

# Estimating variable structure and dependence in Multi-task learning via gradients

Justin Guinney<sup>1,2</sup>

Qiang Wu<sup>1,3,4</sup>

Sayan Mukherjee<sup>1,3,4</sup>

JHG9@DUKE.EDU

QIANG@STAT.DUKE.EDU

SAYAN@STAT.DUKE.EDU

<sup>1</sup> *Institute for Genome Sciences & Policy*

<sup>2</sup> *Program in Computational Biology and Bioinformatics*

<sup>3</sup> *Department of Statistical Science*

<sup>4</sup> *Department of Computer Science*

*Duke University*

*Durham, NC 27708, USA*

Editor: ?

## Abstract

We consider the problem of learning gradients in the supervised setting where there are multiple, related tasks. Gradients provide a natural interpretation to the geometric structure of data, and can assist in problems requiring variable selection and dimension reduction. By extending this idea to the multi-task learning (MTL) environment, we present methods for simultaneously learning structure within each task, and the shared structure across all tasks. Our methods are placed within the framework of Tikhonov regularization, providing (a) robustness to high-dimensional data, and (b) a mechanism for incorporating *a priori* knowledge of task (dis)similarity. We provide an implementation for multi-task gradient learning for classification and regression, and demonstrate the utility of our algorithms on simulated and real data.

**Keywords:** multi-task learning, dimension reduction, covariance estimation, inverse regression

## 1. Introduction

In the last decade, researchers in the biological and physical sciences have enjoyed a stunning growth in the number of coordinates or variables they can measure. In many scientific domains, however, the growth of measurable variables has outpaced the ability to produce samples, giving rise to the well-known “large  $p$ , small  $n$ ” problem. The modern day, canonical example is micro-array gene expression data: gene probes number in the many thousands, yet high costs and difficulties in obtaining biological specimens allow only moderate sample sizes. In the effort to construct well-conditioned analyzes, researchers will typically reduce the number of features to the most relevant, or augment the number of samples by combining related data, or tasks. We propose a method that accomplishes both by considering the problem of dimension reduction and feature selection where there are multiple, related tasks.

In this paper, we demonstrate how multi-task learning and gradient learning can be combined into a single framework for inferring the common structure across related data sets. We do this in the supervised setting, where class labels or response variates are available. Our motivation for this is two-fold. First, by treating gradients as a measure of feature covariance (see section 2.2), we can

obtain a sparse, independent feature set for building robust predictive models. Second, by inferring the structure of data across tasks, we gain both global and local intuition about task (dis)similarity. Such information could be used in deciding how, or if, data can be combined in building future models. We are currently unaware of any other algorithm that is capable of estimating feature covariance within the multi-task paradigm for the purposes we describe.

## 2. Estimates of Gradients for Multi-Task Learning

In this section, we formulate the algorithms for learning gradients for the multi-task setting. We first provide some background on the concepts of multi-task learning and learning gradients and then formulate the algorithms for both classification and regression.

### 2.1 Multi-Task Learning

Multi-task learning is the idea of pooling related samples (*tasks*) together in a joint analysis. Previous empirical work has shown that combining tasks in a predictor model can improve performance (Evgeniou et al., 2005; Caruana, 1997; Ben-David and Schuller, 2003), especially under conditions where there are few samples. Although MTL is a specific formulation conceived in the machine learning community, the idea of conjoint data analysis is well-studied in the statistics community under the name of hierarchical models with mixed effects, or hierarchical Bayesian mixture models. These models assume the form of separate task models connected by shared hyper-priors. Both MTL and hierarchical models are premised on the same idea: similar data can be best exploited when studied in a single model, rather than each separately. While definitions of similarity are not uniform, in this paper, we take the definition of *similar* to mean overlapping task features that contribute to the classification or regression function. Our end goal, however, is not just a better classification or regression function. We believe an equally important use of the multi-task model is to uncover shared structure between tasks (dependent task variables) as well as the task specific structure (independent task variables). Recent literature within the machine learning community suggests a burgeoning interest in this direction (Obozinski et al., 2006; Argyriou et al., 2006; Ando and Zhang, 2005; Jebara, 2004).

We now formally define the multi-task framework and introduce notation. We are given  $T$  tasks,  $t \in \{1, \dots, T\}$ , with  $n_t$  samples  $(x_{it}, y_{it})$  with  $x_{it} \in \mathbb{R}^p$ ,  $y_{it} \in \mathbb{R}$  for regression and  $y \in \{-1, 1\}$  for classification, and  $i \in \{1, \dots, n_t\}$ . The total number of samples is  $n = \sum_t n_t$ . We will denote the samples from the task  $t$  as  $D_t$  and  $D$  as the set of all the samples:  $D = \{D_1, \dots, D_T\}$ . The objective in multi-task modeling is to build a regression or classification function,  $F_t(x)$ , for each task  $t$  that has a baseline term  $f_0(x)$  over all tasks and a task specific correction  $f_t(x)$ :

$$F_t(x) = f_0(x) + f_t(x) + \varepsilon, \varepsilon \stackrel{iid}{\sim} \text{No}(0, \sigma^2). \quad (1)$$

Given a convex loss function  $\ell(y, f(x))$ , the empirical error for task  $t$  is defined to be

$$\mathcal{E}_{D_t}(F_t) = \frac{1}{n_t} \sum_{i=1}^{n_t} \ell(y_{it}, F_t(x_{it})).$$

For each  $t$ , one may minimize this quantity to build a classification or regression estimator  $\hat{F}_t$  for the future prediction. However, if the amount of similarity between tasks is known, it may be desirable

to model these tasks simultaneously. Following the approach in Evgeniou and Pontil (2004), we attempt to minimize the averaged error

$$\frac{1}{T} \sum_{t=1}^T \mathcal{E}_{D_t}(f_0, f_t).$$

Moreover, in order to control the stability and reflect the similarity between tasks, we add regularization terms to  $f_0$  and  $f_t$  to obtain the general optimization problem:

$$\arg \min_{(f_0, f_t)} \left\{ \frac{1}{T} \sum_{t=1}^T \mathcal{E}_{D_t}(f_0, f_t) + \frac{\mu}{T} \sum_{t=1}^T \|f_t\|^2 + \lambda \|f_0\|^2 \right\}, \quad (2)$$

where  $\mu$  and  $\lambda$  are regularization parameters. When  $\frac{\mu}{\lambda}$  is large, the model behaves as a single task, while for small  $\frac{\mu}{\lambda}$  the tasks are treated as if they are  $T$  independent tasks.

## 2.2 Learning Gradients for Single-Task, Feature Selection and Dimension Reduction

Before introducing the concept of learning gradients, we first make explicit the distinction between variable selection and dimension reduction. Whereas the former learns sparse features for building robust predictor models (e.g. Recursive Feature Elimination (RFE) (Guyon et al., 2002)), the latter focuses on the intrinsic structure of the data and makes no attempt to preserve the original features (e.g. PCA (Hotelling, 1933)). In mathematical terms, feature selection looks for subspaces in the span of the original features, while dimension reduction locates new bases (as some linear combination of the original features) that capture the phenomenology of the observations. The appropriate method is context dependent, and in many cases it will not depend on the data itself. For example, a clinical device for gene-based cancer prediction would favor a sparse gene selection model due to the high cost of gene expression assays.

Methods for dimension reduction presuppose a certain structure to the data, namely, that it resides on or near a low-dimensional manifold. For variables with a large number of interdependencies, this is often a reasonable assumption. A key problem in most cases is that we are not given points on the manifold, but instead must infer the manifold from coordinates in the ambient space. This requires additional assumptions, and can be trivially reduced to a few considerations such as global vs. local and linear vs. non-linear structure of the data. PCA is an example of a global, linear method, and has been successfully used on many different types of data. In recent years, a number of algorithms have been proposed that can operate on complex, non-linear data, are robust to high-dimensional data, and emphasize the local structure of the data (Mika et al., 1999; Roweis and Saul, 2000; Belkin and Niyogi, 2002; Tenenbaum et al., 2000). However, a key limitation of these techniques is their inability to incorporate response variables. A notable exception is linear discriminant analysis (LDA) (Fisher, 1936), although its assumptions<sup>1</sup> often make it unsuitable for many applied contexts.

The class of methods that considers the intrinsic structure of the data and response variables simultaneously falls under the broad label of inverse regression or simultaneous dimensionality reduction and regression. These ideas were developed in sliced inverse regression (SIR) (Li, 1991),

---

1. In LDA, class density is modeled as multivariate Gaussian, and the between class covariances are assumed to be the same.

(conditional) minimum average variance estimation (MAVE) (Xia et al., 2002), and sliced average variance estimation (SAVE) (Cook and Weisberg, 1991). The focus on these approaches was on linear subspaces. The methods also did not extend to the high-dimensional manifold setting. In series of papers Mukherjee and Zhou (2006); Mukherjee and Wu (2006); Mukherjee et al. (2007), the method of learning gradients was developed to allow for simultaneous dimension reduction and regression in the manifold setting.

The method of learning gradients suggested the simultaneous estimation of the regression or classification function  $f_\phi$  and its gradient  $\nabla f_\phi$ . In these papers, the relevance of gradients is premised on the following two ideas:

1. variable selection: large norms of the partial derivative  $\|\frac{\partial f_\phi}{\partial x}\|$  indicate a large change in the discriminative function  $f_\phi$  and considered more relevant,
2. variable dependence: the inner product between partial derivatives  $\langle \frac{\partial f_\phi}{\partial x^j}, \frac{\partial f_\phi}{\partial x^l} \rangle$  indicate large covariation between the  $j$ -th and  $l$ -th coordinates.

These ideas have motivated feature selection and dimension reduction via gradient estimates.

We now review the formulation of the learning gradients algorithms. We start with a binary classification setting where  $y \in \{-1, 1\}$ . Let  $\phi$  be a convex loss functions, e.g.,  $\phi(t) = \log(1+e^{-t})$ . The optimal classifier is given by

$$f_\phi = \arg \min \mathbb{E}(\phi(yf(x)))$$

in the sense that its sign is the Bayes rule. Suppose that  $f_\phi$  is smooth. We can approximate the function using the first order Taylor expansion, written as

$$f_\phi(x) \approx f_\phi(u) + \nabla f_\phi(x) \cdot (x - u), \text{ for } x \approx u.$$

If a function  $f$  and a vector valued function  $\vec{f} = (f_1, \dots, f_p)$  approximates  $f_\phi$  and its gradient well, then given the data  $D = \{(x_i, y_i)\}_{i=1}^n$ , where  $x \in \mathbb{R}^p$  and  $y \in \{-1, 1\}$ , the expected error

$$\mathbb{E}(yf_\phi(x)) \approx \mathcal{E}_D^\phi(f, \vec{f}) = \frac{1}{n^2} \sum_{i,j=1}^n w_{i,j}^{(s)} \phi(y_i(f(x_j) + \vec{f}(x_i) \cdot (x_i - x_j)))$$

is small, where  $w_{i,j}^{(s)}$  is a weight function with bandwidth  $s$  restricting the locality by  $w_{i,j}^{(s)} \rightarrow 0$  as  $\|x_i - x_j\| \rightarrow 0$ . This intuition leads to estimate  $f_\phi$  and its gradient simultaneously by minimizing the quantity  $\mathcal{E}(f, \vec{f})$ . Regularizing in a reproducing kernel Hilbert space leads to the following algorithm:

$$(f_D, \vec{f}_D) = \arg \min_{(f, \vec{f}) \in \mathcal{H}_K^{p+1}} \left\{ \mathcal{E}_D(f, \vec{f}) + \lambda_1 \|f\|_K^2 + \lambda_2 \|\vec{f}\|_K^2 \right\}, \quad (3)$$

where  $f_D$  and  $\vec{f}_D$  are estimates of  $f_\phi$  and  $\nabla f_\phi$ , respectively,  $\|\vec{f}\|_K^2 = \sum_{i=1}^p \|f_i\|_K^2$ , and  $\lambda_1, \lambda_2$  are regularization parameters. The bandwidth function imposes localization of the samples as required by the Taylor expansion, while the regularization parameters provide numeric stability to the classification and gradient functions estimates.

By the representer theorem (Wahba, 1990; Mukherjee and Wu, 2006) the problem reduces to a finite-dimensional optimization of the coefficients

$$f_D(x) = \sum_{i=1}^n \alpha_i K(x, x_i), \quad \vec{f}_D(x) = \sum_{i=1}^n c_i K(x, x_i),$$

with  $c = (c_1, \dots, c_n) \in \mathbb{R}^{p \times n}$ , and  $\alpha = (\alpha_1, \dots, \alpha_n)^T \in \mathbb{R}^p$ . Assuming a loss function that is twice differentiable, gradient descent methods such as Newton-Raphson can be used to solve the minimization problem and obtain an estimate for the coefficients in an efficient manner.

For the regression setting, the idea is similar where the least square error for a vector valued function  $\vec{f}$  is defined as

$$\mathcal{E}_D(\vec{f}) = \frac{1}{n^2} \sum_{i,j=1}^n w_{ij}^{(s)} (y_i - y_j - \vec{f}(x_i) \cdot (x_j - x_i))^2$$

which should be small if  $\vec{f}$  approximates  $\nabla f_r$ , the gradient of the regression function

$$f_r = \arg \min \mathbb{E}(y - f(x))^2.$$

The learning gradients algorithm is then

$$\vec{f}_D = \arg \min_{f \in \mathcal{H}_K^p} \left\{ \mathcal{E}_D(\vec{f}) + \lambda \|\vec{f}\|_K^2 \right\}.$$

We refer to Mukherjee and Zhou (2006); Mukherjee et al. (2007) for more details.

While we have given a general appraisal of gradients for learning structure in data, we have not provided explicit justification for their use. In Mukherjee and Wu (2006), the gradient was assumed to be a reasonable measure of feature covariation. In Maggioni et al. (2007), this premise is made formal in an expression directly linking gradients and the inverse-regression problem. Given the gradient of the classification or regression function  $\nabla f = \left( \frac{\partial f}{\partial x^1}, \dots, \frac{\partial f}{\partial x^n} \right)^T$  define the gradient outer product (GOP) matrix  $\Gamma$  with  $\Gamma_{i,j} = \left\langle \frac{\partial f}{\partial x^i}, \frac{\partial f}{\partial x^j} \right\rangle$ . The equality is written as

$$\Gamma = \sigma_Y^2 \left( 1 - \frac{\sigma_\epsilon^2}{\sigma_Y^2} \right)^2 \Sigma_X^{-1} \Omega_{X|Y} \Sigma_X^{-1}, \quad (4)$$

where  $\Omega_{x|y} = \text{cov}(\mathbb{E}[X|Y])$ ,  $\sigma_Y^2 = \text{var}(Y)$ , and  $\Sigma_x = \text{cov}(X)$ .

From (4), we see that the gradient learning output  $f_D$  models the forward regression  $Y|X$ , and  $\vec{f}_D$  models the inverse regression  $X|Y$  via the estimate of the GOP matrix  $\hat{\Gamma}$ , where

$$\hat{\Gamma} = c_D^T \mathbf{K} c_D \approx \mathbb{E}(\nabla f \otimes \nabla f)$$

and  $\mathbf{K}$  is the kernel matrix. Important features will have large values along the diagonal of  $\hat{\Gamma}$ , and can be used to build predictive models. Further refinement can be obtained by observing the off-diagonal elements  $\hat{\Gamma}_{i,j}$  as a measure of covariance between features, thereby allowing the selection of significant *and* independent features. Moreover, Mukherjee et al. (2007) demonstrated how a spectral decomposition of the GOP matrix could be used for dimension reduction, as the eigenvectors of the top eigenvalues offered a low-dimensional embedding of the data.

### 2.3 Learning gradients in the multi-task setting

We now adapt the method of learning gradients to the multi-task paradigm. Given  $T$  tasks, our goal is to estimate the regression or classification functions  $\{f_0(x), f_1(x), \dots, f_T(x)\}$  and gradients  $\{\nabla f_0(x), \nabla f_1(x), \dots, \nabla f_T(x)\}$ . These estimates can be used to obtain the GOP matrix specific to each task,  $\Gamma^{(f_t)}$ , and the baseline GOP for all tasks,  $\Gamma^{(f_0)}$ .

As with the single-task gradient learning algorithm, we are motivated by the Taylor expansion and extend it to the multi-task setting:

$$F_t(x) \approx f_0(u) + \nabla f_0(x) \cdot (x - u) + f_t(u) + \nabla f_t(x) \cdot (x - u), \text{ for } x \approx u \quad (5)$$

From the formulation in (5), the sum of gradients introduces a problem of interpretability. To justify this equation, we assume that  $\nabla f_0 \perp \nabla f_t$ , i.e. the task corrected gradient is in the null space of the common gradient. While we believe this is a reasonable assumption given the construction of the multi-task problem, this remains an open question which we hope to explore in future work.

We now derive the methods for multi-task gradient learning (MTGL) for classification and regression. We first consider the classification problem. Suppose each task is a binary classification problem. Given a convex loss function  $\phi$ , we can define the error for task  $t$  as

$$\mathcal{E}_{D_t}^\phi(f_0, f_t, \vec{f}_0, \vec{f}_t) = \frac{1}{n_t^2} \sum_{i,j=1}^{n_t} w_{i,j;t} \phi \left( y_{it} \left( (f_0(x_{jt}) + f_t(x_{jt})) + (\vec{f}_0(x_{jt}) + \vec{f}_t(x_{jt})) \cdot (x_{it} - x_{jt}) \right) \right)$$

where  $\vec{f}_0$  and  $\vec{f}_t$  are vector valued functions and model the gradient of  $f_0$  and  $f_t$  respectively. In order to estimate all these functions simultaneously define the average error as

$$\mathcal{E}_D^\phi \left( f_0, \{f_t\}_{t=1}^T, \vec{f}_0, \{\vec{f}_t\}_{t=1}^T \right) = \frac{1}{T} \sum_{i=1}^T \mathcal{E}_{D_i}^\phi(f_0, f_t, \vec{f}_0, \vec{f}_t).$$

We regularize it in RKHS and minimize the penalized error function to formulate the following algorithm:

$$\begin{aligned} (f_{D,0}, \{f_{D,t}\}_{t=1}^T, \vec{f}_{D,0}, \{\vec{f}_{D,t}\}_{t=1}^T) = \arg \min & \left\{ \mathcal{E}_D^\phi \left( f_0, \{f_t\}_{t=1}^T, \vec{f}_0, \{\vec{f}_t\}_{t=1}^T \right) \right. \\ & \left. + \frac{\lambda}{2} \left( \|f_0\|_K^2 + \nu \|\vec{f}_0\|_K^2 \right) + \frac{\mu}{2T} \sum_{t=1}^T \left( \|f_t\|_K^2 + \nu \|\vec{f}_t\|_K^2 \right) \right\}. \end{aligned} \quad (6)$$

We turn to the regression setting. Since we have the estimates for the functions at given points  $y_{it} \approx F_t(x_{it})$  we need only estimates the gradients. For each task, we define the first order remainder error as

$$\mathcal{E}_{D_t}(\vec{f}_0, \vec{f}_t) = \frac{1}{n_t^2} \sum_{i,j=1}^{n_t} w_{i,j;t} \left( y_{it} - y_{jt} - (\vec{f}_0(x_{jt}) + \vec{f}_t(x_{jt})) \cdot (x_{it} - x_{jt}) \right)^2.$$

Minimizing the regularized error functional leads to the algorithm

$$(\vec{f}_{D,0}, \vec{f}_{D,t}) = \arg \min \left\{ \frac{1}{T} \sum_{t=1}^T \mathcal{E}_{D_t}(\vec{f}_0, \vec{f}_t) + \frac{\lambda}{2} \|\vec{f}_0\|_K^2 + \frac{\mu}{2T} \sum_{t=1}^T \|\vec{f}_t\|_K^2 \right\}. \quad (7)$$

In both algorithms, we expect the  $\vec{f}_0$  and  $\vec{f}_t$  to provide good estimates of the gradients of  $f_0$  and  $f_t$  respectively.

### 3. Optimization

In this section we consider how to solve the algorithms for learning gradients for multitask problems. The key will be to relate them with the one task problem which has been well studied and shown to be efficiently solvable in Mukherjee and Zhou (2006); Mukherjee and Wu (2006).

#### 3.1 Classification

First, we have the following representer theorem.

**Proposition 1** *There exist  $\alpha_{0,t,i}, \alpha_{t,i} \in \mathbb{R}$  and  $c_{0,t,i}, c_{t,i} \in \mathbb{R}^p$  so that*

$$\begin{aligned} f_{D,0} &= \sum_{t=1}^T \sum_{i=1}^{n_t} \alpha_{0,t,i} K(x_{it}, \cdot) & f_{D,t} &= \sum_{i=1}^{n_t} \alpha_{t,i} K(x_{it}, \cdot) \\ \vec{f}_{D,0} &= \sum_{t=1}^T \sum_{i=1}^{n_t} c_{0,t,i} K(x_{it}, \cdot) & \vec{f}_{D,t} &= \sum_{i=1}^{n_t} c_{t,i} K(x_{it}, \cdot) \end{aligned} \quad (8)$$

Proposition 1 allows us to reduce the optimization problem to a finite dimensional one. By plugging the above representation into the optimization problem (6) and setting the partial derivatives to 0, we derive the following equations for the coefficients.

$$\alpha_{0,t,i} = \frac{\mu}{T\lambda} \alpha_{t,i} \quad c_{0,t,i} = \frac{\mu}{T\lambda} c_{t,i} \quad (9)$$

and

$$\begin{aligned} &\frac{1}{n_t^2} \sum_{i=1}^{n_t} w_{i,j;t} \phi'(\Upsilon_{i,j,t}) + \mu \alpha_{t,j} \\ &\frac{1}{n_t^2} \sum_{i=1}^{n_t} w_{i,j;t} \phi'(\Upsilon_{i,j,t}) (x_{it} - x_{jt}) + \mu \nu c_{t,j} \end{aligned} \quad (10)$$

where

$$\begin{aligned} \Upsilon_{i,j,t} &= y_{it} \left[ \sum_{s=1}^T \sum_{l=1}^{n_s} \alpha_{0,s,l} K(x_{ls}, x_{jt}) + \sum_{l=1}^{n_t} \alpha_{t,l} K(x_{lt}, x_{jt}) \right. \\ &\quad \left. + \left( \sum_{s=1}^T \sum_{l=1}^{n_s} c_{0,s,l} K(x_{ls}, x_{jt}) + \sum_{l=1}^{n_t} c_{t,l} K(x_{lt}, x_{jt}) \right) \cdot (x_{it} - x_{jt}) \right] \end{aligned}$$

By (9), we need only solve the coefficients  $\alpha_{t,i}$  and  $c_{t,i}$ . These may be realized by Newton's method to (10). However, since we have  $n(p+1)$  coefficients and equations, using Newton's method directly is problematic when the dimension  $p$  is very large.

Recall that one-task learning gradient algorithms have been shown to reduce to very low dimensional optimization problems in Mukherjee and Zhou (2006); Mukherjee and Wu (2006). We hope the multi-task gradient learning shares the same properties. To see this, we establish the following relation between the one-task and multi-task algorithms.

Denote by  $\delta_{st}$  the Kronecker notion meaning that  $\delta_{st} = 1$  only if  $s = t$  and  $\delta_{st} = 0$  otherwise. Let  $\tilde{W}_t$  be the  $n_t \times n_t$  matrix with entries  $\tilde{W}_t(i, j) = \frac{1}{n_t^2} w_{i,j;t}$  and  $\tilde{W} = \text{diag}(\tilde{W}_1, \dots, \tilde{W}_T)$ .

Denote by  $\tilde{D} = \{(\tilde{x}_i, \tilde{y}_i)_{i=1, \dots, n}\}$  the samples rearranged in task order and  $t_i$  is the task associated with sample  $\tilde{x}_i$ . In addition we defined the kernel

$$\tilde{K}((x, s), (x', t)) = K(x, x') \left( \frac{\mu}{T\lambda} + \delta_{st} \right). \quad (11)$$

We have the following conclusion.

**Proposition 2** *Consider the following one-task learning gradient problem*

$$(g_{\tilde{D}}(x, t), \vec{g}_{\tilde{D}}(x, t)) = \arg \min_{g, \vec{g} \in \mathcal{H}_K^{p+1}} \left\{ \mathcal{E}_{\tilde{D}, \tilde{W}}^\phi(g, \vec{g}) + \mu \|g\|_K^2 + \mu\nu \|\vec{g}\|_K^2 \right\} \quad (12)$$

We have

$$f_{D,0} + f_{D,t} = g_{\tilde{D}}(\cdot, t) \text{ and } \vec{f}_0 + \vec{f}_t = \vec{g}(\cdot, t). \quad (13)$$

**Proof** The conclusion follows by noting that the optimization problem for (12) has the same formula as in (10); see Mukherjee and Wu (2006) for details.  $\blacksquare$

By (9), we have

$$f_{D,0} = \frac{\mu}{T\lambda} \sum_{t=1}^T f_{D,t} \text{ and } \vec{f}_{D,0} = \frac{\mu}{T\lambda} \sum_{t=1}^T \vec{f}_{D,t}. \quad (14)$$

This in connection with (13) implies that the solution to multi-task learning gradient can be solved through a one-task learning gradient problem. The techniques to reduce the matrix size and computational complexity in the single-task learning gradient algorithms then apply. Thus the multitask gradient learning can be solved through a low dimensional optimization problem.

### 3.2 Regression

Similarly, we have the representer theorem.

**Proposition 3** *There exists  $\alpha_{0,t,i}, c_{t,i} \in \mathbb{R}^p$  so that the solution to the problem (7) is*

$$\vec{f}_0 = \sum_t \sum_i \alpha_{0,t,i} K(x_{it}, \cdot) \quad \vec{f}_t = \sum_i c_{t,i} K(x_{it}, \cdot) \quad (15)$$

Plugging the above representation into the optimization problem (7) we reduce the problem to a finite dimensional optimization problem for the coefficients. By setting the partial derivatives to 0, we obtain the following linear system for solving these coefficients:

$$\alpha_{0,t,i} = \frac{\mu}{T\lambda} c_{t,i} \quad (16)$$

and

$$\mu c_{t,j} + B_{t,j} \left( \sum_{s=1}^T \sum_{l=1}^{n_s} K(x_{ls}, x_{jt}) \alpha_{0,s,l} + \sum_{l=1}^{n_t} K(x_{lt}, x_{jt}) c_{t,l} \right) = Y_{t,j} \quad (17)$$

where

$$B_{t,j} = \sum_i \frac{1}{n_t^2} w_{i,j;t} (x_{it} - x_{jt})(x_{it} - x_{jt})^T \quad \text{and} \quad Y_{t,j} = \sum_{i=1}^{n_t} \frac{1}{n_t^2} w_{i,j;t} (y_{it} - y_{jt})(x_{it} - x_{jt}).$$

Let  $\tilde{K}$  be the same as in (11). Let  $\tilde{B}$  be the  $np \times np$  matrix composed by  $T \times T$  blocks where the  $(s, t)$  block is an  $n_{sp} \times n_{tp}$  sub-matrix so that

$$\tilde{B}_{st} = 0 \text{ if } s \neq t \text{ and } \tilde{B}_{st} = \text{diag}(B_{t,1}, \dots, B_{t,n_t}) \text{ if } s = t.$$

Let  $\tilde{Y}_t = (Y_{t,1}^{\mathbf{T}}, \dots, Y_{t,n_t}^{\mathbf{T}})^{\mathbf{T}}$  and  $\tilde{Y} = (\tilde{Y}_1^{\mathbf{T}}, \dots, \tilde{Y}_T^{\mathbf{T}})^{\mathbf{T}}$ . We can rewrite the linear system (17) as

$$\left( \mu I_{np} + \tilde{B}(\tilde{K} \otimes I_p) \right) c = \tilde{Y} \quad (18)$$

where  $c = (c_{1,1}^{\mathbf{T}}, \dots, c_{1,n_1}^{\mathbf{T}}, c_{2,1}^{\mathbf{T}}, \dots, c_{2,n_2}^{\mathbf{T}}, \dots, c_{T,1}^{\mathbf{T}}, \dots, c_{T,n_T}^{\mathbf{T}})^{\mathbf{T}}$ . By the discussion in Mukherjee and Zhou (2006), (18) solves the following one-task learning gradient problem

$$\vec{f}_{\tilde{D}}(x, t) = \arg \min \sum_{i,j} \tilde{W}_{i,j} \left( \tilde{y}_i - \tilde{y}_j - \vec{f}(\tilde{x}_i, t_i) \cdot (\tilde{x}_i - \tilde{x}_j) \right)^2 + \mu \|\vec{f}\|_{\tilde{K}}^2.$$

Moreover, we have

$$\vec{f}_{D,0}(x) + \vec{f}_{D,t}(x) = \vec{f}_{\tilde{D}}(x, t).$$

By (16), we have

$$\vec{f}_{D,0} = \frac{\mu}{T\lambda} \sum_{t=1}^T \vec{f}_{D,t}.$$

Therefore, the above argument shows that the solution to the multi-task learning gradient can be solved through a single-task learning gradient problem, the same as in the classification setting. The techniques discussed in Mukherjee and Zhou (2006) can be used to reduce the solution of the coefficients to a linear system of order  $O(n^2)$  which is efficient if  $n$  is small.

#### 4. Using Gradients in the Multi-Task Setting

In the multi-task setting, we estimate  $T + 1$  matrices where  $\hat{\Gamma}_0$  is the GOP estimate across all the tasks, and  $\hat{\Gamma}_1 \dots \hat{\Gamma}_T$  are the task specific GOP estimates. As in the single-task setting, we can use the GOP matrix for feature selection by choosing features corresponding to large RKHS norms. We can also obtain low-dimensional subspaces of the data for each of the tasks, and for the subspace shared between all tasks using spectral decompositions of the GOP matrices. Although we assumed in our formulation of the MTGL algorithm that the task specific subspaces are in the null space of the common subspace, in practice this is rarely true. We therefore construct a measure of subspace overlap, or similarity, to inform us of the relatedness of these subspaces for the linear case. We can also use this score to measure similarity between tasks. We consider the following measure:

**Definition 4** Let  $A$  and  $B$  be two  $p \times p$  symmetric matrices with entries in  $\mathbb{R}$ , with  $W_A$  a  $d$ -dimensional subspace of  $A$ , and  $W_B$  a  $f$ -dimensional subspace of  $B$ . Also, let  $\{v_1^{(A)}, \dots, v_p^{(A)}\}$ ,  $\{\lambda_1^{(A)} \dots \lambda_p^{(A)}\}$  and  $\{v_1^{(B)} \dots v_p^{(B)}\}$ ,  $\{\lambda_1^{(B)} \dots \lambda_p^{(B)}\}$  be the eigenvectors, eigenvalues of  $A$  and  $B$ , respectively. We define the subspace similarity (SS) score of  $A$  and  $B$  as

$$SS_{score_{A,B}} = \frac{SS_{A \rightarrow B}}{2} + \frac{SS_{B \rightarrow A}}{2} = \frac{\sum_{i=1}^p \lambda_i^{(A)} \|P_{\perp}^{(B)} v_i^{(A)}\|_{L^2}}{2 \sum_{i=1}^p \lambda_i^{(A)}} + \frac{\sum_{i=1}^p \lambda_i^{(B)} \|P_{\perp}^{(A)} v_i^{(B)}\|_{L^2}}{2 \sum_{i=1}^p \lambda_i^{(B)}} \quad (19)$$

We denote  $P_{\perp}^{(A)}$  as the orthogonal projection matrix onto  $W_A$ , and determine the subspace using the top  $d$  eigenvectors, for specified  $\epsilon \in [0, 1]$ , such that

$$\frac{\sum_i^d \lambda_i^{(A)}}{\sum_i^p \lambda_i^{(A)}} < \epsilon$$

Scores are in the interval  $[0, 1]$ , and subspaces with complete symmetric overlap will have scores close to 1. In the case where  $W_A \subset W_B$ , we would expect  $SS_{A \rightarrow B} \approx 1$  and it may therefore be useful to consider the two terms from (19) separately. While we provide no theoretical justification for this scoring metric, it agrees with our intuition for weighted projections between the subspaces.

We summarize the two primary uses of gradient learning within the multi-task setting:

1. Prediction – Gradients provide information for the selection of features or relevant subspaces for building robust predictive models.
2. Task Structure/Similarity – Multi-task modeling traditionally presumes similarity among the tasks; we pose an alternative use of the multi-task model, which is that of *learning* task similarity. We introduce a scoring metric to assist in this point.

These two uses should not be considered as distinct, but as complementary. An optimal analysis should approach the problem iteratively to guide appropriate combinations of the data for building predictive models, and to use the discriminative information to emphasize relevant structure within the data.

## 5. Experiments

We apply the multi-task gradient learning algorithm to simulated and real data for simultaneous classification and feature discovery. We explore the effect of the regularization parameters in modulating the bias-variance trade-off (Hastie et al., 2001) and its impact on predictive performance. We also compute subspace similarity scores to aid in our interpretation of the structure we infer. For brevity and clarity, we restrict our analysis to the classification setting and only a few tasks (although the algorithm generalizes to any number of tasks).

### 5.1 Simulation

We construct two tasks containing 40 samples each (20 in class **1**, 20 in class **-1**) in a 120-dimensional space. We generate a data matrix for binary classification that contains features that are common to both tasks as well as features that are specific to each task. The matrix is initialized with background noise drawn from  $\mathbf{No}(0, .2)$ , defined as normal distribution  $\mathbf{No}(\mu, \sigma^2)$ . We then generate samples according to the following table:

1. task 1, class 1:  $\{x_i\}_{i=1}^{20}$

$$x^j \sim \mathbf{No}(2, 2), \text{ for } j = 1, \dots, 10; \quad x^j \sim \mathbf{No}(2, .5), \text{ for } j = 11, \dots, 20,$$

$$x^j \sim \mathbf{No}(2, 2), \text{ for } j = 61, \dots, 70; \quad x^j \sim \mathbf{No}(2, .5), \text{ for } j = 71, \dots, 80$$

2. task 1, class 2:  $\{x_i\}_{i=21}^{40}$

$$x^j \sim \mathbf{No}(2, 2), \text{ for } j = 91, \dots, 100; \quad x^j \sim \mathbf{No}(2, .5), \text{ for } j = 101, \dots, 110,$$

3. task 2, class 1:  $\{x_i\}_{i=41}^{60}$

$$x^j \sim \mathbf{No}(2, 2), \text{ for } j = 31, \dots, 40; \quad x^j \sim \mathbf{No}(2, .5), \text{ for } j = 41, \dots, 50,$$

$$x^j \sim \mathbf{No}(2, 2), \text{ for } j = 61, \dots, 70; \quad x^j \sim \mathbf{No}(2, .5), \text{ for } j = 71, \dots, 80$$

4. task 2, class 2:  $\{x_i\}_{i=21}^{40}$

$$x^j \sim \mathbf{No}(-2, 2), \text{ for } j = 91, \dots, 100; \quad x^j \sim \mathbf{No}(-2, .5), \text{ for } j = 101, \dots, 110$$

We run MTGL on the simulated data with variations on the regularization parameters  $(\mu, \lambda)$  and observe their effect on predicting class membership for all the samples. Recalling our definition for the multi-task function,

$$F_t = f_0 + f_t$$

we can observe the parameters' effects on predicting class membership in Figure (1b-d) for  $F_t$  (red '\*'') and  $f_0$  (blue 'o'). Consistent with our expectations, when  $\mu \gg \lambda$ , the model behaves as if it is one task and we see  $f_t \rightarrow 0$ , Figure (1c). Similarly, when  $\mu \ll \lambda$ , the model behaves as 2 independent tasks and  $f_0 \rightarrow 0$ , Figure (1d).

Using the same data matrix constructed above, we turn to the case of feature selection. Figure (2) contains the plots for the RKHS norm for the common and task specific features. Here, we clearly see large norms corresponding to significant features. We also see that the algorithm is able to differentiate features shared between tasks (Figure (2a)), and features that are specific to each task, Figure (2b,c). Finally, we observe the effect of the regularization parameters  $\mu$  and  $\lambda$  on features with high variance. When both parameters are small, we reduce the shrinkage effect on the estimated gradients, allowing features with higher variance to play a larger role in the classification function. This effect is clearly visible in Figure (3) where  $\mu$  and  $\lambda$  are both small. We note how task specific, high variance features will dominant even shared, low variance features in the *common* model, Figure (3a).

We calculate the *SSscores* for this data with  $\epsilon = .95$ . When we compare the dominant subspaces of task 1 and task 2, we obtain a score of .18. This result is at first surprising, until we recognize that although  $40/120 \approx 33\%$  of the features are overlapping, half of these are negatively correlated. The negative correlation imposes a condition of orthogonality that may not be the desired interpretation; this can be removed by taking the absolute value of the GOP matrices. Recomputing the subspace score using the absolute value produces a score of .63. Computing the subspace score for task 1 and the common subspace, we obtain a score of .82. Looking at the individual terms of the score, we see that  $SS_{\text{common} \rightarrow T_1} = .67$  and  $SS_{T_1 \rightarrow \text{common}} = .96$ . From this, we conclude that the subspace of task 1 is contained almost entirely within the common subspace. This agrees with what we would expect given our construction of the data.

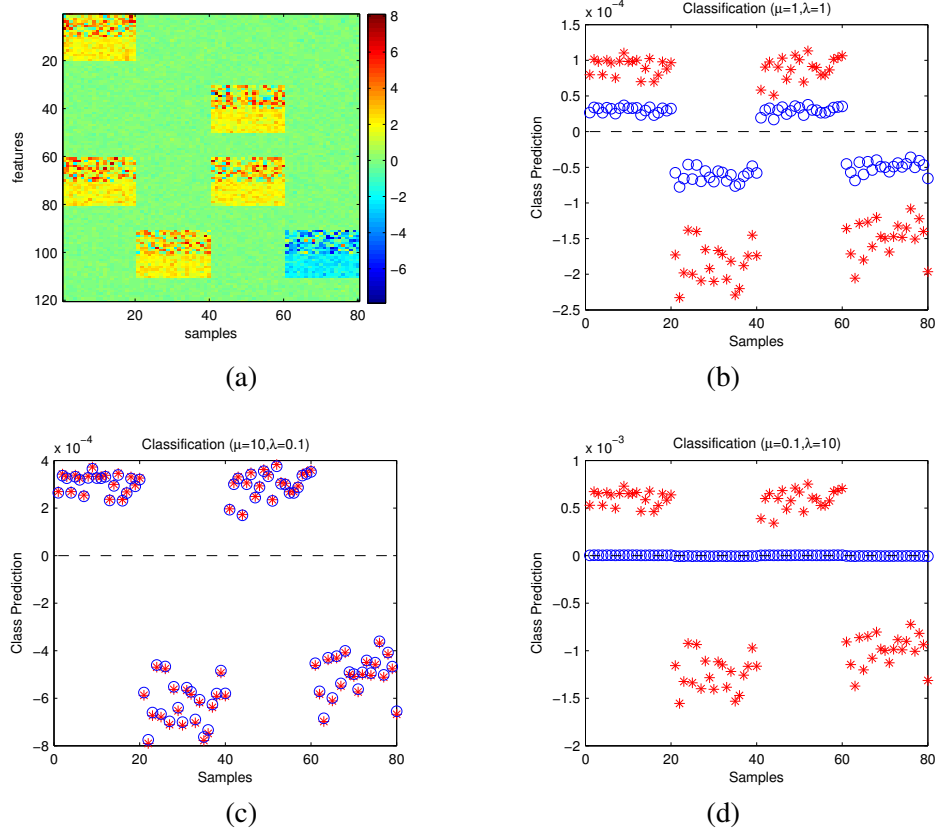


Figure 1: (a) The data matrix  $x$  where each sample corresponds to a column; samples  $1 \dots 20$  correspond to task 1, class +1; samples  $21 \dots 40$ , task 1, class -1; samples  $41 \dots 60$  task 2, class +1; samples  $61 \dots 80$  task 2, class -1, (b,c,d) Class predictions with different regularization parameters; blue 'o':  $f_0$ ; red '\*':  $F_t$ .

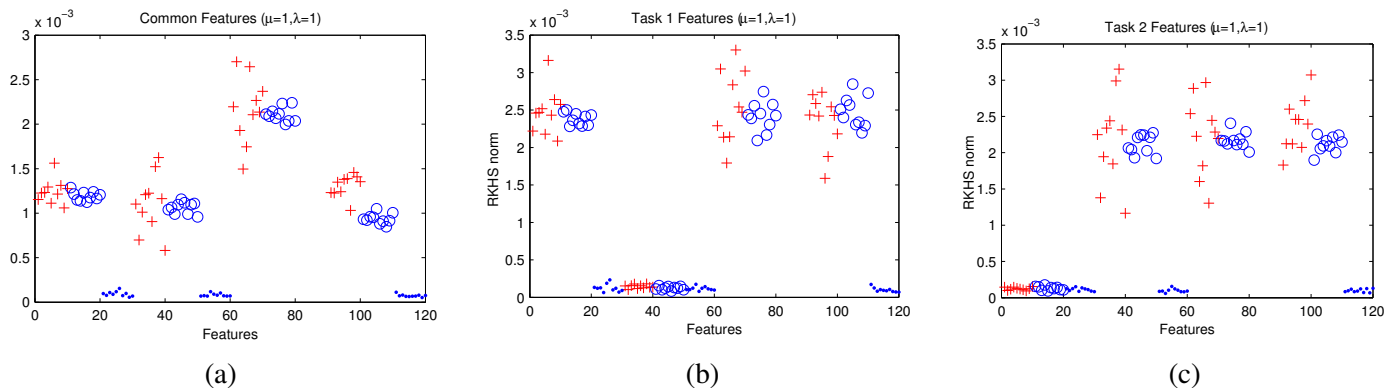


Figure 2: Feature selection,  $\mu = 1, \lambda = 1$ . Red '+' are features with high variance, blue 'o' are features with low variance. The subplots are the RKHS of (a) common features of task 1 and 2 (b) task 1 specific features (c) task 2 specific features

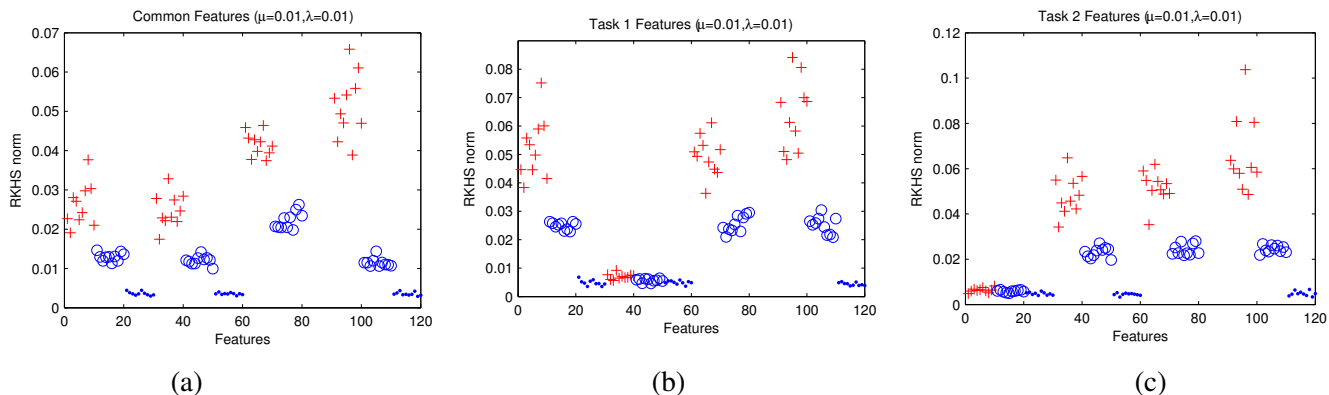


Figure 3: Feature selection,  $\mu, \lambda = .01$ . Red '+' are features with high variance, blue 'o' are features with low variance. The subplots are the RKHS of (a) common features, task 1 and 2 (b) task 1 features (c) task 2 features

## 5.2 Digits - Feature Selection

The MNIST (LeCun) digit database is a standard dataset in the machine learning community for benchmarking classification algorithms. The dataset consists of thousands of hand-written numbers (0-9) captured as 768-dimension vectors corresponding to the 28 pixel by 28 pixel image. All images have been centered and normalized. Our experiment uses the **3**, **5**, and **8** digits by considering ‘3 vs 8’ as one task, and ‘5 vs 8’ as a second task. Our motivation for choosing these digits appears natural when considering the similarities between 3 and 5: their bottom halves are identical, while their top halves are rough mirror images. The relevant features of 3 and 5 can also be considered as approximate subsets of 8, making it an interesting classification problem. As in the simulation experiments, our goal is to learn the features of 3 and 5 that jointly distinguish themselves from an 8, and simultaneously learn the features that distinguish a ‘3 vs 8’ and ‘5 vs 8’.

We build our data matrix  $X$  with a random selection of 50 3’s, 50 5’s, and 100 8’s, where  $X_i \in \mathbb{R}^{784}$  and  $i \in \{1, \dots, 200\}$ . We then run MTGL on the data with regularization parameters  $\lambda = 1$ ,  $\mu = 10$  to obtain a sparse set of features. The plots of the RKHS norm in Figure (4) shows the relevant *common* features and the task specific features. In Figure (4a), we observe how the open loop in the left lower quadrant is common to ‘3 and 5’ vs 8. In Figure (4b) (‘3 vs 8’), our method identifies the open loop in the top left and bottom left quadrants as significant, and in Figure (4c) (‘5 vs 8’), the open loops in the top right and bottom left quadrants are identified. These results agree with our intuition with respect to how the numbers differ given their canonical form.

We next test the robustness of these features for building predictive models. We rank the features by their RKHS norm and select the top  $l = \{5, 10, 15, 20\}$  features to build a binary regression model. We train the regression models using 50 randomly selected images from our 3,5, and 8 digit sets, and then test the models for correctly predicting ‘3 and 5’ vs 8, ‘3 vs 8’, and ‘5 vs 8’ on the test digit set. We iterate this procedure 100 times and obtain an average of the results. To establish a baseline for comparison, we repeat the same experiment by randomly choosing features of length  $l$ ; we also compare the results to a support vector machine (SVM) (Vapnik, 1998) model using all features. Results are displayed in Table 1. While the features obtained from the gradient learning method clearly outperform a random selection of features, we notice how the SVM using all features has the highest performance. We suspect this is a result of the high amount of interdependence among the top features obtained by the MTGL, seen as a decline in predictive performance as we add more features. To deal with this, we turn to the application of dimension reduction.

<b>Digit Classification: Top Common Features</b>				
# features	5	10	15	20
Top norm (bin-reg)	84.2%	85.8%	84.0%	82.2%
Random (bin-reg)	58.9%	67.6%	62.2%	69.8%
All (svm)	88.0%			

Table 1: Prediction with top RKHS features: ‘3 and 5’ vs ‘8’

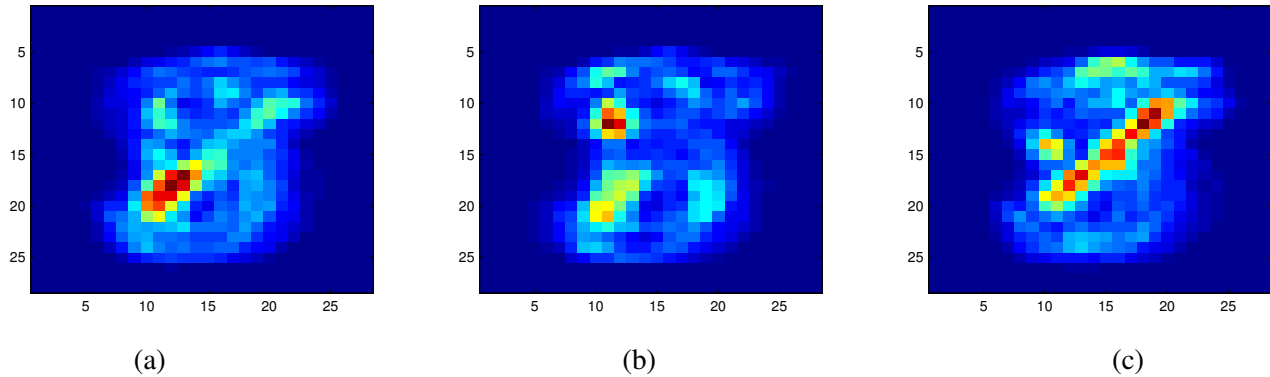


Figure 4: Plots of RKHS norm for digit image data. (a) Common features ('3 and 5' vs 8) (b) Task 1 features (3 vs 8) (c) Task 2 features (5 vs 8)

### 5.3 Digits - Dimension Reduction

By using the RKHS norm to rank features, we expect many top features will have a high amount of interdependence. A robust discriminative model will want to exclude features that are highly collinear as they do not add anything to the model. Because the GOP matrices provides an estimate of the covariance between features, we can obtain an orthogonal (independent) basis for our data through a spectral decomposition of the GOP matrix.

Using the same 3, 5, and 8 matrix  $X$ , and the GOP matrix obtained from our previous experiment, we perform a spectral decomposition of the *common* GOP matrix and use the top  $l = \{1, 2, 3, 4\}$  significant eigenvectors. We train a binary regression model with logistic loss by projecting our training data onto the reduced basis, and then predict '3 and 5' vs 8 with the test digits data. We compare these results performing the same experiment, but with projection matrices obtained from PCA and regularized LDA operating on  $X$ . Results are displayed in Table 2. We see that MTGL outperforms PCA and rLDA in finding the relevant subspace for building a predictive model. We also note that most of the discriminative power is captured in the first principal component obtained from the decomposition of the GOP matrix.

<b>Digit Classification: Dimension Reduction</b>				
# dim.	1	2	3	4
MTGL	88.2%	88.5%	88.5%	88.5%
PCA	64.0%	73.1%	77.7%	80.3%
rLDA	85.9%	–	–	–

Table 2: Prediction in reduced subspace: '3 and 5' vs '8'

## 5.4 Oncogenic Pathways

Cancer is a disease that operates principally at the level of genes. While cancer is extremely heterogeneous, it exhibits common observable characteristics (phenotypes) across all cancer types. Therefore, it is reasonable to ask *what is the common genetic basis of cancer, and how does each cancer type differ from one another?* Such a question fits nicely within the MTL framework.

Molecular biologists and cancer researchers have identified a set of genes called oncogenes that are believed to play a significant role in cancer’s early manifestations and progression. We use five cell lines from Bild et al. (2006) that have each had one of their Myc, E2F3,  $\beta$ -catenin, Ras, or Src oncogenes knocked-out (silenced). The expression data captures the downstream perturbations to the gene network as a result of this oncogenic knock-out. Our goal is to see if we can recover the oncogenic signatures specific for each cell line (task), and to see if we can infer an underlying biology common to all lines. Cells from each of these cell lines were hybridized to mRNA microarrays and the amount of mRNA was measured for all known genes. There are 7-10 samples from each cell line, which we combined with a control cell line to build a data matrix composed of 5 tasks, 95 total samples, and 20647 genes.

We run MTGL on this matrix, and create heat maps of the expression data for the top 100 common and task specific genes/features, Figure (5). We immediately notice an extreme contrast in gene regulation, where the top common genes (Figure (5a)) in the oncogenic pathways show significant down-regulation, and the top genes for E2F3 and Myc expression exhibit up-regulation, Figure (5b,e). We also observe that the high ranking genes in the Myc, E2F3, and Ras expression profiles are highly differentiated (Figure (5b,e,f)), unlike the Src and  $\beta$ -catenin, Figure (5c,d). This suggests that significant genes for the Myc, E2F3, and Ras pathways function independently from each other, while the Src and  $\beta$ -catenin genes have a mode of behavior common to all the pathways. Patterns such as these could be helpful to cancer researchers when considering drug therapies targeted for specific cancer types versus therapies that act more generally against the underlying biology of cancer.

To determine if we recovered significant genes, we compare our top 100 genes for each task with the oncogenic signatures obtained in Bild et al. (2006) for each cell line. We use the hypergeometric distribution to test for significance. Results are provided in Table 3 with the columns containing the p-values for the task specific features tested against each of the 5 oncogenic signatures. We clearly see that our top ranked genes for each task show significant enrichment with respect to their associated oncogenic signature, seen along the diagonal.

We next explore the biological significance of the *common* ranked features. We used the molecular signature database (Subramanian et al., 2005) to test for enrichment of our top genes within the database of curated gene sets. We again use the hypergeometric, and obtain as our top 2 gene sets the ET743\_HELA\_UP and OSAWA\_TNFA\_HEPATOCYTE\_UP pathways. Not surprisingly, both gene sets are associated with apoptosis (programmed cell death), a principal characteristic of most cancer. Also in our top 5 was CHEN\_HOXA5\_TARGETS\_UP, a known pathway involved with tumorigenesis.

We compute the SS scores between all combinations of the subspaces we obtain for each task and the common. Results are displayed in Table 4, where the values above the diagonal represent the score by projecting the subspace of the row into the column, and values below the diagonal represent the reverse. For example, row 2, column 3 is the computation of  $SS_{myc \rightarrow src} = .41$ , and row 3, column 2 is  $SS_{src \rightarrow myc} = .77$ . Using these scores, we would conclude that the subspace we

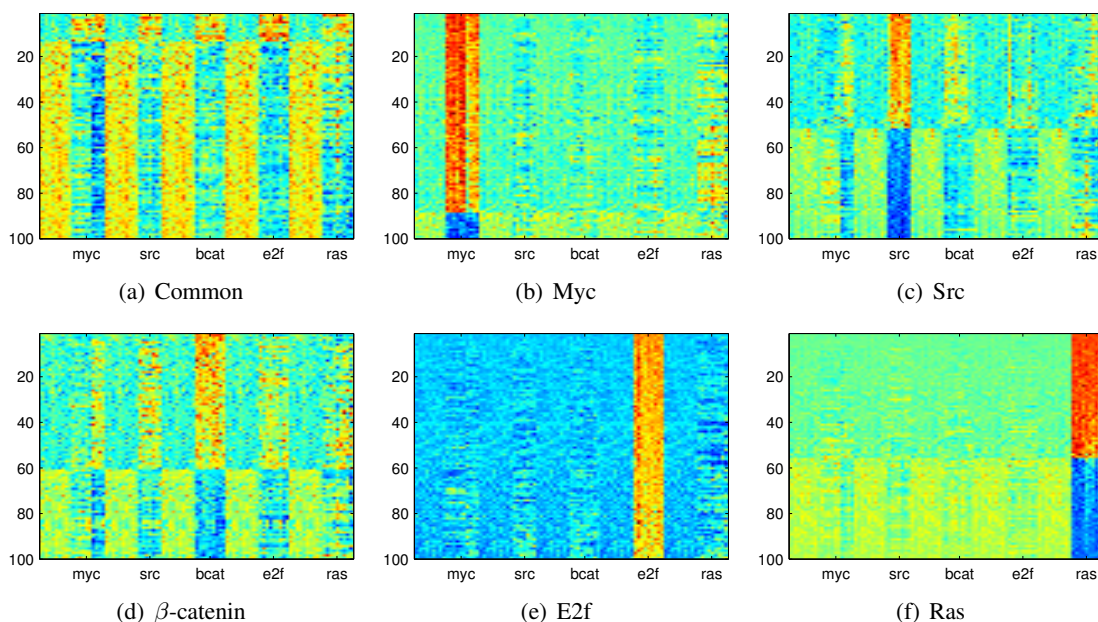


Figure 5: Top 100 genes for the oncogenic pathways

Signature / Top	Myc	Src	$\beta$ -catenin	E2F3	Ras
Myc	<b>4e-89</b> (57)	.74	.28	.36	.001
Src	.35	<b>1e-11</b> (9)	.008	.30	.002
$\beta$ -catenin	.19	.22	<b>2e-16</b> (13)	.22	.008
E2F3	.37	.74	.65	<b>2e-137</b> (79)	.27
Ras	.19	.23	.28	.36	<b>7e-61</b> (46)

Table 3: P-values and enrichment scores (gene overlaps) of top 100 genes per oncogenic signature

obtain for Myc is in the span of the Src subspace, although the opposite is less true. With this table, we also observe that Ras appears to be the least similar to all the other pathways. Ras and E2F3 show the least similarity with an average score of .35, and  $\beta$ -catenin and Src exhibit the most with an average score of .71.

As a final exercise, we construct graphical models of gene networks to observe the dependence structure of the variables. We also hope to observe larger patterns of co-expression. Using  $\hat{\Gamma}$  as the covariance matrix of a multivariate Gaussian distribution, the theory of Gauss-Markov graphs (Speed and Kiiveri, 1986) states that the precision matrix  $\hat{\Gamma}^{-1}$  is the conditional independence matrix between all variables. We use this result to build undirected graphical models, where  $\hat{\Gamma}_{ij}^{-1}$  is a measure of the dependence between variables  $i$  and  $j$  conditioned on all other variables.

	Common	myc	src	$\beta$ -catenin	e2f3	ras
Common	–	.73	.72	.75	.73	.59
myc	.66	–	.41	.51	.44	.44
src	.77	.77	–	.74	.72	.62
$\beta$ -catenin	.80	.62	.68	–	.64	.53
e2f3	.73	.48	.48	.59	–	.33
ras	.59	.46	.50	.43	.38	–

Table 4: Subspace Similarity (SS) Scores

We look at two graphs specifically, derived from the common and Ras GOP matrices  $\hat{\Gamma}_0$  and  $\hat{\Gamma}_{ras}$ . The graphs in Figures (6,7) reconfirm many of our previous findings. In Figure (6), we see two large clusters dominated by the the  $\beta$ -catenin, Myc, and Src nodes, with a strong presence of E2F3 in one cluster but not the other. We also notice the absence of Ras nodes around these two clusters, consistent with our earlier subspace scoring. The Ras graph depicted in Figure (7) shows two strong Ras (red) clusters. We observe the presence of Src (light-blue) nodes around the Ras clusters, suggesting some dependency between Ras and Src related genes. This is again consistent with our subspace similarity scoring, as Src scores highest with Ras among all the other cell lines.

## 6. Discussion

The MTGL method we introduce in this paper is based on Tikhonov regularization with an RKHS norm. This allows for the estimates to be effective in high-dimensional problems. The method has three regularization parameters: parameters on the common and task specific norms  $(\mu, \lambda)$ , and the bandwidth parameter  $(s)$ . While we have provided an intuition of the effect of these parameters we have not discussed criteria for setting the parameters. In the case of classification or regression the accuracy of the model assessed by cross-validation or generalized approximate cross-validation (GACV) can be used to set the parameters. In the MTGL setting the accuracy of the model as well as the structure of the data are both important, therefore the use of cross-validation or other predictive measures may not be optimal. In some cases, it may be more desirable to emphasize task distinctiveness rather than task similarity. Likewise, when considering the bandwidth parameter, we must balance our choice of maximizing the classification function versus discovery of salient features. In choosing larger windows, the classification function is smoother and consequently less emphasis is given to local structure. Our analysis finds that larger windows will improve performance in predictive models with few variables.

Parameterization choices need not reflect *a priori* knowledge of task similarity; another consideration is the *a posteriori* analysis. This suggests the development of a coherent Bayesian framework for MTGL to allow for a posterior distribution on the regularization parameters and generalize the types of norms in the regularization terms to a broader class of priors. For MTL a Bayesian

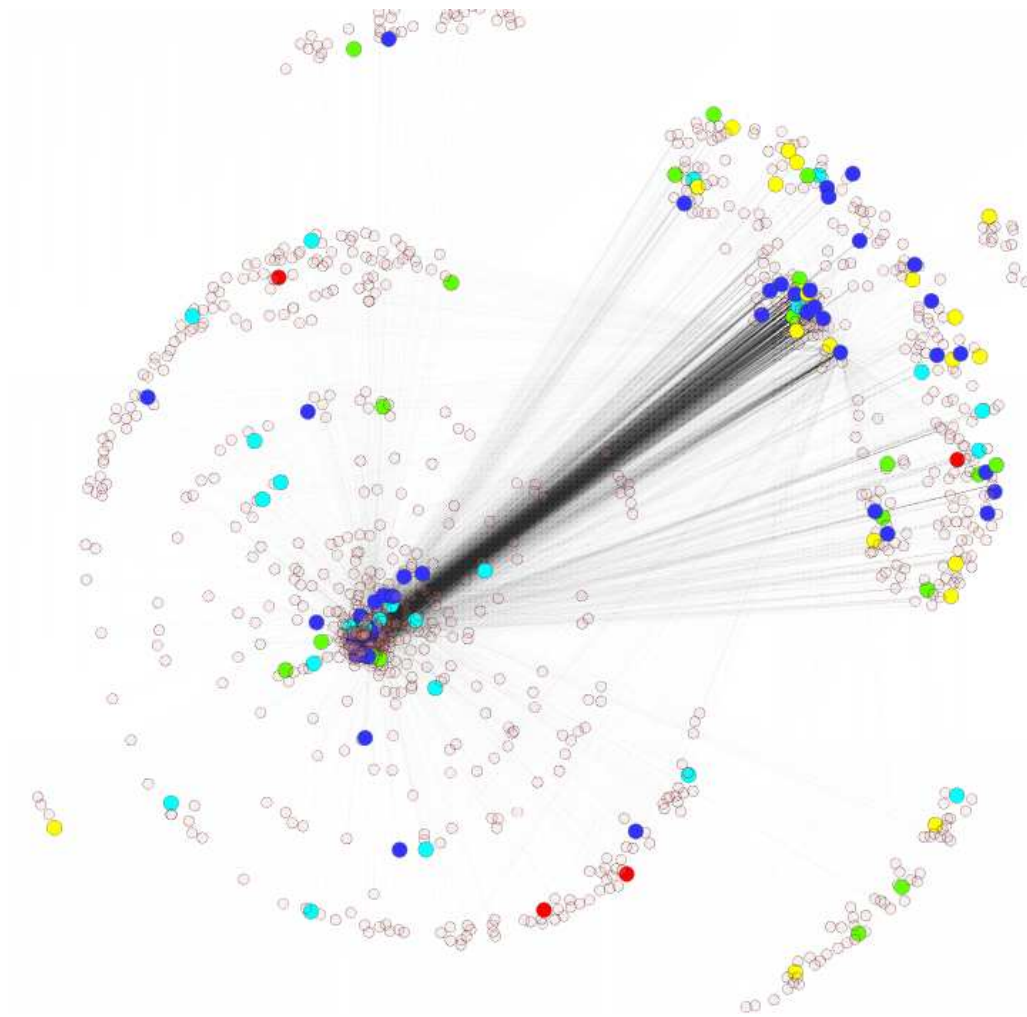


Figure 6: Graphical Model of Common Oncogenic Pathway. Colored nodes represent corresponding cell line signatures from Bild et al. (2006).  $\beta$ -catenin in dark-blue; E2F3 in yellow; Myc in green; Ras in red; Src in light-blue

model was explored in Xue et al. (2007). Integrating the ideas from Xue et al. (2007) with the non-parametric Bayesian kernel models developed in Liang et al. (2007) should provide a modeling framework for a Bayesian analysis and estimates of uncertainty.

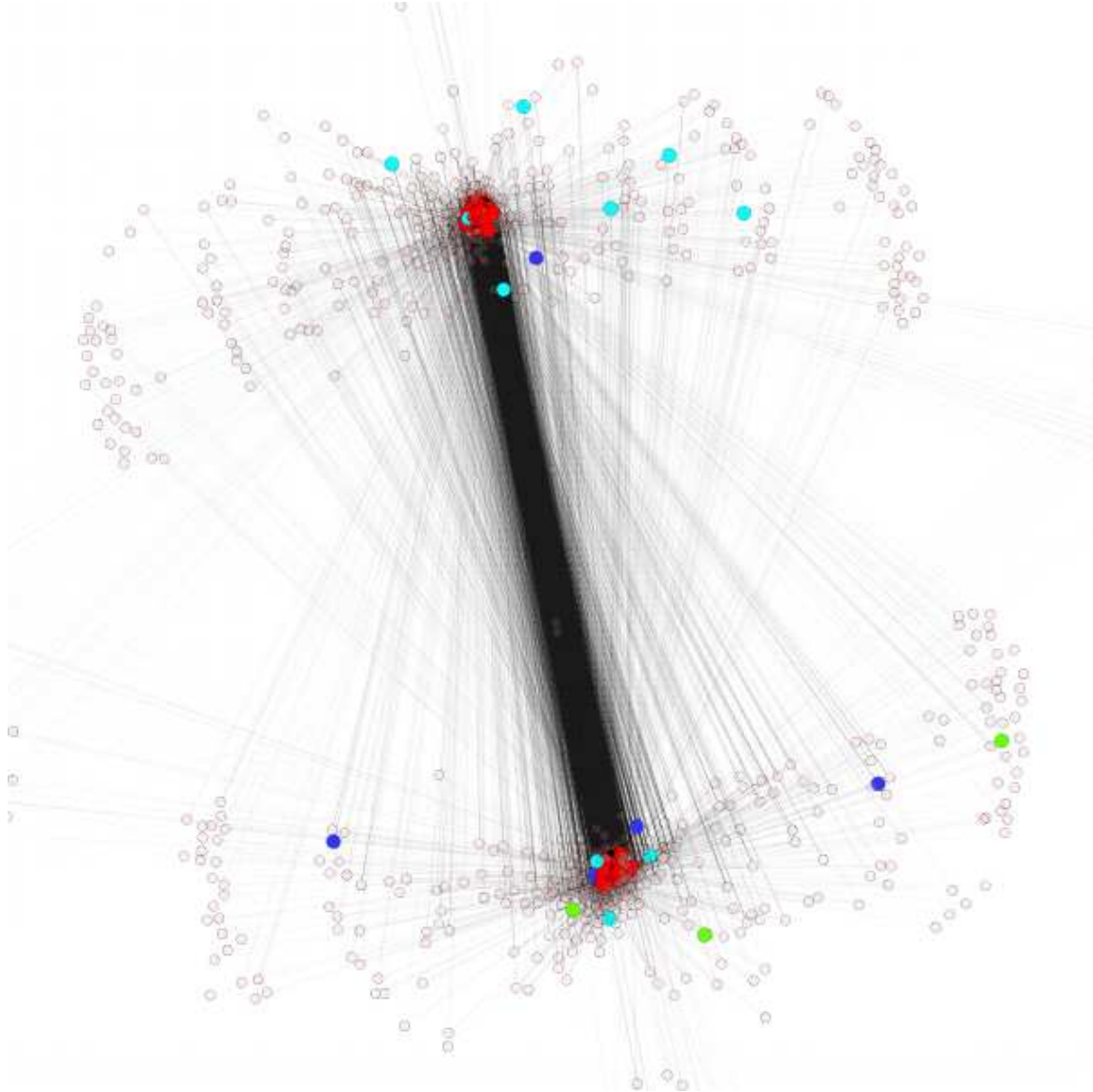


Figure 7: Graphical Model of Ras Oncogenic Pathway. Colored nodes represent corresponding cell line signatures from Bild et al. (2006).  $\beta$ -catenin in dark-blue; E2F3 in yellow; Myc in green; Ras in red; Src in light-blue

## References

- R. Ando and T. Zhang. A framework for learning predictive structure from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.
- A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *NIPS 20*, 2006.
- M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *NIPS*, 14, 2002.
- S. Ben-David and R. Schuller. Exploiting task relatedness for multiple task learning. In *Proc. of Computational Learning Theory (COLT)*, 2003.
- A. Bild, G. Yao, J. Chang, Q. Wang, A. Potti, D. Chasse, M. Joshi, D. Harpole, J. Lancaster, A. Berchuck, J. Olson, J. Marks, H. Dressman, M. West, and J. Nevins. Oncogenic pathway signatures in human cancers as a guide to targeted therapies. *Nature*, 439:353–357, 2006.
- R. Caruana. Multi-task learning. *Machine Learning*, 28:41–75, 1997.
- R.D. Cook and S. Weisberg. Discussion of "sliced inverse regression for dimension reduction". *J. Amer. Statist. Assoc.*, 86:328–332, 1991.
- T. Evgeniou and M. Pontil. Regularized multi-task learning. In *Proc. Conference on Knowledge Discovery and Data Mining*, 2004.
- T. Evgeniou, C. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.
- R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7: 179–188, 1936.
- I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.
- T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.
- H. Hotelling. Analysis of a complex of statistical variables in principal components. *Journal of Educational Psychology*, 24:417–441, 1933.
- T. Jebara. Multi-task feature and kernel selection for svms. In *Proc. of ICML*, 2004.
- Y. LeCun. Mnist database. URL <http://yann.lecun.com/exdb/mnist>.
- K.C. Li. Sliced inverse regression for dimension reduction. *J. Amer. Statist. Assoc.*, 86:316–342, 1991.
- F. Liang, K. Mao, S. Mukherjee, M. Liao, and M. West. Non-parametric Bayesian kernel models. 2007.
- M. Maggioni, S. Mukherjee, Q. Wu, and J. Guinney. Learning gradients: predictive models that reflect geometry and dependencies. 2007.

- S. Mika, B. Schölkopf, A.J. Smola, K.R. Müller, M. Scholz, and G. Rätsch. Kernel pca and denoising in feature spaces. In *NIPS*, volume 11, 1999.
- S. Mukherjee and Q. Wu. Estimation of gradients and coordinate covariation in classification. *Journal of Machine Learning Research*, pages 519–549, 2006.
- S. Mukherjee and D. Zhou. Learning coordinate covariances via gradients. *Journal of Machine Learning Research*, pages 519–549, 2006.
- S. Mukherjee, Q. Wu, and D. Zhou. Learning gradients and feature selection on manifolds. 2007.
- G. Obozinski, B. Taskar, and M. Jordan. Multi-task feature selection. Technical report, Dept. of Statistics, University of California, Berkeley, 2006.
- S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.
- T. Speed and H. Kiiveri. Gaussian markov distributions over finite graphs. *Annals of Statistics*, 14: 138–150, 1986.
- A. Subramanian, P. Tamayo, V. Mootha, S. Mukherjee, B. Ebert, M. Gillette, A. Paulovich, S. Pomeroy, T. Golub, E. Lander, and P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *PNAS*, 102(43):15545–15550, 2005.
- J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.
- V.N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.
- G. Wahba. *Splines Models for Observational Data*. Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia, 1990.
- Y. Xia, H. Tong, W. Li, and L-X. Zhu. An adaptive estimation of dimension reduction space. *J. Roy. Statist. Soc. Ser. B*, 64(3):363–410, 2002.
- Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8:35–63, 2007.