatorname{argmin}} \gamma_A w^T w + \gamma_I w^T X^T L X w + \frac{1}{l} \sum_{i=1}^{l} V(y_i, w^T x_i + b) \tag{6.1}$$

Here, X is the $n \times d$ data matrix, where the rows are the training examples. Thus, the Linear LapRLS algorithm, using the standard squared loss function $V(y, w^T x + b) = (y - w^T x - b)^2$, solves the following equation to obtain a weight vector, w:

$$(X_l^T X_l + \gamma_A l I + \gamma_I l X^T L X) w = X_l^T Y \tag{6.2}$$

where X is the data matrix with examples as rows, X_l is the labeled data matrix, Y is the label vector, and L is the Laplacian matrix. For the regularization parameters, we set $\gamma_A l = 0.005$ and $\gamma_I I = 0.045$, so that 90% of the regularization is from the Laplacian matrix.

6.1.5 Scaling Results

While the LapRLS demonstrates great potential for network intrusion detection, such nonparametric models are extremely slow, since every new test point must be compared with every labeled and unlabeled sample in the training set. This is particularly problematic when attempting to use extremely large amounts of unlabeled data. Therefore, we experimented with methods for speeding up the algorithms, including implementing the Linear LapRLS (Sindhwani et al., 2005b) described above. Again, for our tests, we use the data from the Kyoto2006+ dataset (Song et al., 2011). The utilized portion of the data consists of example types from three classes; normal connections, known attacks identified by the deployed intrusion detection system, and unknown attacks discovered by running a shellcode detection tool, Ashula. The training sets consist of all connections from January 1, 2008 minus any unknown attacks. There are 111,563 connections. The test set is the same data used in Kishimoto et al. (2011), which consists of over 1.2 million connections observed over the course of 12 days spread over the latter half of 2008.

Our initial scaling attempts focused on parallelization of the graph construction during training and the model application during testing. See Figures 6.7 and 6.8 for timing results.

In Table 6.9, we report the times and accuracy levels of the Linear LapRLS in comparison with the standard LapRLS. We observe that accuracy is maintained with the scalable approach, while reducing test time by nearly three orders of magnitude.

Table 6.7: Parallel graph building: Training time speedup on the graph construction phase (which is the majority of the training time) by computing nearest neighbors on different threads. 2 × 2.93 GHz 6-core Intel Xeon processors with hyper-threading and 16GB 1333 MHz DDR3 RAM.

Method	Graph Size	Graph Construction Time	Speedup
Serial	3037	$34504msec$	n/a
Parallel	3037	$4967msec$	7×

Table 6.8: Parallel testing: Test phase speedup by setting up duplicate models on different threads. 2 × 2.93 GHz 6-core Intel Xeon processors with hyper-threading and 16GB 1333 MHz DDR3 RAM.

Classifier	Test Time	Speedup
LapRLS	$8456sec$	n/a
Parallel LapRLS	$435sec$	19×

Note that a model's runtime is particularly important in this domain, where one must be able to keep up with large bursts of network traffic, including during denial of service (DOS) attacks. In addition, we compare results with and without the unlabeled data to demonstrate how much the unlabeled data affects the resulting model. See Table 6.10 for results.

6.2 Text processing

Language processing involves applications on which multi-task learning has proven extremely valuable. Although the complex systems assumptions do not hold as

Table 6.9: Scaling with Linear LapRLS. Single processor testing for both models.

Classifier	Recall	False Positive %	AUC Score	Test Time	Speedup
LapRLS	0.99749	0.000220	0.99968	$140min$	n/a
Linear LapRLS	0.99749	0.000412	0.99861	$12sec$	700×

Table 6.10: Effect of Unlabeled Data

Classifier	Recall	False Positive %	AUC Score
Linear LapRLS	0.99749	0.000412	0.99861
Linear LapRLS (no unlabeled)	1.00	0.035265	0.80755

strongly, it is still possible to find structure in the data by looking at relations among the variables. In the case of Ando and Zhang (2005), *auxiliary tasks* were used to find useful structures for prediction in Named Entity Recognition (Tjong Kim Sang and De Meulder, 2003). These additional problems primarily involved prediction of some features given others, which allows semi-supervised application to unlabeled data. However, the task construction method was chosen specifically for the data domain, as opposed to automatically, as we do here. Using the unlabeled data as additional information for finding predictive structures, the authors achieved the best known results on the CoNLL 2003 task (Tjong Kim Sang and De Meulder, 2003; Ando and Zhang, 2005), despite using very simple features as a base from which more complex structures can be found. The ability to use extremely large amounts of unlabeled data for text processing problems through our scalable framework should allow us to build very effective models in this domain.

6.2.1 Text categorization

Text categorization is a common high-dimensional classification task that allows us to investigate the feature discovery aspects of our approach in a straightforward manner. Most text categorization tasks involve training a model over a corpus of documents in which a human has sorted the documents into coarse, subject-based categories. Our goal in this domain is to use large amounts of unlabeled data and multiple, automatically selected tasks, to attempt to discover more complex predictive features than we would normally consider if we had to construct such features manually. We test our approach on two binary classification tasks from the well-known 20 newsgroup data.

In all cases, we preprocessed the data to remove headers and stopwords from the documents. Stopwords are terms, like artcles (e.g. *a* and *the*), that are known *a priori* to have no useful properties for the task at hand. Our stopwords were selected from a standard list. Next, the terms were stemmed using the Porter stemming algorithm, which reduces words to their base forms. For example, a stemmer would reduce the word *running* to its base form, *run*. Finally, the stems were used as features with values determined by their TF-IDF scores (term frequency / inverse document frequency).

The first categorization task involved discriminating between the atheism and religion newsgroup documents. This first task used 1424 documents and contained 21,140 features. Note that attempts to learn more complex feature combinations are very problematic since consideration of all possible feature combinations would result in $2^n - 1$ possible features. The second task was based on learning to discriminate between the baseball and hockey newsgroups. This second task used 1992 documents and contained 22,059 features. In both cases, we used 100 labeled documents to choose joint learning tasks based on Spearman rank correlation coefficients. We simply find the top ranked features based on correlation with the labels, and use those as our joint learning tasks for discovery. Each of the experiments used 100 labeled examples, and 100 automatically generated auxiliary learning tasks over the rest of the data.

6.2.2 Complex feature discovery in text

Feature discovery results are shown in Table 6.11. This is a domain that is well understood by humans, and therefore, I would not necessarily expect an automated method to derive better features than one might generate through manual feature construction. However, at the top of the matrix, we already see very insightful features being generated, and I was able to learn something about a domain in which I

Table 6.11: Feature discovery in text categorization. The rows of a table contain the highest scoring features (in absolute value) for a given row in the matrix.

Task	Row	Top Positive Terms	Top Negative Terms
Religion vs. Atheism	1	elohim	
Religion vs. Atheism	2		gammaray, collide, vibrate galaxy, dwarf, photon binary uncharged, larsonian, quasar astrophysicist, accelerate solar, pulsar
Baseball vs. Hockey	2	may june	gm, utica, springfield adirondack, rochester binghamton, cape, moncton breton, providence, cdi

considered my knowledge to be nearly complete. For example, *elohim* would appear to be a very useful term for prediction of the *religion* category, since it is one of the Jewish words for God. I was not aware of the term prior to this analysis. Furthermore, the second row of the *religion vs. atheism* matrix contains a group of words from physics. It would make sense that this group would make a good feature for prediction of the *atheism* category. Even in the *baseball vs. hockey* matrix, I discovered some very useful knowledge that I did not expect. The highest positive terms in the second row, *may* and *june* are indeed relevant to baseball more than hockey, but the highest negative weight terms appeared to be noise at first glance. However, after further investigation, I found that almost all of those terms were names of cities that had AAA or college hockey teams. This is the type of discovery I would expect in a field that is more opaque to human knowledge, so to encounter such interesting and relevant predictive terms in this domain is extremely significant.

6.3 Climate

Climate research involves one of the most complex systems known to man. Interdependencies among variables can be strong based on a variety of factors, including some that span long distances and long periods of time (Steinhaeuser et al., 2011). Traditionally, a prevalent trend in climate science has been to pull variables out of their system context to evaluate their predictive importance due to the large size of climate datasets. However, given the high interactivity of the variables, this does not provide much confidence in the results. Only recently have climate scientists begun to look at the data in a more realistic system context, but there are problems scaling traditional multivariate analytical approaches, and better analysis methods are needed (Ganguly, 2009).

If one considers the true dimensionality of the system, including unmeasured variables, fine-grained climate observations, and complex variables that are non-linear combinations of atomic measurements, then the sample complexity problem becomes obvious despite the incredible amount of observational data. Further complicating factors is the amount of missing data for each variable. Very few variables are actually measured in every single global grid point or time period, so data imputation is common in the climate domain, where it is often referred to as *reification*. Since our methodology can be used to deal with sample complexity, computational complexity, and missing data in the same framework, it has the potential to provide new insights in climate science. In this domain it should be obvious that we care about multiple target predictions at the same time, since it is important to understand the interrelations among temperatures, precipitation, crop yield, soil moisture, ozone content, etc., if one hopes to understand our climate more fully. Since these targets interact and influence one another and are influenced by many of the same variables, a multitask learning approach has great potential.

We tested our approach on an ozone prediction dataset (see Zhang and Fan (2008)). The problem contains 72 variables, but simply by adding all possible combinations of two features as both sums and differences to the atomic variables, we end up with 10,296 features to consider. In this automatic feature expansion scenario, we actually get improved performance when using the top 10 ranked complex features as auxiliary prediction tasks. The best result (among 8 different learning models) from

Zhang and Fan (2008) on the ozone prediction task had an AUC score of 0.495 using cross-validation with ninety percent of the data being used for training. We tested our approach given unlabeled data as well as labeled data, and we used 10% of the labels during cross-validation, yet achieved an AUC score of 0.638.

In addition, we show feature discovery results in Table 6.12. The primary features discovered in the smaller Ozone subset are interesting because they appear to relate to two of the eight variables used in a formula designed by experts in this domain, i.e. *wind speed near sunrise* and *wind speed mid-day* (see Zhang and Fan (2008)). On the other hand, the first row of the matrix in the larger set seems to have found a very relevant, but more complex feature. HT70 is the geopotential height at the 700 hPa level, and HT85 is the geopotential at the 850 hPa level. Therefore, the difference between the two is a measure of the geopotential thickness in that layer of the atmosphere. This is interesting because there is evidence in other studies that there is a relationship between geopotential thickness and ozone anomalies (Jiang et al., 2008). The MINE statistics for the two variables are also listed in Table 6.13, along with a few other relationships for comparison. The relationship does appear to be fairly strong, but there are many other variables in the dataset with even stronger relationships. The difference is that most of the other relationships don't appear to be predictive with respect to ozone anomalies. Thus, as one might expect, it appears that the ability to center our discovery around prediction is vital. In addition, the MINE statistics for some of the wind variable pairs are given in Table 6.14. These statistics suggest that finding the wind variable combination from Table 6.12 would be difficult without joint consideration of the target task.

Table 6.12: Feature discovery in ozone prediction. The rows of a table contain the highest scoring features (in absolute value) for a given row in the matrix.

Task	# Labels	# Features	Row	Top Pos	Top Neg	AUC
Ozone Predition	253	72	1		WSR5 WSR13 WSR12 WSR9 WSR6	.638
Ozone Prediction	1000	10296	1	HT85	HT70	0.806

Table 6.13: MINE statistics for select pairs of variables from the Ozone Prediction dataset. For comparison purposes, several simple X, Y relationships are also included. HT: geopotential height; T: temperature; Precp: precipitation; PK: peak.

Relationship	MIC	MEV	MAS	MCN
HT70; HT85	0.637	0.637	0.024	2.6
HT50; Precp	0.900	0.900	0.359	3.6
T6; T8	0.929	0.929	0.046	3.6
Precp; T_PK	0.878	0.878	0.342	3.3
Circle (x, y - plot)	0.677	0.333	0.022	3.2
$y = sin(18\pi x)$	1.0	1.0	0.915	5.2
$y = 2x + 3$	1.0	1.0	0.0	2.0

Table 6.14: MINE statistics for select wind variable pairs from the Ozone Prediction dataset.

Relationship	MIC	MEV	MAS	MCN
WSR5; WSR6	0.864	0.864	0.063	3.8
WSR12; WSR13	0.823	0.823	0.061	3.0
WSR12; WSR5	0.354	0.354	0.025	2.6
WSR12; WSR6	0.324	0.324	0.033	2.6
WSR13; WSR5	0.359	0.359	0.043	2.6
WSR13; WSR6	0.305	0.305	0.035	2.0

Chapter 7

Conclusions

7.1 Summary

In this work, we have demonstrated multiple new methods for dealing with the *curse of dimensionality* in many real world applications by integrating disparate sources of information in more natural and scalable ways. In many important domains, we are addressing some of these issues for the first time, and our hope is that this work will lead to increased interest and progress in designing better solutions based on structure-based learning methods.

Several major contributions of this work center around the expanded use of graph-based semi-supervised learning. In particular, we expand on the ability to encode and leverage complex knowledge by modifying the graph construction process. Robust application in noisy domains becomes much more feasible by using carefully chosen methods for encoding target-specific information into the graph. Included in this work are results published in Symons et al. (2012), including very significant results on an application to a Brain-Computer Interface problem that has been particularly challenging for semi-supervised learning methods (Chapelle et al., 2006). Our method demonstrated the best performance over a large set of previous semi-supervised results (Chapelle et al., 2006) using a simplistic implementation of our approach. To the best of our knowledge, we also provide the first complete theoretical bound on sample complexity in graph-based semi-supervised learning by combining existing semi-supervised and supervised bounds that can only apply when the graph-construction method uses label information.

These initial successes in graph modification suggested a new approach to encoding information from multiple views into the graph. To the best of our knowledge, this was the first real attempt to address a multi-view, budgeted learning problem motivated by multiple real applications that we have been exposed to in our work. The results demonstrated a consistent and significant ability to encode useful knowledge from training-only views into a model that operated without access to them. Much of this work was published in Symons and Arel (2011).

The graph modification approaches allowed us to leverage structure from multiple views and unlabeled data to address the curse of dimensionality based on the concentration of measure phenomenon. In addition, we showed that scalable approaches using the graphs are viable, both in terms of speed and accuracy.

Subsequent work addressed the ability to leverage the interconnections in complex systems to jointly address both the computational side and the sample complexity side of the curse of dimensionality in complex systems. We described and demonstrated a new, scalable framework for complex feature discovery via multi-task learning in complex systems, including several compelling feature discovery examples in application domains. We also demonstrated the ability to handle extremely high-dimensional data while actually improving performance, despite label information being limited.

In addition, we covered and expanded on work in Symons and Beaver (2012), and demonstrated the use of learning methods for network intrusion detection that are extremely effective, scalable, and practically trainable. To the best of our knowledge, the results are the first on real operational data that demonstrate an ability to catch most zero-day attacks, with negligible false positive rates that are orders of magnitude lower than all alternatives. Perhaps most significant is that the models were obtained with trivial amounts of labeled data (by leveraging large amounts of unlabeled data). This allows such models to be easily trainable in practice.

7.2 Future Directions

7.2.1 Missing data

We outlined how the approaches discussed in this work allow better use of data with missing variables. In the case of graph modifications, auxiliary information can be used to modify appropriate sections of the graph as availability permits, while sections of the graph with missing information are simply left unchanged. In the multi-task framework it is possible to divide data as finely as possible to create learning problems without missing measurements. In this manner, available information can be incorporated via the task matrix without the need for imputation of missing values. Future work should involve more sophisticated data handling methods that can take advantage of this ability seamlessly.

7.2.2 Supporting complex task transformations

As described in 5.2.1, leveraging joint learning tasks that have extremely complex relationships is not a problem for our framework theoretically. However, from a practical standpoint sophisticated methods for mapping the transformations between tasks are required to allow more information to be leveraged effectively.

7.2.3 Additional applications

While we covered many application domains throughout the course of this work, we believe that the learning frameworks developed here can play an important role in many areas that we have not addressed to this point.

Bibliography

Bibliography

Abe, N., Zadrozny, B., and Langford, J. (2006). Outlier detection by active learning. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 13, 82

Aliferas, C. F., Statnikov, A., Tsamardinos, I., Mani, S., and Koutsoukos, X. D. (2010). Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation. *Journal of Machine Learning Research*, 11:171–234. 61

Ando, R. K. and Zhang, T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853. 12, 15, 59, 63, 65, 67, 68, 69, 70, 93

Arel, I., Rose, D. C., and Karnowski, T. P. (2010). Research frontier: deep machine learning–a new frontier in artificial intelligence research. *Comp. Intell. Mag.*, 5(4):13–18. 71

Asuncion, A. and Newman, D. (2007). Uci machine learning repository. 54

Bakker, B. and Heskes, T. (2003). Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, 4:83–99. 15, 16

Balcan, M.-F. (2008). *New Theoretical Frameworks for Machine Learning*. Phd thesis. 28, 29, 30, 32

Balcan, M.-F. and Blum, A. (2006). *An Augmented PAC Model for Semi-Supervised Learning*. MIT Press, Cambridge, MA. 21, 28

Balcan, M.-F., Hanneke, S., and Wortman, J. (2008). The true sample complexity of active learning. In *the 21st Annual Conference on Learning Theory (COLT)*. 12, 13

Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396. 9, 18, 32, 41

Belkin, M. and Niyogi, P. (2004). Semi-supervised learning on riemannian manifolds. *Machine Learning*, 56:209–239. 24, 33, 37, 39, 51, 52

Belkin, M., Niyogi, P., and Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434. 34, 35, 36, 37, 51, 52, 53, 77, 82, 83, 90

Bellman, R. E. (1961). *Adaptive Control Processes: A Guided Tour*. Princeton University Press. 3

Ben-David, S. and Borbely, R. S. (2008). A notion of task relatedness yielding provable multiple-task learning guarantees. *Machine Learning*, 73:273–287. 15, 65, 66

Ben-David, S. and Schuller, R. (2003). Exploiting task relatedness for multiple task learning. In *Proceedings of the 16th Annual Conference on Learning Theory (COLT)*. 15

Bengio, Y. (2009). Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127. 71

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer. 8

Blitzer, J., McDonald, R., and Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *EMNLP '06*, pages 120–128. 67

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *the Conference on Computational Learning Theory (COLT)*. 14, 46, 69

Blum, A. L. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271. 4

Brinker, K. (2003). Incorporating diversity in active learning with support vector machines. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, pages 59–66. 13, 14

Bryll, R., Gutierrez-Osuna, R., and Quek, F. (2003). Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. *Pattern Recognition*, 36:1291–1302. 23

Caruana, R. (1997). Multitask learning. *Machine Learning*, 28:41–75. 15

Cesa-Bianchi, N., Shalev-Shwartz, S., and Shamir, O. (2009). Some impossibility results on budgeted learning. In *27th International Conference on Machine Learning (ICML), Budgeted Learning Workshop*. 46, 47

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection - a survey. *ACM Computing Surveys*, 41(3). 79

Chapelle, O., Scholkopf, B., and Zien, A. (2006). *Semi-Supervised Learning*. MIT Press, Cambridge, MA. 9, 37, 38, 99

Chen, C., Gong, Y., and Tian, Y. (2008). Semi-supervised learning methods for network intrusion detection. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 2603 –2608. 81

Chu, C.-T., Kim, S. K., Lin, Y.-A., Yu, Y. Y., Bradski, G., Ng, A. Y., and Olukotun, K. (2006). Map-reduce for machine learning on multicore. In *Advances in Neural Information Processing (NIPS)*. 60

Chung, F. R. K. (1997). *Spectral Graph Theory.* American Mathematical Society, Providence, RI. 9

Cohen, W. W. (2004). Minorthird: Methods for identifying names and ontological relations in text using heuristics for inducing regularities from data. 84

Cohn, D., Atlas, L., and Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, 15(2):201–221. 12, 13, 14

Corona, I., Giacinto, G., Mazzariello, C., Roli, F., and Sansone, C. (2009). Information fusion for computer security: State of the art and open issues. *Information Fusion*, 10(4):274–284. 44, 60

Dasgupta, S. (2005). Coarse sample complexity bounds for active learning. In *Advances in Neural Information Processing (NIPS).* 12, 13

Dasgupta, S., Kalai, A. T., and Monteleoni, C. (2005). Analysis of perceptron-based active learning. In *Eighteenth Annual Conference on Learning Theory.* 12, 14

Dhir, C. S. and Lee, S. Y. (2009). Hybrid feature selection: Combining fisher criterion and mutual information for efficient feature selection. In *International Conference on Neural Information Processing (ICONIP).* 38, 68

Donoho, D. L. (2000). High-dimensional data analysis: The curses and blessings of dimensionality. aide-memoire of a lecture at ams conference on math challenges of the 21st century. 2

Druck, G., Settles, B., and McCallum, A. (2009). Active learning by labeling features. In *Empirical Methods in Natural Language Processing (EMNLP).* 12, 14

Dua, S. and Du, X. (2011). *Data Mining and Machine Learning in Cyber Security.* CRC Press. 78

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification.* Wiley-Interscience. 8

Flach, P., Hernandez-Orallo, J., and Ferri, C. (2011). A coherent interpretation of auc as a measure of aggregated classification performance. In *the 28th International Conference on Machine Learning*. 85

Foster, D. P., Johnson, R., Kakade, S. M., and Zhang, T. (2009). Multi-view dimensionality reduction via canonical correlation analysis. Technical Report TTI-TR-2008-4, Toyota Technological Institute, Chicago. 15

Freund, Y., Seung, H. S., shamir, E., and Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning*, 28:133–168. 12, 13, 14

Ganguly, A. R. (2009). Climate extremes: A challenge for math at the petascale. Technical report, ORNL/TM-2009/134. 60, 96

Goldberg, A., Zhu, X., Singh, A., Xu, Z., and Nowak, R. (2009). Multi-manifold semi-supervised learning. In *12th International Conference on Artificial Intelligence and Statistics (AISTATS)*. 18

Goldberg, A. B., Zhu, X., and Wright, S. (2007). Dissimilarity in graph-based semi-supervised classification. In *the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS)*. 11, 21

Goldberg, Y., Zakai, A., Kushnir, D., and Ritov, Y. (2008). Manifold learning: The price of normalization. *Journal of Machine Learning Research*, 9:1909–1939. 18

Greiner, R., Grove, A. J., and Roth, D. (2002). Learning cost-sensitive active classifiers. *Artificial Intelligence*, 139(2):137–174. 46, 47

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182. 4

Irle, A. and Kauschke, J. (2011). On kleinberg's stochastic discrimination procedure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(7):1482–1486. 6, 24, 28

Jiang, X., Pawson, S., Camp, C. D., Nielsen, J. E., Shia, R.-L., Liao, T., Limpasuvan, V., and Yung, Y. L. (2008). Interannual variability and trends of extratropical ozone. part i: Northern hemisphere. *Journal of Atmospheric Sciences*, 65. 97

Johnson, R. and Zhang, T. (2008). Graph-based semi-supervised learning and spectral kernel design. *IEEE Transactions on Information Theory*, 54(1):275–288. 9

Kishimoto, K., Yamaki, H., and Takakura, H. (2011). Improving performance of anomaly-based ids by combining multiple classifiers. In *Applications and the Internet (SAINT), 2011 IEEE/IPSJ 11th International Symposium on*, pages 366–371. 12, 84, 85, 91

Klein, D. and Manning, C. (2002). Conditional structure versus conditional estimation in nlp models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 5

Kleinberg, E. M. (1990). Stochastic discrimination. *Annals of Mathematics and Artificial Intelligence*, 1:207–239. 6, 24

Kleinberg, E. M. (2000a). A mathematically rigorous foundation for supervised learning. *Lecture Notes in Computer Science*, 1857. 27, 28, 30, 31, 32

Kleinberg, E. M. (2000b). On the algorithmic implementation of stochastic discrimination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):473–490. 24, 25, 26, 27, 28

Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324. 4

Kraskov, A., Stgbauer, H., and Grassberger, P. (2004). Estimating mutual information. *Physical Review E*, 69(6):066138. PRE. 68

Landgrebe, D. (2002). Hyperspectral image data analysis. *IEEE Signal Processing Magazine*, 19(1053-5888):17–28. 40

Lane, T. (2006). *A Decision-Theoretic, Semi-Supervised Model for Intrusion Detection*. Springer London. 82

Liang, F., Mao, K., Liao, M., Mukherjee, S., and West, M. (2007a). Nonparametric bayesian kernel models. Technical report, Duke University. 34, 35, 82, 83

Liang, F., Mukherjee, S., and West, M. (2007b). The use of unlabeled data in predictive modeling. *Statistical Science*, 22(2):189–205. 34, 35, 82, 83

Lin, T. and Zha, H. (2008). Riemannian manifold learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):796–809. 9, 18

Lippmann, R. P., Fried, D. J., Graf, I., Haines, J. W., Kendall, K. R., McClung, D., Weber, D., Webster, S. E., Wyschogrod, D., Cunningham, R. K., and Zissman, M. A. (2000). Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation. In *the 2000 DARPA Information Survivability Conference and Exposition*. 75, 81

Liu, H. and Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):491–502. 4

Mann, G. and McCallum, A. (2008). Generalized expectation criteria for semi-supervised learning of conditional random fields. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*. 14

Mao, C.-H., Lee, H.-M., Parikh, D., Chen, T., and Huang, S.-Y. (2009). Semi-supervised co-training and active learning based approach for multi-view intrusion detection. In *Proceedings of the 2009 ACM symposium on Applied Computing*, SAC '09, pages 2042–2048, New York, NY, USA. ACM. 82

Marwala, T. (2009). *Computational Intelligence for Missing Data Imputation, Estimation, and Management: Knowledge Optimization Techniques*. Information Science Reference - Imprint of: IGI Publishing, Hershey, PA. 5, 46

McCallum, A. and Nigam, K. (1998). Employing em and pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 350–358. Morgan Kaufmann Publishers Inc. 13

Mell, P., Hu, V., Lippmann, R., Haines, J., and Zissman, M. (2003). An overview of issues in testing intrusion detection systems. Technical Report NIST IR 7007, National Institute of Standards and Technology. 75

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill. 4, 8

Nadler, B., Lafon, S., and Coifman, R. R. (2005). Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In *Advances in Neural Information Processing (NIPS)*. 9, 18

Nakamura, E. F., Loureiro, A. A., and Frery, A. C. (2007). Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Computing Surveys*, 39(3). 44

Orbanz, P. and Teh, Y. W. (2010). *Bayesian Nonparametric Models*. Springer. 46

Ouimet, M. and Bengio, Y. (2005). Greedy spectral embedding. In *the 10th Workshop on Artificial Intelligence and Statistics (AISTATS)*. 33, 52

Pillai, N. S., Wu, Q., Liang, F., Mukherjee, S., and Wolpert, R. L. (2007). Characterizing the function space for bayesian kernel models. *Journal of Machine Learning Research*, 8:1769–1797. 34, 35, 82, 83

Pyle, D. (1999). *Data Preparation for Data Mining, Volume 1*. Morgan Kaufmann. 5, 84

Reshef, D. N., Reshef, Y. A., Finucane, H. K., Grossman, S. R., McVean, G., Turnbaugh, P. J., Lander, E. S., Mitzenmacher, M., and Sabeti, P. C. (2011). Detecting novel associations in large data sets. *Science*, 334:1518–1524. 68, 71

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290. 9, 18

Roy, N., Gordon, G., and Thrun, S. (2005). Finding approximate pomdp solutions through belief compression. *Journal of Artificial Intelligence Research*, 23:1–40. 82

Sassano, M. (2002). An empirical study of active learning with support vector machines for japanese word segmentation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 505–512. 13, 14

Schapire, R. E. and Freund, Y. (2012). *Boosting: Foundations and Algorithms*. MIT Press. 27

Schapire, R. E., Freund, Y., Barlett, P., and Lee, W. S. (1997). Boosting the margin: A new explanation for the effectiveness of voting methods. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 322–330, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. 27

Schohn, G. and Cohn, D. (2000). Less is more: Active learning with support vector machines. In *Proceedings of the 17th International Conference on Machine Learning*, pages 839–846. 13, 14

Settles, B. (2009). Active learning literature survey. Technical Report Technical Report 1648, University of Wisconsin - Madison. 12

Shen, D., Zhang, J., Su, J., Zhou, G., and Tan, C.-L. (2004). Multi-criteria-based active learning for named entity recognition. In *Proceedings of the ACL 2004*. 13, 14

Sindhwani, V., Niyogi, P., and Belkin, M. (2005a). A co-regularization approach to semi-supervised learning with multiple views. In *Workshop on Learning with Multiple Views, 22nd International Conference on Machine Learning*. 15, 46, 50

Sindhwani, V., Niyogi, P., Belkin, M., and Keerthi, S. (2005b). Linear manifold regularization for large scale semi-supervised learning. In *22nd International Conference on Machine Learning, Workshop on Learning with Partially Classified Training Data*. 90, 91

Sindhwani, V. and Rosenberg, D. S. (2008). An rkhs for multi-view learning and manifold co-regularization. In *Proceedings of the 25th International Conference on Machine Learning*. 15, 46, 50

Singh, R., Vatsa, M., and Noore, A. (2009). Multimodal medical image fusion using redundant discrete wavelet transform. In *International Conference on Advances in Pattern Recognition*. 15

Skurichina, M. and Duin, R. P. W. (2001). Bagging and the random subspace method for redundant feature spaces. In Kittler, J. and Roli, F., editors, *Multiple Classifier Systems (MCS) 2001*, volume 2096, pages 1–10. LNCS. 23

Sommer, R. and Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In *Proceedings of IEEE Symposium on Security and Privacy*. 75, 80

Song, J., Takakura, H., Okabe, Y., Eto, M., Inoue, D., and Nakao, K. (2011). Statistical analysis of honeypot data and building of kyoto 2006+ dataset for nids evaluation. In *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns*. 12, 77, 80, 81, 83, 84, 86, 87, 88, 89, 91

Steinhaeuser, K., Ganguly, A. R., and Chawla, N. V. (2011). Multivariate and multiscale dependence in the global climate system revealed through complex networks. *Climate Dynamics*. 96

Symons, C. T. and Arel, I. (2011). Multi-view budgeted learning under label and feature constraints using label-guided graph-based regularization. In *28th*

International Conference on Machine Learning, Workshop on Combining Learning Strategies to Reduce Label Cost. 15, 44, 100

Symons, C. T. and Beaver, J. M. (2012). Nonparametric semi-supervised learning for network intrusion detection: Combining performance improvements with realistic in-situ training. In *Proceedings of the 5th ACM Workshop on Artificial Intelligence and Security (AISEC)*. 75, 100

Symons, C. T., Vatsavai, R. R., Jun, G., and Arel, I. (2012). Bias selection using task-targeted random subspaces for robust application of graph-based semi-supervised learning. In *11th International Conference on Machine Learning and Applications*. 17, 99

Taylor, M. E., Fern, A., Driessens, K., Stone, P., Maclin, R., and Shavlik, J. (2008). Aaai workshop on "transfer learning for complex tasks". 15

Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290. 9, 18, 41

Tjong Kim Sang, E. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language independent named entity recognition. In *CoNLL '03*, pages 142–147. 93

Tong, S. and Koller, D. (2002). Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66. 13

Tsai, C.-F., Hsu, Y.-F., Lin, C.-Y., and Lin, W.-Y. (2009). Intrusion detection by machine learning: A review. *Expert Systems with Applications*, 36(10):11994 – 12000. 78

Van Breukelen, M., Duin, R. P. W., Tax, D. M. J., and Den Hartog, J. (1998). Handwritten digit recognition by combined classifiers. *Kybernetika*, 34(4):381–386. 54

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience. 8

Weinberger, K. Q. and Saul, L. K. (2006). Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70(1):77–90. 9, 18

Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. (2007). Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8:35–63. 15

Zhang, K. and Fan, W. (2008). Forecasting skewed biased stochastic ozone days: analyses, solutions and beyond. *Knowl. Inf. Syst.*, 14(3):299–326. 96, 97

Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *the 21st International Conference on Machine Learning (ICML)*, pages 919–926. 70

Zhu, X. (2003). Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the ICML-2003 Workshop on the Continuum from Labeled to Unlabeled Data*. 13, 14

Vita

Christopher Todd Symons was born in the suburbs of Washington, D.C. in 1971. He graduated from the North Carolina School of Science and Mathematics in 1988, and matriculated at the University of North Carolina at Chapel Hill, where he studied between 1988 and 1990. He graduated from the University of Tennessee, Knoxville with a Bachelor of Science in Mathematics in 1998 and a Master of Science in Foreign Language/ESL Education in 2000. From 2004 through the middle of 2007, he worked as a Research Associate at the Joint Institute for Computational Science and the Oak Ridge Institute for Science and Education. In 2007 and 2008 he was employed in the position of Research Scientist I through the Joint Institute for Computational Science. In August 2008, he joined the Oak Ridge National Laboratory (ORNL) as an Associate R&D Staff Member, where he was subsequently promoted to the position of R&D Staff Member in 2012. His work at ORNL is focused on developing new approaches to large-scale, high-dimensional, statistical machine-learning, which he has applied to many real-world problem domains, including natural language processing, cyber security, biomedical informatics, and climate analysis. He has been awarded three UT-Battelle Significant Event Awards during his time at ORNL. He is the author of 24 publications, including 1 refereed journal paper, 15 refereed conference papers, 1 refereed conference abstract, and 7 technical reports.