Supervision Reduction by Encoding Extra Information about Models, Features and Labels

Yi Zhang

January, 2011

Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Jeff Schneider, Chair
Geoff Gordon
Tom Mitchell
Xiaojin Zhu (University of Wisconsin-Madison)

Abstract

Learning with limited supervision presents a major challenge to machine learning systems in practice. Fortunately, various types of extra information exist in real-world problems, characterizing the properties of the model space, the feature space and the label space, respectively. With the goal of supervision reduction, this thesis studies the representation, discovery and incorporation of extra information in learning.

Extra information about the model space can be encoded as compression operations and used to regularize models in terms of compressibility. This leads to learning compressible models. Examples of model compressibility include local smoothness, compacted energy in frequency domains, and parameter correlation. When multiple related tasks are learned together, such a compact representation can be automatically inferred as a matrix-variate normal distribution with sparse inverse covariances on the parameter matrix, which simultaneously captures both task relations and feature structures.

Extra information about the feature space can usually be conveyed by certain feature reduction. We propose the projection penalty to encode any feature reduction without the risk of discarding useful information: a reduction of the feature space can be viewed as a restriction of the model search to certain model subspace, and instead of directly imposing such a restriction, we can search in the full model space but penalize the projection distance to the model subspace. In multi-view learning, the projection penalty framework provides an opportunity to simultaneously address both overfitting and underfitting.

Extra information about the label space can be extracted and exploited to improve multi-label predictions. To achieve this goal, we present error-correcting output codes (ECOCs) for multi-label classification: label dependency is represented by the most predictable directions in the label space and extracted by canonical correlation analysis (CCA) and its variants; the output code is designed to include these most predictable directions in the label space to correct prediction errors. Decoding of such codes can be efficiently performed by mean-field approximation and significantly improves the accuracy of multi-label predictions.

Effective collection of supervision signals is an indispensable part of supervision reduction. In this thesis, we consider active learning for multiple prediction tasks when their outputs are coupled by constraints. A cross-task value of information criteria is designed, which encodes output constraints to measure not only the uncertain of the prediction for each task but also the inconsistency of predictions across tasks. A specific example of this criteria leads to the cross entropy between the predictive distributions of coupled tasks, which generalizes the notion of entropy used in single-task uncertainty sampling.

Keywords: Learning with Limited Supervision, Regularization, Compression, Matrix-Variate Normal Distributions, Dimension Reduction, Error-Correcting Output Codes, Canonical Correlation Analysis

Contents

1 Introduction

1.1 Motivations

Learning an unknown function from a set of training examples and generalizing well on unseen samples is the central goal of machine learning. In many real-world applications, direct supervision is limited due to the cost of obtaining high-quality labeled examples. This presents a major challenge to modern machine learning systems. Fortunately, training examples is far from the only source of information: various types

of extra information exist, revealing properties of the model space, the feature space, and the label space. In this thesis, we study learning with limited supervision by encoding extra information.

Extra information about models, when available, can be used as an inductive bias for learning. A well-studied example is model sparsity, i.e., the number of nonzero model coefficients is small. In addition to model sparsity, various types of information about models are available in real-world problems, and encoding such information is important for learning with limited supervision. We first study learning compressible models, where domain knowledge about the model is encoded as a compression operation in regularization. For text-related problems, we propose to learn the structure of the model space from seemingly irrelevant unlabeled text. We then consider learning multiple tasks, where a compact representation of multiple models can be automatically inferred as a matrix-normal distribution on the matrix of model coefficients.

Extra information about features can usually be characterized by certain feature reduction, e.g., a subset of selected features, a clustering of low-level features, or a general feature subspace (or manifold). Directly performing a feature reduction, however, may discard useful information and lead to potential loss of predictive power. We propose the projection penalty framework to encode information from a feature reduction *without* the risk of information loss: a reduction of the feature space can be viewed as a restriction of the model search to certain model subspace, and instead of directly imposing such a restriction, we can search in the full model space but penalize the projection distance to the model subspace.

Extra information about labels is valuable for multi-label prediction. Indeed, a fundamental assumption of multi-label learning is the existence of certain dependency among labels. Otherwise, it is sufficient to solve a set of independent single-label learning problems. We consider the key issue of representing, extracting and encoding the label dependency in order to improve multi-label learning. We propose multi-label error-correcting output codes (multi-label ECOCs). Label dependency is represented and extracted as the most predictable directions in the label space using canonical correlation analysis (CCA) and its variants; an output code is then designed to encode these predictable directions to correct prediction errors.

Active learning with extra information focuses on the effective collection of supervision signals in the presence of extra information. We consider an active learning scenario when multiple prediction tasks are coupled in the sense that their outputs need to satisfy certain logical constraints. Such a coupled learning paradigm is common when we build prediction models to classify objects into a taxonomy, e.g., reading the web and assigning extracted facts into an ontology. In this case, the active learning strategy should consider not only the uncertain of the prediction for each task but also the inconsistency of predictions across tasks.

1.2 Organization

The thesis will be organized to address the following questions:

- Part I: how to effectively encode extra information about the *model* space into learning? (Section 3 & 4 & 5)
- Part II: how to effectively encode extra information about the *feature* space into learning? (Section 6 & 7)
- Part III: how to effectively encode extra information about the *label* space into learning? (Section 8 & 9)
- Part IV: how to effectively *collect supervision signals* in the presence of extra information? (Section 10)

Also, Section 2 reviews related work and Part V provides summary and timeline for the proposed research.

2 Survey

2.1 Regularization

Regularization is a principled way to control model complexity in learning and has been the focus of statistics and machine learning for decades (Hastie et al., 2001). Classical examples include ridge regression (Tikhonov & Arsenin, 1977) in statistics and support vector machines (Boser et al., 1992; Cortes & Vapnik, 1995) in machine learning, which correspond to minimizing either squared loss or hinge loss with $\ell 2$ regularization. Meanwhile, $\ell 1$ regularization has become very popular for learning in high-dimensional spaces since the introduction of lasso (Tibshirani, 1996). A fundamental assumption of $\ell 1$ regularization is the sparsity of model parameters. Sparse models automatically select relevant features and have the advantage of being easy to interpret and good generalization ability in high-dimensional problems.

Recently, designing informative regularization has been one of the main approaches for multi-task learning (Argyriou et al., 2006), transfer learning (Raina et al., 2006) and semi-supervised learning (Belkin et al., 2006). The key idea is to encode information from related tasks, source domains and unlabeled data into the penalty. Also, additional structure assumptions on models can be imposed via $\ell 1$ regularization. Fused lasso (Tibshirani et al., 2005) includes an $\ell 1$ penalty on the differences of successive model coefficients and leads to piecewise constant estimations. Group lasso (Yuan et al., 2006) adds further restrictions on the standard sparsity: model coefficients in the same group tend to be set to zero together. Structured sparsity (Huang et al., 2009) generalizes the group lasso to allow other structured assumptions on the sparsity pattern.

The proposed work in Part I and Part II is based on the framework of regularization, where we encode extra information about the model space and the feature space into regularization penalties.

2.2 Compressed Sensing

Compressive sampling (Candes, 2006) or compressed sensing (Donoho, 2006) was recently developed for signal acquisition, and has received considerable attention (Baraniuk et al., 2008). According to this theory, one can successfully acquire a signal, e.g., an image, from many fewer measurements than required by Nyquist-Shannon sampling theory. The key assumption is that signals like natural images are compressible, i.e., nearly sparse in a compression domain. Under this assumption, $\ell 1$ regularized reconstruction algorithms can reconstruct a signal from only a few linear measurements, where the key is to minimize the measurement errors plus a penalty (or constraint) on the $\ell 1$ norm of the reconstructed signal in a predefined compression domain. The use of a compression for image reconstruction motivates our formulation of learning compressible models (Section 3), which encodes information about the model space as compression operations.

2.3 Multi-Task Learning

Multi-task learning has been an active research area for more than a decade (Baxter, 1995; Thrun & O'Sullivan, 1996; Caruana, 1997). For joint learning of multiple tasks, connections need to be established to couple related tasks. One direction is to find the feature structure shared by tasks. Along this direction, researchers propose to infer the feature structure by performing covariance estimation (Argyriou et al., 2006; Argyriou et al., 2007), principal components (Ando & Zhang, 2005; Chen et al., 2009) and independent components (Zhang et al., 2006) on the model parameters, to select a common subset of features (Brown & Vannucci, 1998; Obozinski et al., 2009), as well as to use shared hidden nodes in neural networks (Baxter, 1995; Caruana, 1997). On the other hand, assuming all tasks are equally similar is risky. Researchers

recently began to directly infer the relatedness of tasks. These efforts include using mixtures of Gaussians (Bakker & Heskes, 2003) or Dirichlet processes (Xue et al., 2007) to model task groups, encouraging clustering of tasks via a convex regularization penalty (Jacob et al., 2008), identifying "outlier" tasks by robust t-processes (Yu et al., 2007b), and inferring a task similarity matrix (Bonilla et al., 2008; Yu et al., 2007a; Zhang & Yeung, 2010). In Section 5, we propose a matrix-variate normal penalty with sparse inverse convariances to systematically select and encode both feature structures and task relations.

2.4 Error-Correcting Output Codes

Error-correcting output codes (ECOCs) offer a general framework to decompose a multiclass classification problem into a number of binary classification problems (Dietterich & Bakiri, 1994). Via ECOCs, a multiclass problem can be solved using binary classifiers. More importantly, the binary problems provide a redundant representation of the multiclass problem. As a result, prediction errors can be corrected using such redundancy, as studied in channel coding and error-correcting codes (Cover & Thomas, 1991).

The *encoding* of ECOCs decomposes the multiclass problem into a set of binary problems, and defines the *codeword* as the outcomes of the binary problems. Popular ECOC decomposition strategies include one-versus-all (Dietterich & Bakiri, 1994), one-versus-one (Hastie & Tibshirani, 1997), random partitions (Allwein et al., 2001), and partitions obtained by problem-dependent heuristic search (Crammer & Singer, 2002; Pujol et al., 2006). On the other hand, the *decoding* of ECOCs decides the class of an example given the prediction on its codeword. This is achieved by examining all the q candidate classes (for a q-class problem) and choosing the class that minimizes a distance function (Dietterich & Bakiri, 1994), minimizes a loss function (Allwein et al., 2001), maximizes a probability function (Hastie & Tibshirani, 1997; Passerini et al., 2004) or optimizes certain other criteria (Escalera et al., 2010) w.r.t. the predicted codeword.

In Part III of this thesis, we present error-correcting output codes for *multi-label* classification.

2.5 Canonical Correlation Analysis

Since the introduction of canonical correlation analysis by Hotelling (Hotelling, 1935; Hotelling, 1936), CCA has become a fundamental tool to analyze the relations between two *sets* of variables. CCA extracts projection directions for both sets of variables such that their correlation in the projected space is maximized. A recent overview of CCA with application to learning problems is given in (Hardoon et al., 2004). Several variants of CCA have been recently proposed: sparse CCA (Witten et al., 2009; Hardoon & Shawe-Taylor, 2009) enforces the sparsity of projection vectors and leads to interpretable models; kernel CCA (Fyfe & Lai, 2001; Hardoon et al., 2004) handles nonlinear associations between variables; a nonparametric Bayesian extension of CCA, sparse infinite CCA (Rai & Daume, 2009), shows good predictive power. In Part III of this thesis, we will use CCA as the building block of our multi-label error-correcting output codes.

2.6 Active Learning

Active learning selects unlabeled samples for labeling in order to maximally reduce the generalization error of the classifier using limited labeling efforts. Since the generalization error is difficult to measure directly, many other criteria have been proposed for sample selection in active learning (Settles, 2009), e.g., uncertainty sampling (Lewis & Catlett, 1994), query-by-committee (Seung et al., 1992; Freund et al., 1997), version space reduction (Tong & Koller, 2002), expected error reduction (Roy & McCallum, 2001).

Recently there has been interest in active learning for multiple prediction tasks. Co-testing (Muslea et al., 2006) is a multi-view active learning strategy, in which examples receiving different predictions from multiple views are selected. In (Reichart et al., 2008), multi-task active learning is performed by iteratively selecting samples from each task or aggregating the selection scores from all tasks. In (Qi et al., 2008), it is proposed to estimate the correlation of labels and predict a joint label distribution to guide active learning. In structured prediction, active learning can query either an entire structured instance or subcomponents of an instance (Roth & Small, 2006). In Part IV of this thesis, we study active learning with multiple tasks coupled by output constraints. Co-testing is a special case of this setting: tasks are to predict the same label from different views, and task outputs are coupled by *agreement* constraints (i.e., predictions should agree).

Part I
Learning with Extra Information about Models

In Part I, we focus on encoding extra information about the model space into learning. In Section 3, we propose learning compressible models (Zhang et al., 2010), where domain knowledge about the model space can be encoded as a compression operation in model regularization. In Section 4, we study a case where the correlation structure of the model space can be learned from large amounts of irrelevant unlabeled text (Zhang et al., 2008). In Section 5, we consider learning multiple related tasks: a compact representation of multiple models can be automatically inferred as a matrix-normal distribution with sparse inverse covariances (Zhang & Schneider, 2010a) and used to couple and regularize multiple tasks.

3 Learning Compressible Models (Completed)

We consider *learning compressible models* to encode domain knowledge about the model space as a compression operation and then regularize the learning process in terms of model compressibility:

$$\min_{\mathbf{w}\in R^p, b} L_{\mathbf{D}}(\mathbf{w}, b) + \lambda||\mathbf{P}\mathbf{w}||_1 \tag{1}$$

where $\mathbf{w}$ is the p-dimensional parameter vector, b is the intercept term, and $L_{\mathbf{D}}$ is an empirical loss defined w.r.t. the training set $\mathbf{D}$. A key part of (1) is the compression operation $\mathbf{P}$ that encodes extra information about the model space: the model $\mathbf{w}$ is compressed before being penalized by the $\ell 1$ penalty, and thus $\mathbf{w}$ tends to follow the compression pattern encoded in $\mathbf{P}$ (i.e., sparse in the compressed domain). We restrict our attention to the case where $\mathbf{P}$ is a $p \times p$ matrix, representing a linear and invertible compression operation. In this case, optimization of (1) can be performed efficiently (Zhang et al., 2010).

3.1 Model Compression: Local Smoothness

Many useful functions have compact representations: constant functions, linear functions, piecewise linear functions, quadratic functions, and so on. A key quality of these functions is *smoothness*, which is a property of their *derivatives*: a constant function has zero first-order derivatives, a (piecewise) linear function has zero second-order derivatives (at most locations), a quadratic function has zero third-order derivatives, etc. In this part, we define compression operations related to the local smoothness of model coefficients.

Order-1 *smoothness* assumes that model coefficients do not change very often along a natural order, which has been studied in fused lasso (Tibshirani et al., 2005) and total variation minimization (Rudin et al., 1992). This corresponds to *sparse first-order derivatives* and leads to (piecewise) constant estimation. Order-1 smoothness can be imposed by plugging into (1) a compression $\mathbf{P}$ that calculates the first-order derivatives at successive locations of $\mathbf{w}$ (Zhang et al., 2010). *Order-2 smoothness* assumes *sparse second-order derivatives* and leads to (piecewise) linear estimation. The compression $\mathbf{P}$ for order-2 and other *higher-order smoothness* can be defined recursively based on the order-1 smoothness compression (Zhang et al., 2010). *Hybrid smoothness* happens when model coefficients have several groups, and each group has its own natural order and smoothness property. In this case, $\mathbf{P}$ can be defined as a *block diagonal* matrix.

3.2 Model Compression: Energy Compaction

The energy of many real-world signals is concentrated in a few frequencies, i.e., compacted in the frequency domain. This is a foundation of both image (Wallace, 1992; Christopoulos et al., 2000) and audio (Spanias, 1994) compression. As a result, a model needs to operate only on a few (relevant) frequencies to accurately classify these signals (e.g., images), i.e., a good model also has compacted energy in the frequency domain. In this sense, a frequency domain transform can be used in learning compressible models, e.g., the discrete cosine transform (DCT) used in the JPEG standard (Wallace, 1992). The 2*D DCT* is a linear operation on $m \times n$ images and thus can be rewritten as a linear operation on $p \times 1$ vectors, where $p = mn$ is the dimension of the linear model $\mathbf{w}$ for classifying images. Plugging this operation as the compression $\mathbf{P}$ in (1) will lead to a model estimation that has compacted energy in the frequency domain.

3.3 Experimental Results

In our experiments (Zhang et al., 2010), we study brain-computer interface and handwritten digit recognition, where local smoothness and energy compaction are appropriate model assumptions, respectively. In brain-computer interface, we classify Electroencephalography (EEG) brain signals. An EEG signal contains several EEG channels, and each channel is a time series. In this sense, we assume *channel-wise smoothness*: model coefficients are smooth (along time) within each channel. We use a diagonal block compression matrix, as discussed in Section 3.1, where each block is an order-1 smoothness compression for a channel. The resulting compressible logistic regression reduces the classification error of sparse logistic regression from 30.0% to 20.92%. In digit recognition experiments, we learn to recognize handwritten digits. We assume energy compaction in the frequency domain for the model and use 2D DCT as the model compression. The learned model has sparse coefficients in the frequency domain, gives better recognition rates, and more interestingly, shows meaningful patterns in the original pixel domain about the digits being recognized.

4 Learning the Semantic Word Correlation from Irrelevant Text (Completed)

Certain structure of the model space can be inferred from unlabeled data. For text-related learning problems, the large amount of unlabeled text from the Web is a valuable source of information. However, the Web is an uncontrolled environment and thus unlabeled text in the Web may not be relevant to a specific learning task. This violates the assumption of many semi-supervised learning methods. In this section we show that, for

text-related learning problems, the correlation structure of the model space can be learned from seemingly irrelevant unlabeled text and then used to improve learning of any specific task (Zhang et al., 2008).

4.1 Learning the Semantic Correlation of Words

We first identify the semantic correlation of words[1] as a structure of the model space that can be transferred from unlabeled text. Consider a document classification problem, where we have only one positive example containing two words {gasoline, truck} and one negative example containing two words {vote, election}. Most people will agree that a new document with words {gallon, vehicle} should be classified as positive, although *gallon* and *vehicle* have never been observed in the training set. The key reason is that *gallon* is the unit of *gasoline*, and *truck* is a type of *vehicle*. Since the classifier should have positive weights on *gasoline* and *truck* (as they appear in the only positive example), *gallon* and *vehicle* are likely to receive positive weights, too. Formally, the *semantic correlation of words* corresponds to a correlation structure of the model coefficients and provides a strong inductive bias in the model space. Also, this is an intrinsic structure of the language and thus will not change dramatically even in irrelevant unlabeled text.

We propose to infer the semantic word correlation from seemingly irrelevant unlabeled text and incorporate it into learning of any specific task (Zhang et al., 2008). We first extract a large number of latent topics from unlabeled text, by repeatedly applying bootstrapping and topic modeling. We then infer the word correlation from the word composition of the extracted topics. The resulting correlation structure is used in $\ell 2$-regularization for learning any specific task (as the correlation in the Gaussian prior):

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, \mathbf{w}^T \mathbf{x}_i + b) + \lambda \mathbf{w}^T \boldsymbol{\Sigma}_s^{-1} \mathbf{w} \tag{2}$$

where $\mathbf{w}$ is the vector of model coefficients, b is the intercept term, L is the empirical loss defined on the training examples $\{\mathbf{x}_i, y_i\}_{i=1}^n$, and $\boldsymbol{\Sigma}_s$ is the semantic correlation of all words (in the problem domain).

4.2 Experimental Results

In our empirical study (Zhang et al., 2008), we construct 190 text classification tasks from a real-world benchmark. For each task, the majority of the unlabeled text are from irrelevant tasks, and thus most semi-supervised learning techniques are ineffective. Surprisingly, however, *most of the 190 tasks are significantly improved by encoding the semantic word correlation inferred from irrelevant unlabeled text*.

5 Multi-task Learning with A Sparse Matrix-Normal Penalty (Completed)

In this section, we propose a matrix-variate normal penalty with sparse inverse covariances to encode the model space and couple multiple tasks (Zhang & Schneider, 2010a). Recent methods on discovering common feature structures among tasks (Argyriou et al., 2006) and directly inferring task similarity (Jacob et al., 2008) are variants of the special cases of our formulation. Learning multiple (parametric) models can be viewed as estimating a parameter matrix, whose rows and columns correspond to tasks and features. Matrix-variate normal distributions are powerful tools for characterizing the structure of a matrix. We follow the matrix normal density and design a penalty that decomposes the full covariance of matrix elements into the

[1] We consider the bag-of-word feature space for simplicity. The proposed method can also be applied to n-gram feature space.

Kronecker product of row covariance and column covariance, which characterize task relations and feature representations, respectively. We then perform sparse covariance selection (via $\ell 1$ penalties) on the inverse of task and feature covariances in order to automatically select meaningful task and feature structures.

5.1 Matrix-Variate Normal Distributions

The matrix-variate normal distribution is one of the most widely studied matrix distributions (Dawid, 1981; Gupta & Nagar, 1999). Consider an $m \times p$ matrix $\mathbf{W}$. Since $\mathbf{W}$ has mp elements, the covariance for the elements of $\mathbf{W}$ is of size $mp \times mp$, which is prohibitively large. To utilize the structure of $\mathbf{W}$ as a matrix, matrix normal distributions assume that the full covariance of $\mathbf{W}$ can be decomposed as the Kronecker product $\mathbf{\Sigma} \otimes \mathbf{\Omega}$, where $\mathbf{\Omega}$ is an $m \times m$ covariance matrix of m rows and $\mathbf{\Sigma}$ is an $p \times p$ covariance matrix of p columns. As a result, $\mathbf{W}$ follows a matrix normal distribution with the log-density (Gupta & Nagar, 1999):

$$\log P(\mathbf{W}) \quad \propto \quad \frac{p}{2}\log(|\mathbf{\Omega}|) - \frac{m}{2}\log(|\mathbf{\Sigma}|) - \frac{1}{2}tr\{\mathbf{\Omega}^{-1}(\mathbf{W}-\mathbf{M})\mathbf{\Sigma}^{-1}(\mathbf{W}-\mathbf{M})^T\} \tag{3}$$

where $\mathbf{M}$ is the $m \times p$ expectation matrix, and $|\ |$ and tr are determinant and trace of a square matrix.

Consider a set of n samples $\{\mathbf{W}_i\}_{i=1}^n$ where each $\mathbf{W}_i$ is an $m \times p$ matrix generated by a matrix-variate normal distribution as eq. (3). The maximum likelihood estimation (MLE) of mean $\mathbf{M}$ is (Dutilleul, 1999):

$$\hat{\mathbf{M}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{W}_i \tag{4}$$

The MLE estimators of $\mathbf{\Omega}$ and $\mathbf{\Sigma}$ are solutions to the following fixed-point equations:

$$\begin{cases} \hat{\mathbf{\Omega}} &= \frac{1}{np}\sum_{i=1}^{n}(\mathbf{W}_i-\hat{\mathbf{M}})\hat{\mathbf{\Sigma}}^{-1}(\mathbf{W}_i-\hat{\mathbf{M}})^T \\ \hat{\mathbf{\Sigma}} &= \frac{1}{nm}\sum_{i=1}^{n}(\mathbf{W}_i-\hat{\mathbf{M}})^T\hat{\mathbf{\Omega}}^{-1}(\mathbf{W}_i-\hat{\mathbf{M}}) \end{cases} \tag{5}$$

It is efficient to update $\mathbf{\Omega}$ and $\mathbf{\Sigma}$ as (5) until convergence, i.e., the "flip-flop" algorithm (Dutilleul, 1999).

5.2 Learning with a Matrix Normal Penalty

Consider a multi-task learning problem with m tasks and p features. Models are represented by an $m \times p$ matrix $\mathbf{W}$, where each row corresponds to a task. The matrix normal density (3) provides a structure to couple multiple tasks in $\mathbf{W}$: 1) we set the expectation $\mathbf{M} = \mathbf{0}$ to prefer simple models; 2) the $m \times m$ row covariance $\mathbf{\Omega}$ describes the *task similarity*; 3) the $p \times p$ column covariance matrix $\mathbf{\Sigma}$ represents a *feature structure* shared by tasks. This yields the following total loss $\mathcal{L}$ w.r.t. $\mathbf{W}$, $\mathbf{\Omega}$ and $\mathbf{\Sigma}$:

$$\mathcal{L} = \sum_{t=1}^{m}\sum_{i=1}^{n_t} L(y_i^{(t)}, \mathbf{x}_i^{(t)}, \mathbf{W}(t,:)) + \lambda\left[p\log|\mathbf{\Omega}| + m\log|\mathbf{\Sigma}| + tr\{\mathbf{\Omega}^{-1}\mathbf{W}\mathbf{\Sigma}^{-1}\mathbf{W}^T\}\right] \tag{6}$$

where λ controls the strength of the matrix-normal penalty, $L()$ is a convex empirical loss, $(y_i^{(t)}, \mathbf{x}_i^{(t)})$ is the ith training example of the tth task, and $\mathbf{W}(t,:)$ is the tth row, i.e., the parameter vector of the tth task.

We minimize (6) by alternating optimization. When $\mathbf{\Omega}$ and $\mathbf{\Sigma}$ are fixed, we solve $\mathbf{W}$ by minimizing:

$$\sum_{t=1}^{m}\sum_{i=1}^{n_t} L(y_i^{(t)}, \mathbf{x}_i^{(t)}, \mathbf{W}(t,:)) + \lambda\, tr\{\mathbf{\Omega}^{-1}\mathbf{W}\mathbf{\Sigma}^{-1}\mathbf{W}^T\} \tag{7}$$

which is a convex function w.r.t. $\mathbf{W}$. When $\mathbf{W}$ in (6) is fixed, we can infer $\mathbf{\Omega}$ and $\mathbf{\Sigma}$ by minimizing:

$$p \log |\mathbf{\Omega}| + m \log |\mathbf{\Sigma}| + tr\{\mathbf{\Omega}^{-1}\mathbf{W}\mathbf{\Sigma}^{-1}\mathbf{W}^T\} \tag{8}$$

which is solved as the MLE estimation of $\mathbf{\Omega}$ and $\mathbf{\Sigma}$ in a matrix normal as (5), given one observation $\mathbf{W}$.

Several recent multi-task learning formulations (Argyriou et al., 2006; Jacob et al., 2008; Hariharan et al., 2010; Zhang & Yeung, 2010) are variants of the special cases of (6). They either learn a feature structure $\mathbf{\Sigma}$ (but ignore the task structure) or include a task relation $\mathbf{\Omega}$ (but ignore the feature representation).

5.3 Sparse Covariance Selection in the Matrix-Normal Penalty

Covariance selection enforces zero entries in the Gaussian inverse covariance and thus discovers conditional independence between variables (Dempster, 1972; Banerjee et al., 2008; Friedman et al., 2007). Use of the matrix-normal density in (6) enables us to perform covariance selection to select task and feature structures. When $\mathbf{\Omega}$ in (6) has a sparse inverse, task pairs corresponding to zero entries in $\mathbf{\Omega}^{-1}$ are not explicitly coupled. Similarly, a zero entry in $\mathbf{\Sigma}^{-1}$ indicates no direct interaction between two corresponding features.

Formally, we rewrite (6) to include two additional $\ell 1$ penalties on the inverse of $\mathbf{\Omega}$ and $\mathbf{\Sigma}$:

$$\mathcal{L} = \sum_{t=1}^{m}\sum_{i=1}^{n_t} L(y_i^{(t)}, \mathbf{x}_i^{(t)}, \mathbf{W}(t,:)) + \lambda[p \log |\mathbf{\Omega}| + m \log |\mathbf{\Sigma}| + tr\{\mathbf{\Omega}^{-1}\mathbf{W}\mathbf{\Sigma}^{-1}\mathbf{W}^T\}] + \lambda_\Omega ||\mathbf{\Omega}^{-1}||_1 + \lambda_\Sigma ||\mathbf{\Sigma}^{-1}||_1 \tag{9}$$

where $||\ ||_1$ is the $\ell 1$-norm of a matrix, and λ_Ω and λ_Σ control the strength of two $\ell 1$ penalties. Due to the additional penalties, optimizing $\mathbf{\Omega}$ and $\mathbf{\Sigma}$ given $\mathbf{W}$ becomes a new problem:

$$\min_{\mathbf{\Omega},\mathbf{\Sigma}}\ p \log |\mathbf{\Omega}| + m \log |\mathbf{\Sigma}| + tr\{\mathbf{\Omega}^{-1}\mathbf{W}\mathbf{\Sigma}^{-1}\mathbf{W}^T\} + \frac{\lambda_\Omega}{\lambda}||\mathbf{\Omega}^{-1}||_1 + \frac{\lambda_\Sigma}{\lambda}||\mathbf{\Sigma}^{-1}||_1 \tag{10}$$

To solve (10), as in the flip-flop algorithm (5), we iteratively optimize $\mathbf{\Omega}$ and $\mathbf{\Sigma}$ until convergence:

$$\begin{cases} \hat{\mathbf{\Omega}} &= \operatorname{argmin}_{\mathbf{\Omega}}\ p \log |\mathbf{\Omega}| + tr\{\mathbf{\Omega}^{-1}(\mathbf{W}\mathbf{\Sigma}^{-1}\mathbf{W}^T)\} + \frac{\lambda_\Omega}{\lambda}||\mathbf{\Omega}^{-1}||_1 \\ \hat{\mathbf{\Sigma}} &= \operatorname{argmin}_{\mathbf{\Sigma}}\ m \log |\mathbf{\Sigma}| + tr\{\mathbf{\Sigma}^{-1}(\mathbf{W}^T\hat{\mathbf{\Omega}}^{-1}\mathbf{W})\} + \frac{\lambda_\Sigma}{\lambda}||\mathbf{\Sigma}^{-1}||_1 \end{cases} \tag{11}$$

Note that both equations in (11) are $\ell 1$ regularized covariance selection problems, for which efficient optimization has been intensively studied (Banerjee et al., 2008; Friedman et al., 2007).

5.4 Experimental Results

In our experiments (Zhang & Schneider, 2010a), we study a landmine detection problem and a face recognition problem, where multiple tasks correspond to detecting landmines at different landmine fields and classifying faces between different subjects, respectively. We compare to recent multi-task learning methods that either infer the feature structure (Argyriou et al., 2006) or the task relation (Jacob et al., 2008). Experiments are conducted with varied amounts of training samples and paired T-tests (over 30 random runs) are provided. Experimental results show that the sparse matrix-normal regularization provides a flexible framework to couple multiple tasks and outperform the competitors with statistical significance.

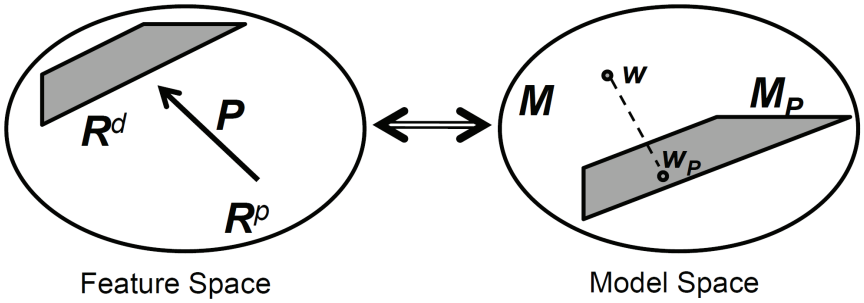

Figure 1: The idea of projection penalties (linear cases)

Part II
Learning with Extra Information about Features

In Part II, we focus on encoding extra information about the feature space into learning. Extra information about feature structures can usually be conveyed by a dimensionality reduction, e.g., a subset of important features, a clustering of low-level features, a general feature subspace or manifold. Directly performing a feature reduction, however, can potentially lead to loss of information and predictive power. In Section 6, we propose the projection penalty framework that effectively encodes an arbitrary feature reduction into learning but avoids the risk of information loss. In Section 7, we consider multi-view learning, where projection penalties offers an opportunity to simultaneously address both overfitting and underfitting.

6 Projection Penalties: Dimensionality Reduction without Loss (Completed)

In this section, we propose the projection penalty (Zhang & Schneider, 2010c): reducing the feature space can be viewed as restricting the model search to certain model subspace; instead of directly imposing such a restriction, we can search in the full model space but penalize the projection distance to the model subspace. In this sense, information from the feature reduction is used to *guide* the model search rather than to completely *restrict* the model search to the reduced model subspace. As a result, projection penalties encode a feature reduction into learning while alleviate the risk of information loss.

6.1 Linear Cases

The idea of projection penalties in linear cases is shown in Fig. 1. A linear feature reduction P is an $d \times p$ matrix that projects data from R^p to R^d, where p and d are the dimension of the original and reduced feature space. This feature reduction is equivalent to a restriction of the model space $\mathbf{M} \to \mathbf{M}_P$, and we propose to learn the model $\mathbf{w}$ in the full space $\mathbf{M}$ but penalize the projection distance to the model subspace $\mathbf{M}_P$.

Given a feature reduction P, learning models in the reduced feature space can be formulated as:

$$\operatorname*{argmin}_{\mathbf{v}\in R^d,b} \sum_{i=1}^{n} L(y_i, \mathbf{v}^T(P\mathbf{x}_i) + b) \tag{12}$$

where $\{\mathbf{x}_i \in R^p, y_i\}_{i=1}^n$ are n training examples in the original feature space, P is an $d \times p$ linear reduction, $(\mathbf{v} \in R^d, b)$ is the model in the reduced space, and L is the empirical loss. This can be rewritten as:

$$\operatorname*{argmin}_{\mathbf{v}\in R^d,b} \sum_{i=1}^{n} L(y_i, (P^T\mathbf{v})^T\mathbf{x}_i + b) \tag{13}$$

Note that $P^T\mathbf{v} \in R^p$ has only d degrees of freedom as $\mathbf{v} \in R^d$. Define $\mathbf{w} = P^T\mathbf{v}$, eq. (13) is equivalent to:

$$\operatorname*{argmin}_{\mathbf{w}\in \mathcal{M}_P,b} \sum_{i=1}^{n} L(y_i, \mathbf{w}^T\mathbf{x}_i + b) \tag{14}$$

where $\mathcal{M}_P$ is a model subspace in R^p defined as:

$$\mathcal{M}_P = \{\mathbf{w} \in R^p \mid \mathbf{w} = P^T\mathbf{v},\ \exists \mathbf{v} \in R^d\} \tag{15}$$

In (14) we see that performing a linear feature reduction P is equivalent to restricting the model search to a model subspace $\mathcal{M}_P$ as defined in (15). The risk of such a restriction is that, although P highlights the relevant part of the feature space, the optimal model does not necessarily belong to the model subspace $\mathcal{M}_P$. Thus, we propose to search models in the full model space and penalize the projection distance to $\mathcal{M}_P$. This leads to *the formulation of projection penalties for linear feature reduction*:

$$\operatorname*{argmin}_{\mathbf{w}\in R^p,b} \sum_{i=1}^{n} L(y_i, \mathbf{w}^T\mathbf{x}_i + b) + \min_{\mathbf{w}_P\in\mathcal{M}_P} \lambda J(\mathbf{w} - \mathbf{w}_P) \tag{16}$$

where λ is a regularization parameter, J is a penalty function such as the $\ell 2$ or $\ell 1$ norm, and $\mathbf{w}_P$ is the *projection* of $\mathbf{w}$ onto $\mathbf{M}_P$ under the penalty measure J. Optimization of the projection penalty formula (16) is detailed in (Zhang & Schneider, 2010c), which depends on the choice of empirical loss L and penalty J.

6.2 Kernel-Based and Other Nonlinear Cases

The usefulness of projection penalties is limited if we can only encode a linear feature reduction in learning linear models. Therefore, we extend projection penalties to kernel-based and other nonlinear cases.

The kernel-based projection penalty shares the same idea with the linear case, as shown in Fig 1 and eq. (16), but both the feature reduction $\mathbf{P}$ and the model $\mathbf{w}$ are defined in a reproducing kernel Hilbert space (RKHS). In this case, the feature reduction $\mathbf{P}$ maps the data to an RKHS and performs a linear reduction from this RKHS to a low-dimensional subspace. Our goal is to encode the information of this reduction to learn a model $\mathbf{w}$ in the RKHS. To attain this goal, we develop the representer theorem and the dual optimization for kernel-based projection penalties (Zhang & Schneider, 2010c).

In (Zhang & Schneider, 2010c) we also study projection penalties for a given nonlinear feature reduction that is not linear in either the original or any kernel feature space. An example of such a nonlinear feature reduction is a fully generative topic model like latent Dirichlet allocation.

6.3 Experimental Results

In the empirical study (Zhang & Schneider, 2010c), we apply projection penalties to various dimension reduction techniques in different applications, including: 1) principal component analysis and partial least squares in housing price forecasting; 2) kernel PCA, generalized discriminant analysis and Orthogonal Laplacianfaces in face recognition; 3) latent Dirichlet allocation in text classification. Prediction is always improved by using the projection penalty instead of directly performing the reduction. This indicates that the projection penalty is a more effective and reliable way to encode the information from a feature reduction.

7 Projection Penalties for Multi-View Learning

In the presence of multiple views, researchers are confronted with a dilemma. To address *overfitting*, we should emphasize the consistency of multiple views in order to *restrict and regularize* the model space. To handle *underfitting*, on the other hand, we should try to combine the information from multiple views in order to further *expand* the feature and model space. As a result, we consider a multi-view projection penalty framework to address both overfitting and underfitting, based on the following observations:

- The projection penalty is an effective way to find a trade-off between a high-dimensional rich feature space and a low-dimensional restricted feature space.
- Recent research has shown that canonical correlation analysis (CCA) can potentially find a low-dimensional feature space that preserves the information from multiple views (Foster et al., 2008).
- We can learn in a jointly augmented feature space (constructed from multiple views) and apply the projection penalty to the shared low-dimensional feature space (extracted by CCA).

Part III
Learning with Extra Information about Labels

In Part III, we focus on encoding extra information about the label space into learning. In Section 8, we propose error-correcting output codes (ECOCs) for multi-label classification (Zhang & Schneider, 2010b), where predictable directions in the label space are extracted by canonical correlation analysis (CCA) and included into the output code for error correction. In Section 9, we analyze the link between CCA and Partial Least Squares and propose a new procedure to generate more effective output codes.

8 Multi-Label ECOCs with Canonical Correlation Analysis

Error-correcting output codes (ECOCs) are traditionally designed to decompose a multiclass classification problem into a set of binary problems (Dietterich & Bakiri, 1994). As a result, the multiclass problem can be solved using only binary classifiers, and the binary problems also provide a redundant representation to correct prediction errors. Unlike classes, labels in multi-label classification are no longer mutually exclusive. In a q-label problem, the cardinality of the output space $\mathcal{Y} = \{0, 1\}^q$ is 2^q instead of q in a q-class problem. This change of output space in multi-label classification presents new challenges to output coding:

- **Validity of the encoding**. An ECOC encodes the target problem by decomposing it into a number of binary decision problems, each differentiating two subsets of classes. In multi-label problems, however, two subsets of labels can be simultaneously satisfied by certain examples, which makes the binary decision ill-defined on these examples. Ideally, the encoding should be well-defined for all training examples and future testing examples.
- **Efficiency of the decoding**. In multiclass ECOCs, the decoding for each test example is usually performed by searching over all q possible classes and choosing the one that optimizes certain criteria. In multi-label problems, however, searching over 2^q label vectors for a q-label problem is inefficient.
- **Predictability of the codeword**. In ECOCs, codewords need to be predicted by models and thus predictability of codewords is an important concern for encoding. In multi-label ECOCs, encoding via binary problems is ill-defined and a new encoding is needed to produce both valid and predictable codewords.
- **Dependency among labels**. In multiclass problems, classes are mutually exclusive. This eliminates most dependency structures among classes except weak negative correlation. In multi-label problems, however, various dependency structures exist among labels and should be exploited in the output code design.

In this work, we propose an error-correcting output code for multi-label classification. This ECOC provides valid encoding, efficient decoding, and predictable codewords that exploit the label dependency.

8.1 Canonical Correlation Analysis

Consider a set of p variables $\mathbf{x} \in \mathcal{X} \subseteq R^p$ and another set of q variables $\mathbf{y} \in \mathcal{Y} \subseteq R^q$. For a multi-label problem, $\mathbf{x}$ denotes the feature vector and $\mathbf{y}$ denotes the label vector. In this case, we have $\mathcal{Y} = \{0,1\}^q$. In addition, we have a training set of n observations: $\mathbf{D} = (\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^n$, where $\mathbf{X}$ and $\mathbf{Y}$ are of size $n \times p$ and $n \times q$, respectively. Canonical correlation analysis starts with seeking a pair of projection directions $\mathbf{u} \in R^p$ and $\mathbf{v} \in R^q$, such that the *correlation* between $\mathbf{u}^T\mathbf{x}$ and $\mathbf{v}^T\mathbf{y}$ is maximized[2]:

$$\underset{\mathbf{u}\in R^p, \mathbf{v}\in R^q}{\operatorname{argmax}} \quad \frac{\mathbf{u}^T\mathbf{X}^T\mathbf{Y}\mathbf{v}}{\sqrt{(\mathbf{u}^T\mathbf{X}^T\mathbf{X}\mathbf{u})(\mathbf{v}^T\mathbf{Y}^T\mathbf{Y}\mathbf{v})}} \tag{17}$$

This problem can be solved as a generalized eigenproblem (Hardoon et al., 2004), where the solution provides multiple pairs of projection vectors $(\mathbf{u}, \mathbf{v})$. By solving for the first d principal eigenvectors, we can obtain d pairs of projection vectors: $\{(\mathbf{u}_j, \mathbf{v}_j)\}_{j=1}^d$. We denote this process as:

$$\{(\mathbf{u}_j, \mathbf{v}_j)\}_{j=1}^d \leftarrow \text{CCA}(\mathbf{X}, \mathbf{Y}) \tag{18}$$

8.2 Multi-label ECOCs: Encoding

In our encoding, CCA in (18) plays a key role: the d canonical output variates $\{\mathbf{v}_j^T\mathbf{y}\}_{j=1}^d$ are well known as the most predictable variates (Hotelling, 1935), which are ideal candidates to be included in the codeword

[2]For simplicity, one can assume that the data have been centralized such that each dimension has zero mean.

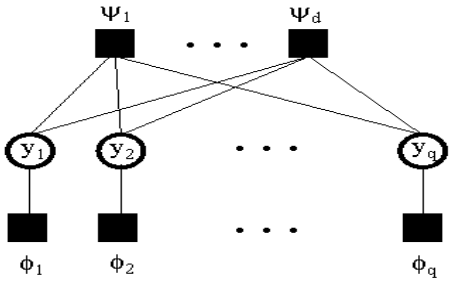

Figure 2: The factor graph representation of undirected graphical model used for decoding the output.

for correcting prediction errors. For an example $\mathbf{x}$ with the label vector $\mathbf{y} = \{y_1, \ldots, y_q\}$ and canonical output variates $\{\mathbf{v}_j^T\mathbf{y}\}_{j=1}^d$, the output encoding for $\mathbf{x}$ is:

$$\mathbf{z} = (y_1, \ldots, y_q, \mathbf{v}_1^T\mathbf{y}, \ldots, \mathbf{v}_d^T\mathbf{y})^T \tag{19}$$

Given the training set $(\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^n$, we will learn q classifiers $\{\hat{p}_1, \ldots, \hat{p}_q\}$ to predict the q labels $\{y_1, \ldots, y_q\}$, and d regression models $\{\hat{m}_1, \ldots, \hat{m}_q\}$ to predict the d canonical variates $\{\mathbf{v}_1^T\mathbf{y}, \ldots, \mathbf{v}_d^T\mathbf{y}\}$.

8.3 Multi-label ECOCs: Decoding

For a test example $\mathbf{x}$, each learned classifier $\hat{p}_j$ predicts a Bernoulli distribution $\phi_j(y_j)$ for a label y_j:

$$\phi_j(y_j) = \hat{p}_j(\mathbf{x})^{y_j}(1 - \hat{p}_j(\mathbf{x}))^{(1-y_j)}, \quad j = 1, 2, \ldots, q \tag{20}$$

and each regression model $\hat{m}_k$ predicts a Gaussian distribution $\psi_k(\mathbf{y})$ for a canonical variate $\mathbf{v}_d^T\mathbf{y}$:

$$\psi_k(\mathbf{y}) \;\propto\; \exp -\frac{(\mathbf{v}_k^T\mathbf{y} - \hat{m}_k(\mathbf{x}))^2}{2\hat{\sigma}_k^2}, \quad k = 1, 2, \ldots, d \tag{21}$$

where the variance term $\hat{\sigma}_k^2$ is estimated by cross validation on the training examples.

In the decoding, the predictive distributions (20) and (21) can be represented as a factor graph in Figure 2, and we have the following joint probability for the label vector $\mathbf{y}$ of the test example $\mathbf{x}$:

$$\log P(\mathbf{y}) = -\log \mathcal{Z} + \sum_{j=1}^{q} \log \phi_j(y_j) + \lambda \sum_{k=1}^{d} \log \psi_k(\mathbf{y}) \tag{22}$$

where $\mathcal{Z}$ is the partition function, and λ is a hyperparameter to balance two types of potentials. Exact inference for $P(\mathbf{y})$ in (22) has a time complexity exponential in q, due to the fact that each Gaussian potential ψ_k in (21) usually involves all the q labels. We consider a mean-field approximation to $P(\mathbf{y})$ in the form:

$$Q(\mathbf{y}) = \prod_{j=1}^{q} Q_j(y_j) \tag{23}$$

$Q(\mathbf{y})$ is in the class of fully factorized distributions where each $Q_j(y_j)$ is a Bernoulli distribution on label y_j. We minimize the KL divergence $KL(Q||P)$ to find the best Q^* in the class (Jordan et al., 1999). Notice that the resulting approximation $Q^*(\mathbf{y}) = \prod_{j=1}^q Q_j^*(y_j)$ provides a set of classifiers $\{Q_j^*(y_j)\}_{j=1}^q$ on labels.

9 Optimal Code Design: Unifying CCA and Partial Least Squares

In our multi-label ECOCs, we use the output projections of canonical correlation analysis (CCA) to produce the output code. Careful examination of the recent variants of CCA (Witten et al., 2009; Hardoon & Shawe-Taylor, 2009; Hardoon et al., 2004) and the connection between CCA and Partial Least Squares (Rosipal & Kramer, 2006) suggests a potentially new procedure to produce more effective codes:

- Neither maximizing the *correlation* between projected variables (as in CCA) nor maximizing the *covariance* between projected variables (as in PLS) explicitly optimizes the predictability of the output code.
- Sparsity can be imposed on the output projections of CCA. This gives a sparse encoding matrix and sparse factor graph in decoding, which captures localized label dependency and improves decoding efficiency.
- Both CCA and PLS can be viewed by an iterative extraction-and-deflation procedure. For output code design, a new deflation that is different from both CCA and PLS may be proposed, which can potentially produce more output projections (and thus more redundancy in codewords) than the standard CCA.

Part IV
Active Learning with Extra Information

10 Multi-Task Active Learning with Output Constraints

In this section, we consider active learning for multiple prediction tasks when their outputs are coupled by constraints (Zhang, 2010). A cross-task value of information criterion is proposed, which encodes output constraints to represent not only the uncertain of the prediction for each task but also the inconsistency of predictions across tasks. A specific example of this criterion leads to the cross entropy between the predictive distributions of coupled tasks, which generalizes the notion of entropy in single-task uncertainty sampling.

10.1 Value of Information for Active Learning

We want to learn a classifier $\hat{p}$: given an example $\mathbf{x}$ from the input space $\mathcal{X}$, we can predict the conditional probability of the label Y: $\hat{p}(Y = y|\mathbf{x}),\ \forall y \in \mathcal{Y}$. In pool-based active learning, we choose from a pool of unlabeled samples $\mathbf{U}$ for labeling. To estimate how useful labeling a sample $\mathbf{x} \in \mathbf{U}$ is for improving the current model $\hat{p} = \hat{p}(Y|\mathbf{x})$, we measure the value of information (Krause & Guestrin, 2009) for $(Y, \mathbf{x})$:

$$VOI(Y, \mathbf{x}) = \sum_y \hat{p}(Y = y|\mathbf{x}) R(\hat{p}, Y = y, \mathbf{x}) \tag{24}$$

which is the sum of the *reward* $R(\hat{p}, Y = y, \mathbf{x})$ of each labeling outcome $Y = y$, weighted by the estimated probability $\hat{p}(Y = y|\mathbf{x})$ of each outcome. Different reward functions are available, and two examples are:

$$R(\hat{p}, Y = y, \mathbf{x}) = -\log_2 \hat{p}(Y = y|\mathbf{x}) \tag{25}$$

$$R(\hat{p}, Y = y, \mathbf{x}) = 1 - \delta(y, \operatorname*{argmax}_{y'} \hat{p}(Y = y'|\mathbf{x})) \tag{26}$$

The reward (25) is the Shannon information content of the outcome $Y = y$ given the distribution $\hat{p}$. An impossible outcome (with $\hat{p} = 0$) has an infinite reward, and an already known outcome (with $\hat{p} = 1$) has no reward. The second reward (26) takes the value 0 if the labeling outcome y agrees with the model prediction y' and takes the value 1 otherwise. Incorporating (25) or (26) into the framework (24), we have:

$$VOI(Y, \mathbf{x}) = -\sum_y \hat{p}(Y = y|\mathbf{x}) \log_2 \hat{p}(Y = y|\mathbf{x}) \quad (27)$$

$$VOI(Y, \mathbf{x}) = 1 - \max_y \hat{p}(Y = y|\mathbf{x}) \quad (28)$$

which are entropy-based uncertain sampling and least-confident sampling, respectively (Settles, 2009).

10.2 Cross-Task Value of Information

Consider a set of T tasks, each with a (categorical) response variable Y_i, $i = 1, 2, \ldots, T$. Our goal is to learn a classifier for each task: $\hat{p}_i = \hat{p}_i(Y_i|\mathbf{x}), i = 1, 2, \ldots, T$. Each sample in our training set $\mathbf{x} \in \mathbf{U}$ is associated with T labels. We use $UL(\mathbf{x})$ to denote unknown labels on $\mathbf{x}$: $UL(\mathbf{x}) = \{Y_i : Y_i \text{ is unknown for } \mathbf{x}\}$. In multi-task active learning, we need to measure the value of information for requesting a label Y_i on a sample $\mathbf{x}$, as follows:

$$VOI(Y_i, \mathbf{x}) = \sum_{y_i} \hat{p}_i(Y_i = y_i|\mathbf{x}) R(Y_i = y_i, \mathbf{x}) \quad (29)$$

where $R(Y_i = y_i, \mathbf{x})$ is the reward for a possible labeling outcome $(Y_i = y_i, \mathbf{x})$ for *all* tasks. Given a set of constraints $\mathbf{C}$ among task outputs, labeling outcome for one label Y_i can provide information for other tasks. Therefore, we define the set of *propagated outcomes* as the outcomes that can be *inferred* from $Y_i = y_i$:

$$Prop_{\mathbf{C}}(Y_i = y_i) = \{Y_j = y_j \mid Y_i = y_i \rightarrow_{\mathbf{C}} Y_j = y_j\} \quad (30)$$

Examples of such inference between task outputs include agreement (in multi-view learning), mutual exclusion, inheritance, etc. For example, if $\mathbf{x}$ is a *politician* then $\mathbf{x}$ is also a *person* ($Y_i = 1 \rightarrow Y_j = 1$).

Using the notion of propagated outcomes, we rewrite the cross-task value of information (29) as follows:

$$VOI(Y_i, \mathbf{x}) = \sum_{y_i} \hat{p}_i(Y_i = y_i|\mathbf{x}) \sum_{\substack{Y_j = y_j \in Prop_{\mathbf{C}}(Y_i = y_i) \\ Y_j \in UL(\mathbf{x})}} R(\hat{p}_j, Y_j = y_j, \mathbf{x}) \quad (31)$$

where $R(\hat{p}_j, Y_j = y_j, \mathbf{x})$ is the reward of an inferred outcome $Y_j = y_j$ for the model $\hat{p}_j$, as (25) or (26).

A Case Study. If we plug the reward in (25) into the framework (31), we have:

$$VOI(Y_i, \mathbf{x}) = \sum_{y_i} \hat{p}_i(Y_i = y_i|\mathbf{x}) \sum_{\substack{Y_j = y_j \in Prop_{\mathbf{C}}(Y_i = y_i) \\ Y_j \in UL(\mathbf{x})}} -\log_2 \hat{p}_j(Y_j = y_j|\mathbf{x}) \quad (32)$$

This new criterion in (32) can be viewed as the sum of cross entropy between the model $\hat{p}_i$ and other coupled models $\hat{p}_j$. Recall that the cross entropy of two distributions $P_i(y)$ and $P_j(y)$ is defined as:

$$H(P_i, P_j) = -\sum_y P_i(y) \log_2 P_j(y) \quad (33)$$

$H(P_i, P_j)$ increases with the discrepancy of P_j from P_i, so it captures the inconsistency of two distributions. Note that P_i and P_j in (33) are defined on the same quantity y, but any two predicted distributions $\hat{p}_i$ and $\hat{p}_j$ in (32) are defined on different task outputs. In this sense, the constraints $\mathbf{C}$ plays a key role to couple the predicted distributions of different tasks. As a result, cross-task VOI in (32) is essentially the sum of cross entropy between the predicted distribution $\hat{p}_i$ and other coupled predicted distributions $\hat{p}_j$. Maximizing (32) will select the sample-task pair $(Y_i, \mathbf{x})$ whose model prediction $\hat{p}_i$ is contradicting other models.

10.3 Empirical Study and Future Work

We conduct our empirical study on web information extraction and document classification (Zhang, 2010). Results on both problems demonstrate the effectiveness of the cross-task value of information in collecting labeled examples for multiple coupled tasks. In the future work we will study: 1) probabilistic constraints, e.g., $P(Y_j = y_j|Y_i = y_i) = 0.9$; 2) class imbalance and active learning; 3) the connection between active learning and semi-supervised learning, e.g., cross-task value of information for semi-supervised learning.

Part V
Summary and Schedule

We summarize the completed work and provide a timeline for the future work in Table 1.

Table 1: Summary and Schedule.

Research Tasks	*Status and Schedule*
Learning with Extra Information about Models	
Learning semantic word correlation from unlabeled text	[NIPS 2008]
Learning compressible models	[SDM 2010]
Learning multiple tasks with a sparse matrix-normal penalty	[NIPS 2010]
Learning with Extra Information about Labels	
Multi-label ECOCs using CCA	Winter 2011 & Spring 2011
Improving multi-label ECOCs	Spring 2011 & Summer 2011
Active learning with label constraints	[AAAI 2010] & Fall 2011
Learning with Extra Information about Features	
Projection penalties	[ICML 2010]
Multi-view learning with projection penalty	Fall 2011
Others	
Applications to other problems	Fall 2011 & Winter 2012
Thesis writing	Winter 2012 -

References

Allwein, E. L., Schapire, R. E., & Singer, Y. (2001). Reducing multiclass to binary: a unifying approach for margin classifiers. *J. Mach. Learn. Res.*, *1*, 113–141.

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

Argyriou, A., Evgeniou, T., & Pontil, M. (2006). Multi-task feature learning. *NIPS*.

Argyriou, A., Micchelli, C. A., Pontil, M., & Ying, Y. (2007). A spectral regularization framework for multi-task structure learning. *NIPS*.

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

Banerjee, O., Ghaoui, L. E., & d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *J. Mach. Learn. Res.*, *9*, 485–516.

Baraniuk, R. G., Candes, E. J., Nowak, R., & Vetterli, M. (2008). Compressive Sampling (Special Issue). *IEEE Signal Processing Magazine*, *25*, 12–101.

Baxter, J. (1995). Learning Internal Representations. *COLT* (pp. 311–320).

Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, *7*, 2399–2434.

Bonilla, E., Chai, K. M., & Williams, C. (2008). Multi-task gaussian process prediction. In J. Platt, D. Koller, Y. Singer and S. Roweis (Eds.), *Nips*, 153–160.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *COLT '92: Proceedings of the fifth annual workshop on Computational learning theory* (pp. 144–152).

Brown, P. J., & Vannucci, M. (1998). Multivariate Bayesian Variable Selection and Prediction. *Journal of the Royal Statistical Soceity, Series B, 60(3)*, 627–641.

Candes, E. J. (2006). Compressive Sampling. *Proceedings of International Congress of Mathematicians*.

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

Chen, J., Tang, L., Liu, J., & Ye, J. (2009). A Convex Formulation for Learning Shared Structures from Multiple Tasks. *ICML*.

Christopoulos, C., Skodras, A., & Ebrahimi, T. (2000). The JPEG2000 Still Image Coding System: An Overview. *IEEE Trans. Consumer Electronics*, *46(4)*, 1103–1127.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Mach. Learn.*, *20*, 273–297.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York, NY, USA: Wiley-Interscience.

Crammer, K., & Singer, Y. (2002). On the learnability and design of output codes for multiclass problems. *Mach. Learn.*, *47*, 201–233.

Dawid, A. P. (1981). Some matrix-variate distribution theory: Notational considerations and a bayesian application. *Biometrika*, *68*, 265–274.

Dempster, A. P. (1972). Covariance selection. *Biometrics*.

Dietterich, T. G., & Bakiri, G. (1994). Solving multiclass learning problems via error-correcting output codes. *J. Artif. Int. Res.*, *2*, 263–286.

Donoho, D. L. (2006). Compressed Sensing. *IEEE Trans. Information Theory*, *52(4)*, 1289–1306.

Dutilleul, P. (1999). The MLE Algorithm for the Matrix Normal Distribution. *J. Statist. Comput. Simul.*, *64*, 105–123.

Escalera, S., Pujol, O., & Radeva, P. (2010). On the decoding process in ternary error-correcting output codes. *IEEE Trans. Pattern Anal. Mach. Intell.*, *32*, 120–134.

Foster, D., Kakade, S., , & Zhang, T. (2008). *Multi-view dimensionality reduction via canonical correlation analysis* (Technical Report TTIC-TR-2008-4). Toyota Technological Institute at Chicago.

Freund, Y., Seung, S. H., Shamir, E., & Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning*, *28*, 133–168.

Friedman, J., Hastie, T., & Tibshirani, R. (2007). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*.

Fyfe, C., & Lai, P. L. (2001). Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, *10*, 365–374.

Gupta, A. K., & Nagar, D. K. (1999). *Matrix variate distributions*. Chapman Hall.

Hardoon, D. R., & Shawe-Taylor, J. (2009). Sparse canonical correlation analysis. http://arxiv.org/abs/0908.2724.

Hardoon, D. R., Szedmak, S. R., & Shawe-taylor, J. R. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural Comput.*, *16*, 2639–2664.

Hariharan, B., Vishwanathan, S., & Varma, M. (2010). Large scale max-margin multi-label classification with priors. *ICML*.

Hastie, T., & Tibshirani, R. (1997). Classification by pairwise coupling. *NIPS '97*.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference and prediction*. New York: Springer.

Hotelling, H. (1935). The most predictable criterion. *Journal of Educational Psychology, 26*, 139–142.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, *28*, 321–377.

Huang, J., Zhang, T., & Metaxas, D. (2009). Learning with structured sparsity. *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning* (pp. 417–424).

Jacob, L., Bach, F., & Vert, J.-P. (2008). Clustered multi-task learning: A convex formulation. *NIPS*.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. (1999). An introduction to variational methods for graphical models. *Machine Learning, 37*, 183–233.

Krause, A., & Guestrin, C. (2009). Optimal value of information in graphical models. *J. Artif. Intell. Res., 35*, 557–591.

Lewis, D. D., & Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. *ICML '94* (pp. 148–156).

Muslea, I., Minton, S., & Knoblock, C. A. (2006). Active learning with multiple views. *J. Artif. Intell. Res., 27*, 203–233.

Obozinski, G., Taskar, B., & Jordan, M. I. (2009). Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*.

Passerini, A., Pontil, M., & Frasconi, P. (2004). New results on error correcting output codes of kernel machines. *IEEE Transactions on Neural Networks*, 45–54.

Pujol, O., Radeva, P., & Vitria, J. (2006). Discriminant ecoc: A heuristic method for application dependent design of error correcting output codes. *IEEE Trans. Pattern Anal. Mach. Intell., 28*, 1007–1012.

Qi, G.-J., Hua, X.-S., Rui, Y., Tang, J., & Zhang, H.-J. (2008). Two-dimensional active learning for image classification. *CVPR*.

Rai, P., & Daume, H. (2009). Multi-label prediction via sparse infinite cca. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams and A. Culotta (Eds.), *Advances in neural information processing systems 22*, 1518–1526.

Raina, R., Ng, A. Y., & Koller, D. (2006). Constructing informative priors using transfer learning. *ICML '06: Proceedings of the 23rd international conference on Machine learning* (pp. 713–720).

Reichart, R., Tomanek, K., Hahn, U., & Rappoport, A. (2008). Multi-task active learning for linguistic annotations. *Proceedings of ACL: HLT* (pp. 861–869).

Rosipal, R., & Kramer, N. (2006). Overview and recent advances in partial least squares. *in Subspace, Latent Structure and Feature Selection Techniques, Lecture Notes in Computer Science* (pp. 34–51). Springer.

Roth, D., & Small, K. (2006). Margin-based active learning for structured output spaces. *ECML* (pp. 413–424).

Roy, N., & McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning* (pp. 441–448).

Rudin, L., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D*, *60*, 259–268.

Settles, B. (2009). *Active learning literature survey*Computer Sciences Technical Report 1648). University of Wisconsin–Madison.

Seung, H. S., Opper, M., & Sompolinsky, H. (1992). Query by committee. *COLT '92: Proceedings of the fifth annual workshop on Computational learning theory* (pp. 287–294).

Spanias, A. S. (1994). Speech Coding: A Tutorial Review. *Proceedings of the IEEE, 82(10),* 1541–1582.

Thrun, S., & O'Sullivan, J. (1996). Discovering Structure in Multiple Learning Tasks: The TC Algorithm. *ICML* (pp. 489–497).

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Soceity, Series B, 58(1),* 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., & Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal Of The Royal Statistical Society Series B, 67*, 91–108.

Tikhonov, A. N., & Arsenin, V. Y. (1977). *Solutions of ill-posed problems*. Winston and Sons.

Tong, S., & Koller, D. (2002). Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res., 2*, 45–66.

Wallace, G. K. (1992). The JPEG Still Picture Compression Standard. *IEEE Trans. Consumer Electronics, 38(1),* xviii–xxxiv.

Witten, D. M., Tibshirani, R., & Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, *10*, 515–534.

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

Yu, K., Chu, W., Yu, S., Tresp, V., & Xu, Z. (2007a). Stochastic relational models for discriminative link prediction. *NIPS* (pp. 1553–1560).

Yu, S., Tresp, V., & Yu, K. (2007b). Robust multi-task learning with t-processes. *ICML* (p. 1103).

Yuan, M., Yuan, M., Lin, Y., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B, 68*, 49–67.

Zhang, J., Ghahramani, Z., & Yang, Y. (2006). Learning multiple related tasks using latent independent component analysis. *NIPS* (pp. 1585–1592).

Zhang, Y. (2010). Multi-task active learning with output constraints. *AAAI*.

Zhang, Y., & Schneider, J. (2010a). Learning multiple tasks with a sparse matrix-normal penalty. *NIPS*.

Zhang, Y., & Schneider, J. (2010b). Multi-label output codes using canoical correlatoin analysis. `http://www.cs.cmu.edu/~yizhang1/docs/CCACodingDraft.pdf`.

Zhang, Y., & Schneider, J. (2010c). Projection penalties: Dimension reduction without loss. *ICML*.

Zhang, Y., Schneider, J., & Dubrawski, A. (2008). Learning the Semantic Correlation: An Alternative Way to Gain from Unlabeled Text. *NIPS*.

Zhang, Y., Schneider, J., & Dubrawski, A. (2010). Learning compressible models. *SDM*.

Zhang, Y., & Yeung, D.-Y. (2010). A convex formulation for learning task relationships in multi-task learning. *Proceedings of the Twenty-fourth Conference on Uncertainty in AI (UAI)*.