Probabilistic Joint Feature Selection for Multi-task Learning

Tao Xiong* Jinbo Bi† Bharat Rao‡ Vladimir Cherkassky§

Abstract

We study the joint feature selection problem when learning multiple related classification or regression tasks. By imposing an automatic relevance determination prior on the hypothesis classes associated with each of the tasks and regularizing the variance of the hypothesis parameters, similar feature patterns across different tasks are encouraged and features that are relevant to all (or most) of the tasks are identified. Our analysis shows that the proposed probabilistic framework can be seen as a generalization of previous result from adaptive ridge regression to the multi-task learning setting. We provide a detailed description of the proposed algorithms for simultaneous model construction and justify the proposed algorithms in several aspects. Our experimental results show that this approach outperforms a regularized multi-task learning approach and the traditional methods where individual tasks are solved independently on synthetic data and the real-world data sets for lung cancer prognosis.

1 Introduction

Multi-task learning, other than traditional single task learning which treats each task separately and independently, is a machine learning method that tries to estimate models for several tasks in a joint manner. Facing a sparse data set, multi-task learning could be beneficial by compensating for small sample size by using additional samples from related tasks. Therefore, multi-task learning can be seen as a way to reduce the variance of the model estimate by introducing a little bias. From the hierarchical Bayesian viewpoint, multi-task learning is essentially trying to learn a good prior over all tasks to capture task dependencies. Previous empirical work has shown that using data from related tasks does improve prediction performance [4, 8, 5].

Multi-task learning becomes necessary and can potentially enhance performance when the tasks are similar enough to make joint learning beneficial whereas they are not identical that learning a specific model for each task leads to better predictive capacity. Although almost all existing multi-task learning methods assume some relatedness among tasks, the definition of relatedness varies. In previous work [8], task relationship is modeled through the assumption that noise for different regression estimates are correlated. Another work [1] considers learning multiple tasks in a semi-supervised setting and assumes a common hidden structure for all related tasks. One of the natural ways to capture the task relatedness is through hierarchical Bayesian models [12, 27]. Typically, similarity among tasks is represented by a probabilistic model with unknown parameters. The parameters of the probabilistic model are determined to reflect the relations between tasks and facilitate the model parameter estimation of each individual task. Recent work [26] also investigates nonparametric probabilistic model to learn common prior on tasks, which gives more flexibility of the model relatedness but demands greater computational complexity.

In this paper, we model the across-task relatedness as sharing a common subset of features. The goal is to identify all relevant features that are informative to any of these tasks. From the regularization standpoint, it can be regarded as controlling the overall model complexity of the multi-task learning formulation by eliminating redundant and irrelevant dimensions. Feature selection has long become an important problem in statistics [23, 11] and machine learning [10, 13]. Many machine learning problems can benefit from feature selection in several aspects, such as obtaining better predictive power, improving computational efficiency, providing better interpretability and so on. Very little work has been done on selecting features for multiple related tasks. Jebara proposed in [14] a framework based on Maximum Entropy Discrimination for joint feature selection. Obozinski et cl. [20] proposed to couple feature selection across tasks by applying a joint regularization of the model parameters. Similar work following the regularization principle has also appeared in [29].

We consider the joint feature selection problem in a probabilistic framework based on which a regularized learning formulation can be derived. By imposing an automatic relevance determination prior [17] on the hypothesis classes associated with each of the multiple

*Department of Electrical and Computer Engineering, University of Minnesota. Email: *txiong@ece.umn.edu*. The work was conducted when Tao Xiong was with Siemens Medical Solutions.

†The Computer Aided Diagnosis and Therapy Group, Siemens Medical Solutions. Email: *jinbo.bi@siemens.com*

‡The Computer Aided Diagnosis and Therapy Group, Siemens Medical Solutions. Email: *bharat.rao@siemens.com*

§Department of Electrical and Computer Engineering, University of Minnesota. Email: *cherkass@ece.umn.edu*

tasks and regularizing the variance of the model parameters, similar feature patterns across different tasks are encouraged and features that are relevant to all (or most) of the tasks are identified. The proposed approach of regularizing parameter variance can be seen as a generalization of the idea used in adaptive ridge regression (AdR) [6]. Our analysis shows an interesting connection between our approach and the approach proposed in [20]. More specifically, we show that the model estimated by our approach can be at least as sparse as those from [20]. We determine the model parameters by an alternating optimization algorithm which contains two steps at each iteration. The first step constructs models for each task with a fixed common prior parameter, and the second step learns the common prior parameter with fixed model parameters for individual tasks. Efficient algorithms for solving the sub-problems presented at each step are also discussed. Our experimental results show that this approach outperforms independent l_1 or l_2-based regularized learning models and also the regularized multi-task learning approach [5] on several synthetic and real datasets.

2 Feature Selection for Single Task Learning (STL)

Given a training set $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_i, y_i), \ldots, (\mathbf{x}_n, y_n)\}$ where $\mathbf{x} \in X^d, y \in \{-1, 1\}$ for classification and $y \in R$ for regression, we are interested in learning a prediction function $y = f(\mathbf{x})$ that can be used to predict on future data. We focus on feature or variable selection in the original feature space and limit our discussion to linear prediction functions in the form of $f(\mathbf{x}) = \beta^{\mathbf{T}}\mathbf{x}$.

For the ease of discussion, we use the regression setting to discuss the derivation of single task learning formulations, but the discussion is generally applicable to classification tasks as well. We assume that the targets y in the regression problem are generated by corrupting f with additive Gaussian noise, i.e.

$$p(y|\beta, \mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-(y - \beta^{\mathbf{T}}\mathbf{x})^2/(2\sigma^2)). \tag{2.1}$$

In the Bayesian formalism, a prior over the parameters β needs to be specified to express our beliefs about the parameters before training data is given. Usually a zero mean Gaussian prior with covariance Σ_β is used on the weights, $\beta \propto \mathcal{N}(0, \Sigma_\beta)$. The common choice of a spherical Gaussian prior with $\Sigma_\beta \propto I$ leads to the well known ridge regression models, where I denotes identity matrix of proper dimension.

For the purpose of feature selection, unlike the standard Gaussian prior as used in ridge regression, the key concept is the use of automatic relevance determination (ARD) prior of the following form:

$$\begin{aligned} p(\beta|\mu) &= \Pi_j \mathcal{N}(0, (2\mu_j)^{-1}) \\ &= \Pi_j \frac{1}{\sqrt{\pi\mu_j^{-1}}} \exp(-\mu_j\beta_j^2). \end{aligned} \tag{2.2}$$

Given the likelihood (2.1) and prior (2.2), it can be shown that the maximum-a-posterior (MAP) estimate of β minimizes the following quadratic form:

$$J(\beta) = \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \sum_j \nu_j\beta_j^2, \tag{2.3}$$

where we have used $\nu_j = \sigma^2\mu_j$ for the sake of simplicity.

Different inference strategies exist in the literature to learn the prior parameters for the given ARD prior:

- In Relevance Vector Machine (RVM) [24, 25], an analytical expression for the marginal likelihood $p(y|\mu, \sigma)$ is obtained by integrating out the expansion coefficients that correspond to the variable (β) in our notation. Given the current estimates of β, the most probable μ_{mp} is obtained explicitly by maximizing $p(y|\mu, \sigma)$. These values are used in order to get a new estimate for β. The two steps are iterated until its convergence. During this iterated process, it turns out that some parameters μ_j approach infinity, meaning that the variance of the corresponding prior $p(\beta_j|\mu_j)$ becomes zero. Since expansion coefficients also have zero mean, consequently these coefficients β_j's shrink to zero and the corresponding features are then not selected in the model.

- Notice that RVM is not a pure Bayesian approach. To achieve sparsity in a true Bayesian setting, a common choice of the prior for ν_j is a one-parameter exponential distribution [9]:

  $$p(\nu_j|\gamma) = \frac{\gamma}{2} exp(-\frac{\gamma}{2}\nu_j),$$

  with $\gamma > 0$. Integrating out ν_j yields a Laplace prior distribution:

  $$p(\beta_j|\gamma) = \frac{\gamma}{2} exp(-\sqrt{\gamma}|\beta_j|).$$

  Together with the Gaussian likelihood (2.1), this marginalization leads to the l_1 regularized functional in log-space:

  $$J^{LASSO}(\beta) = \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \sqrt{\gamma}\|\beta\|_1, \tag{2.4}$$

  which is the functional that the Least Absolute Shrinkage and Selection Operator (LASSO) minimizes [23]. It is well known that for a properly chosen regularization parameter, LASSO produces sparse solutions [11].

- Adaptive Ridge (AdR) regression [6] uses a different strategy. It directly constrains the variance of the coefficients β in the prior distribution by optimizing (2.3) with the following constraint:

$$\frac{1}{d}\sum_{i=1}^{d}\frac{1}{\nu_i}=\gamma, \tag{2.5}$$

where γ is a constant. The constraint (2.5) essentially requires that the expected variance of weight components β_j is proportional to γ, thus eliminating the possibility that all of them grow too large simultaneously. By setting γ to a relatively small value, it is expected that some components of β become small so the formulation is in favor of sparse β estimates. Furthermore, the AdR formulation has been proved to be equivalent to LASSO (2.4) as stated in [6, 7].

3 Joint Feature Selection for Multi-task Learning (MTL)

Strategies in Section 2 can be generalized to effectively select features in the multi-task learning setting. Assume we have T datasets, each corresponding to one of the T tasks. Each dataset contains the feature vectors $\mathbf{x}_{t,i}\in R^d$ and corresponding binary labels $y_{t,i}\in\{-1,1\}$ for classification problem, or real targets $y_{t,i}\in R$ for regression problem, for $i=1,\ldots,n_t$ data points in the t-th task, $t=1,\ldots,T$. Our goal is to construct T specific predictive models in the form of $\beta_t^T x,\ t=1,\cdots,T$. We impose the same ARD prior as in (2.2) on the parameter β across different tasks, i.e.,

$$\begin{aligned} p(\beta_t|\nu_j) &= \Pi_j N(0,(2\nu_j)^{-1}) \\ &= \Pi_j \tfrac{1}{\sqrt{\pi\nu_j^{-1}}}\exp(-\nu_j\beta_{tj}^2), \\ & \qquad t=1,\ldots,T. \end{aligned} \tag{3.6}$$

where parameters ν_j are the same for all tasks to represent the across-task similarity and to be determined. The joint feature selection can then be achieved by requiring the similar constraint (2.5) to be applied to the variance of β. The ending effect of using constraint (2.5) in the multi-task learning framework is not only important features are selected, but also significant features with respect to multiple tasks will be weighted proportionally according to ν.

3.1 Sparse Multi-task Classification

In this section, we study the joint feature selection for solving multiple classification problems. We construct separate logistic regression models for each task while identifying the features that are informative across all tasks. Logistic regression is a widely used method for binary classification problem. Its model is given by

$$p(y|\beta,\mathbf{x})=\sigma(\beta^{\mathbf{T}}\mathbf{x})=\frac{1}{1+exp(-y\beta^{\mathbf{T}}\mathbf{x})}.$$

The posterior distribution for all model parameters $B=[\beta_1,\ldots,\beta_T]$ with the logistic link on data (X,y) is

$$p(B|X,y)=\Pi_{t=1}^{T}\{(\Pi_{i=1}^{n_t}\tfrac{1}{1+\exp(-\beta_t^T x_i y_i)})p(\beta_t)\}. \tag{3.7}$$

where (X,y) contains T sets of pairs $\{x_i^t,y_i^t\},\ i=1,\cdots,n_t,\ t=1,\cdots,T$, and $p(\beta_t)$ specifies the prior on β_t as given in (3.6).

For numerical convenience, it is common to employ the log posterior instead of directly using the posterior (3.7).

$$\begin{aligned} l(B|X,y) &= \ln p(B|X,y) \\ &= -\sum_{t=1}^{T}\{\sum_{i=1}^{n_t}\ln(1+\exp(-\beta_t^T x_i y_i)) \\ &\quad +\sum_{j=1}^{d}(\ln\sqrt{\nu_j^{-1}\pi}+\nu_j\beta_{tj}^2)\}. \end{aligned}$$

The MAP estimate of B is then given by solving

$$\arg\max l(B|X,y)_B=\arg\min -l(B|X,y)_B.$$

By imposing the constraint on ν as in (2.5), we formulate the multi-task feature selection approach for classification as the following constrained optimization problem:

$$\begin{aligned} & min_{\beta_1,\ldots,\beta_T,\nu}\ \sum_{t=1}^{T}\{\sum_{i=1}^{n_t}\ln(1+\exp(-\beta_t^T x_i y_i))+\sum_{j=1}^{d}\nu_j\beta_{tj}^2\} \\ & \text{s.t.} \quad \frac{1}{d}\sum_{j=1}^{d}\frac{1}{\nu_j}=\gamma, \quad \nu_i>0. \end{aligned} \tag{3.8}$$

To obtain sparse classifiers, it is desired to have some $\nu_j\to+\infty$, which, however, brings up numerical instability. To avoid this kind of the divergent solutions and achieve numerical stability, we employ the change of variables as follows:

$$\begin{aligned} \alpha_{tj} &= \sqrt{\nu_j\gamma}\beta_{tj}, \quad t=1,\ldots,T, \\ c_j &= \sqrt{(\nu_j\gamma)^{-1}}, \quad j=1,\ldots,d. \end{aligned} \tag{3.9}$$

Correspondingly, $\beta_{tj}=\alpha_{tj}/\sqrt{\nu_j\gamma}$ and $\frac{1}{d}\sum_j c_j^2=\frac{1}{d\gamma}\sum_j\frac{1}{\nu_j}=1$.

Now the optimization problem becomes

$$\min_{\alpha_1,\ldots,\alpha_T,\mathbf{c}}\sum_{t=1}^{T}\{\sum_{i=1}^{n_t}\ln(1+\exp(-y_i\sum_{j=1}^{d}c_j\alpha_{tj}x_{ij})) \tag{3.10}$$

$$+\gamma \sum_{j=1}^{d} \alpha_{tj}^2\}$$

$$\text{s.t.} \qquad \frac{1}{d}\sum_{j=1}^{d} c_j^2 = 1, \quad c_j \geq 0.$$

The final model becomes $\beta_t^T \mathbf{x} = \sum_j c_j \alpha_{tj} x_j$. To effectively solve (3.10), we devise an alternating optimization algorithm, which is, in spirit, similar to the Expectation-Maximization approach. At iteration s, the "E" step estimates the optimal Bayes prior parameters $\mathbf{c}^s$ based on β^{s-1} that is obtained at last iteration. Then the "M" step estimates a new β^s by maximizing the posterior based on $\mathbf{c}^s$.

In the "E" step where $\mathbf{c}$ is to be optimized, the quadratic equality constraint $\frac{1}{d}\sum_j c_j^2 = 1$ makes the sub-optimization problem for solving $\mathbf{c}$ non-convex. To form a convex program in the "E" step, we propose to apply a relaxation scheme $\sum_j c_j^2 \leq (\sum_j c_j)^2$. More specifically, we replace the 2-norm equality constraint with the following 1-norm equality constraint.

$$\sum_{j}^{d} c_j = \sqrt{d}, \quad c_j \geq 0. \tag{3.11}$$

The relaxation not only makes the problem convex, but also makes the estimates of $\mathbf{c}$ more sparse. Note that sparse $\mathbf{c}$ will indicate sparse $\beta_t, t = 1, \ldots, T$. The alternating algorithm is described in details in Algorithm 1.

Algorithm 1: Joint Feature Selection for Classification

- Initialize $c_i = 1, i = 1, \ldots, d$.
- Iterate until convergence
  - Based on current $\mathbf{c}$, for $t = 1, \ldots, T$, solve the following problem for optimal α_t

$$\min_{\alpha_t} \sum_{i=1}^{n_t} \ln(1+\exp(-\sum_{j=1}^{d} c_j \alpha_{tj} x_{ij} y_i)) + \gamma \sum_{j=1}^{d} \alpha_{tj}^2; \tag{3.12}$$

  - Fix α, solve the following program for $\mathbf{c}$

$$\min_{\mathbf{c}} \sum_{t=1}^{T}\sum_{i=1}^{n_t} \ln(1+\exp(-\sum_{j=1}^{d} c_j \alpha_{tj} x_{ij} y_i)) \tag{3.13}$$

$$\text{s.t.} \quad \sum_{j=1}^{d} c_j = \sqrt{d}, \quad c_j \geq 0. \tag{3.14}$$

3.2 Sparse Multi-task Ridge Regression The proposed framework for joint feature selection for multi-task learning is not restricted to classification applications. A sparse multi-task regression formulation can be similarly derived. Specifically, if the likelihood (2.1) and prior (2.2) as discussed in Section 2 are used, we can derive a ridge regression-like multi-task learning formulation as follows:

$$min_{\beta_1,\ldots,\beta_T,\nu} \sum_{t=1}^{T}\{\sum_{i=1}^{n_t}(\beta_t^T x_i - y_i)^2 + \sum_{j=1}^{d} \nu_j \beta_{tj}^2\}$$

$$\text{s.t.} \qquad \frac{1}{d}\sum_{j=1}^{d} \frac{1}{\nu_j} = \gamma, \quad \nu_j > 0.$$

Applying the same "change of variables" (3.9) and "relaxation" (3.11) yields the following optimization problem for multi-task regression feature selection.

$$min_{\alpha_1,\ldots,\alpha_T,\mathbf{c}} \sum_{t=1}^{T}\{\sum_{i=1}^{n_t}(\sum_{j} c_j \alpha_{tj} x_{ij} - y_i)^2 + \sum_{j=1}^{d} \alpha_{tj}^2\}$$

$$\text{s.t.} \qquad \sum_{j=1}^{d} c_j = \sqrt{d}, \quad c_j \geq 0.$$

The corresponding alternating algorithm is presented in Algorithm 2.

Algorithm 2: Joint Feature Selection for Regression

- Initialize $c_i = 1, i = 1, \ldots, d$.
- Iterate until convergence
  - Based on current $\mathbf{c}$, for $t = 1, \ldots, T$

$$\min_{\alpha_t} \sum_{i=1}^{n_t}(\sum_{j=1}^{d} c_j \alpha_{tj} x_{ij} - y_i)^2) + \gamma \sum_{j=1}^{d} \alpha_{tj}^2 \tag{3.15}$$

  - Fix current estimate of α

$$\min_{\mathbf{c}} \sum_{t=1}^{T}\{\sum_{i=1}^{n_t}(\sum_{j=1}^{d} c_j \alpha_{tj} x_{ij} - y_i)^2)\} \tag{3.16}$$

$$\text{s.t.} \quad \sum_{j=1}^{d} c_j = \sqrt{d}, \quad c_j \geq 0. \tag{3.17}$$

3.3 Efficient Algorithms for Optimizing Sub-problems Notice that both Algorithms 1 and 2 consist of 2 steps at every iteration. In the first step, the original optimization problem, when c is fixed, can be

decoupled to optimize individual α_t. Note that the objective function in (3.12) is a negated log-posterior for a logistic regression model with a Gaussian prior. Consequently, the decoupled individual optimization problems as stated in (3.12) and (3.15) are nothing but a regular logistic regression and ridge regression, respectively. Thus, sufficiently efficient algorithms have been explored [28, 11, 18, 15]. Furthermore, both problems (3.12) and (3.15) are strongly convex, and hence any local minimizer is also a global minimizer. A wide variety of optimization algorithms are applicable to convex programs. For classification, many algorithms have been proposed for maximum a posterior (MAP) logistic regression [28, 11, 18, 15]. In this paper, we choose to base our implementation of logistic regression on conjugate gradient (CG) [18, 19]. For ridge regression, an analytic solution can be easily derived for problem (3.15) as

$$\alpha_t = (diag(\mathbf{c})X_t^T X_t diag(\mathbf{c}) + \gamma I)^{-1} diag(\mathbf{c}) X_t^T \mathbf{y}_t.$$

In the second step, an efficient Newton+Armijo algorithm is developed to solve the relaxed optimization problems in (3.13) and (3.16). We illustrate the algorithmic derivation for problem (3.13) in the classification setting. Similar derivation applies to the regression setting.

First, consider using Iteratively Reweighted Least Squares (IRLS) to minimize the objective function (3.13). IRLS is an efficient implementation of Newton's method. Define $l(\mathbf{c})$ the objective function in (3.13). The gradient of this objective is

$$g(c) = \nabla_c l(c) = -\sum_{t=1}^{T}\sum_{i=1}^{n_t}(1-\sigma(y_i c^T \hat{x}_i))y_i \hat{x}_i,$$

where $\hat{x}_i = diag(\alpha)x_i$ and we have omitted subscript t for simplicity. And $\sigma(x) = \frac{1}{1+exp(-x)}$.

The Hessian of the objective is

$$\begin{aligned} H(c) = \frac{d^2 l(c)}{dcdc^T} &= \sum_{t=1}^{T}\sum_{i=1}^{n_t}\sigma(c^T\hat{x}_i)(1-\sigma(c^T\hat{x}_i))x_i x_i^T \\ &= \sum_{t=1}^{T} X_t A_t X_t^T, \end{aligned}$$

where $\hat{x}$'s are columns of X and $a_{ii} = \sigma(c^T\hat{x}_i)(1-\sigma(c^T\hat{x}_i))$.

Then a Newton's step is to compute c_{new} as the minimizer of the following equality constrained convex optimization problem:

$$\begin{aligned} \arg\min_c \quad & g(c^k)^T(c-c^k) \\ & +\frac{1}{2}(c-c^k)^T H(c^k)(c-c^k) \end{aligned} \tag{3.18}$$

$$\text{s.t.} \quad \sum_i^d c_i = \sqrt{d}, \quad c_i \geq 0,$$

where c^k is the optimal c after k^{th} iteration of the Newton+Armijo algorithm.

After finding c_{new}, we perform a backtracking line search (with Armijo's rule) over the step size parameter $\xi \in [0,1]$ to find the next iterate $c^{k+1} = (1-\xi)c^k + \xi c_{new}$. The algorithm is summarized below.

Algorithm: Newton+Armijo for c

1. initialize c^0.
2. for $k = 0$ to maxnumofiterations
3. compute $g(c^k)$ and $H(c^k)$.
4. solve the optimization problem in (3.18) for c_{new}.
5. set $c^{k+1} = (1-\xi)c^k + \xi c_{new}$ where step size ξ is searched using Armijo's rule.
6. break if stopping criterion is satisfied
7. end

Convergence analysis [3] shows that the practical performance of Newton+Armijo method with equality constrained convex optimization problems is exactly like the performance of Newton's method for unconstrained problems. Once c^k is near c^*, the optimum, convergence rate becomes quadratic and very few iterations are needed to reach the global optimum with a very high accuracy.

4 Relationship to Joint Regularized Multi-task Feature Selection [20]

Feature selection for multi-task learning using a joint regularization has been recently proposed by Obozinski et al. in [20] where the parameters β_t corresponding to different tasks are lined up to form a matrix with the t-th row representing the model parameter β_t for the t-th task. Then each column $\beta_j = [\beta_{1j} \ \ldots \ \beta_{Tj}]^T$ corresponds to a specific feature spanning all tasks. The basic idea is to penalize the sum of l_2 norms of respective columns of parameters associated with each feature spanning different tasks. This is considered as a l_1/l_2 norm, in other words, a combination of LASSO on the feature level and the l_2 regularization on the task level. Sparsity at the parameter column level is encouraged due to the l_1 regularization.

Mathematically, for any convex and continuously differentiable loss function $l(\beta, X, y)$, joint regulariza-

tion MTL [20] solves the following optimization problem:

$$\begin{aligned} (4.19) \qquad \min_{\beta} \quad & \textstyle\sum_{t=1}^{T} l(\beta_t, X_t, y_t) \\ \text{s.t.} \quad & \textstyle\sum_{j=1}^{d} \sqrt{\sum_{t=1}^{T} \beta_{tj}^2} \leq \kappa. \end{aligned}$$

We examine the relationship between our proposed algorithm and the joint regularized approach (4.19). The proposed approach as in the mathematical program (3.8) is closely related to the joint regularization approach. Notice that the joint regularized approach (4.19) is a convex program. Hence its KKT conditions are necessary and sufficient optimality conditions. Correspondingly, the KKT conditions for our program are necessary conditions and may not be sufficient. The proposed approach produces solutions that are as sparse as those obtained by the joint regularized approach. The following theorem characterizes our results.

THEOREM 4.1. *Consider the following two programs: Program 1:*

$$\begin{aligned} & \min_{beta} \sum_{t=1}^{T} \{ l(\beta_t, X_t, y_t) + \sum_{j=1}^{d} \nu_j \beta_{tj}^2 \} \\ & s.t. \qquad \frac{1}{d} \sum_{i=1}^{d} \frac{1}{\nu_i} = \gamma, \quad \nu_i > 0. \end{aligned}$$

Program 2:

$$\begin{aligned} \min_{\beta} \quad & \textstyle\sum_{t=1}^{T} l(\beta_t, X_t, y_t) \\ s.t. \quad & \textstyle\sum_{j=1}^{d} \sqrt{\sum_{t=1}^{T} \beta_{tj}^2} \leq \kappa. \end{aligned}$$

For any convex and continuously differentiable loss function $l(\beta, X, y)$, the KKT conditions of Program 1 are identical to the KKT conditions of Program 2.

Proof. The strict equality constraint in Problem 1 is not easy to deal with. Instead, we consider Problem 1 in the following equivalent form:

$$\begin{aligned} & \min_{\beta} \sum_{t=1}^{T} \{ l(\beta_t, X_t, y_t) + \sum_{j=1}^{d} \frac{\beta_{tj}^2}{\nu_j} \} \\ & \text{s.t.} \qquad \frac{1}{d} \sum_{i=1}^{d} \nu_i = \gamma, \quad \nu_i \geq 0. \end{aligned}$$

The corresponding Lagrangian is: $L(\beta, \nu) = \sum_{t=1}^{T} l(\beta_t, X_t, y_t) + \sum_{t=1}^{T} \sum_{j=1}^{d} \frac{\beta_{tj}^2}{\nu_j} + a(\frac{1}{d} \sum_{i=1}^{d} \nu_i - \gamma) - \vec{b} \cdot \vec{\nu}$, where a is a scalar and $\vec{b}$ is a vector with nonnegative components.

Then the KKT necessary conditions are as follows:

$$\begin{aligned} & \frac{\partial L}{\partial \nu_j} = -\sum_{t=1}^{T} \frac{\bar{\beta}_{tj}^2}{\bar{\nu}_j^2} + \frac{\bar{a}}{d} - \bar{b}_j = 0 \\ & \frac{\partial L}{\partial \beta_{tj}} = \frac{\partial l(\bar{\beta}_t, X_t, y_t)}{\partial \beta_{tj}} + 2\frac{\bar{\beta}_{tj}}{\bar{\nu}_j} = 0 \\ & \frac{1}{d} \sum_{i=1}^{d} \bar{\nu}_i = \gamma \\ & \vec{\bar{b}} \geq 0 \\ & \vec{\bar{\nu}} \geq 0 \\ & \bar{b}_j \bar{\nu}_j = 0, j = 1, \ldots, d \end{aligned}$$

After some algebra, we obtain the optimality condition

$$\forall (t = 1, \ldots, T, j = 1, \ldots, d), \begin{cases} \frac{\partial l(\bar{\beta}_t, X_t, y_t)}{\partial \beta_{tj}} + \\ \frac{2}{d\gamma} \left(\sum_{j=1}^{d} \sqrt{\sum_{t=1}^{T} \bar{\beta}_{tj}^2} \right) (\sum_{t=1}^{T} \bar{\beta}_{tj}^2)^{-\frac{1}{2}} \bar{\beta}_{tj} = 0 \\ \text{or } \bar{\beta}_{\cdot j} = 0, \end{cases}$$

where $\beta_{\cdot j} = 0$ denotes that for a specific number j, $\beta_{sj} = 0, \forall s = (1, \ldots, T)$.

It has been proved that Program 2 is equivalent to the following optimization problem when an appropriate λ is chosen to correspond to the choice of κ.

$$(4.20) \min_{\beta} \sum_{t=1}^{T} l(\beta_t, X_t, y_t) + \frac{1}{d\gamma} \left(\sum_{j=1}^{d} \sqrt{\sum_{t=1}^{T} \beta_{tj}^2} \right)^2.$$

Due to the convexity of this problem, its KKT conditions are necessary and sufficient and can be shown to be

$$\forall (t = 1, \ldots, T, j = 1, \ldots, d), \begin{cases} \frac{\partial l(\bar{\beta}_t, X_t, y_t)}{\partial \beta_{tj}} + \\ \frac{2}{d\gamma} \left(\sum_{j=1}^{d} \sqrt{\sum_{t=1}^{T} \bar{\beta}_{tj}^2} \right) (\sum_{t=1}^{T} \bar{\beta}_{tj}^2)^{-\frac{1}{2}} \bar{\beta}_{tj} = 0 \\ \text{or } \bar{\beta}_{\cdot j} = 0, \text{ and } \bar{\beta}_{\cdot j} = 0 \in \partial f(\beta_{\cdot j}), \end{cases}$$

where we use $f(\beta_{\cdot j})$ to denote the objective function in (4.20) as a function of $\beta_{\cdot j}$, and ∂f to denote the subgradient of function f. The use of subgradient is necessary since the objective f becomes nondifferentiable as its argument goes to zero.

Comparing the two KKT conditions, we can see immediately that Program 1 can get solutions at least as sparse as those from Program 2.

5 Experimental Results

We validate the proposed approach and the related algorithms by comparing them to standard approaches where individual tasks are solved independently using logistic regression for classification or ridge regression for regression as well as comparing to the pooling method where a single model is constructed using the available data from all tasks. These methods represent two extreme cases: the former one treats multiple tasks completely independently assuming no relatedness; the latter one treats all tasks identically. Our results clearly show that the multi-task learning approach as proposed is superior to these extreme cases.

Although we theoretically examined the relationship between our approach and the regularized multi-task feature selection approach [20] in Section 4, and hence it is expected to see similar numerical results generated by the two approaches, it is still desirable to perform some numerical experiments for comparison. However, the data used in [20] is not publicly available. It requires intensive research to develop efficient algorithms for solving the optimization problem in [20] and the gradient-descent implementation discussed in [20] is currently not an open source. We hence implemented another multi-task learning approach [5] that is different from [20] but also derived based on the regularization principle and we compared it to the proposed approach in terms of performance.

5.1 Synthetic Data We generated some synthetic data to verify the behavior of the proposed algorithms regarding the selected features and the performance. The synthetic data was generated as shown in the following figure.

Synthetic Data Generation

1. Set number of features $d = 20$, and number of tasks $T = 3$.

2. Generate $X \in R^{20}$ with each component $x_i \sim$ **Uniform**$[-1, 1], i = 1, \ldots, 20$.

3. The coefficient vectors of three tasks are specified as:

$$\beta_1 = [1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0]$$
$$\beta_2 = [1,1,1,1,1,1,1,1,0,1,0,1,0,0,0,0,0,0,0,0]$$
$$\beta_3 = [1,1,1,1,1,1,1,0,1,0,0,0,0,0,1,0,0,0,0,0]$$

4. For each task and each data vector, $y = \text{Sign}(\beta^T X)$.

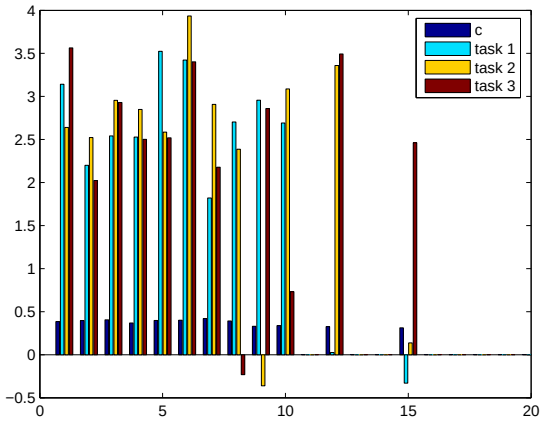

Figure 1: c and coefficient vectors for three tasks.

For each task, we generate 50 data for training and 2000 data for testing, and repeat the process 20 times. In figure 1, we show a bar plot of the averaged estimated coefficient vectors. In the figure, X axis is the variable indices ranging from 1 to 20. The Y axis denotes the coefficient values. For each variable group, from left to right are the values of c and coefficient values form task 1 to 3. Note that from our algorithm descriptions, if $c_i = 0$, then feature i will not be used by all tasks. Even if $c_i \neq 0$, for a particular t-th task, feature i could still be neglected if the coefficient in β_t is zero. Comparing to the ground truth, we see that the proposed joint feature selection algorithm worked well in terms of picking up relevant features, even with a very small sample size of 50 for each task. Note that for such a sparse setting, it is usually very hard to do feature selection under a single task learning setting.

To test the predictive performance of our proposed approach, we vary the smaple size from 50 to 120 with step size 10 and compare with single task learning based LASSO approach. Figure 2 shows the results. For lucid presentation, we have averaged the prediction errors from three tasks over 20 runs and drawn them in figure 2. It can be seen from the figure that our approach clearly outperform the STL based approach and as expected, the difference of these two approaches become smaller as the sample size of each task becomes larger.

5.2 CAD Data: Lung Cancer Prognosis Over the last decade, Computer-Aided Diagnosis (CAD) systems have moved from the sole realm of academic publications, to robust commercial systems that are used by physicians in their clinical practice. Our work presented

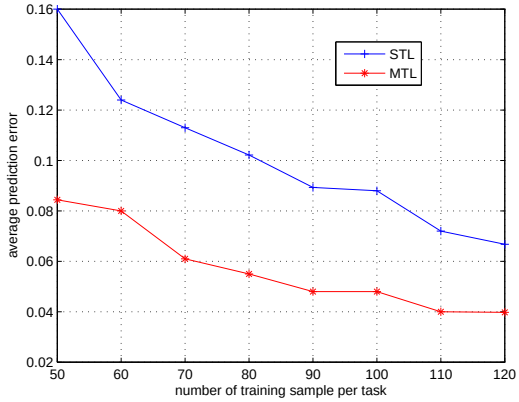

Figure 2: Plot of the prediction errors versus sample size.

in this paper was mainly motivated by the problems arisen in CAD systems. When building such a system, researchers often face a learning problem where multiple detection tasks are involved and are physically and clinically related.

5.2.1 Domain specification In many CAD applications, the goal is to detect potentially malignant tumors and lesions in medical images (computed tomography (CT) scans, X-ray, MRI etc). It is well recognized that the use of CAD not only offers the potential to decrease detection and recognition errors as a second reader, but also to reduce mistakes related to misinterpretation [2, 16]. The standard paradigm for computer aided diagnosis of medical images follows a sequence of three stages: identification of potentially unhealthy candidate regions of interest (ROI) from the image volume, computation of descriptive features for each candidate, and classification of each candidate (eg normal or diseased) based on its features.

In particular, we discuss an automatic lung cancer prognosis system in this paper. Lung cancer is the leading cause of cancer-related death in western countries with a mean 5 year survival rate for all stages of only 14%. The prognosis of stage I cancer is more optimistic with a mean 5 year survival rate of about 49%. Although multi-slice CT scanner allows acquisition of the entire chest with sub-millimeter slice thickness within a breath hold, only 15% of lung cancers are diagnosed at this early stage. Radiologic classification of small adenocarcinoma of lung by means of thoracic thin-section CT discriminates between the ground-glass opacities and solid nodules. The solid nodule is defined as an area of increased opacification more than 5mm in diameter, which completely obscures underlying vascular markings. Ground-glass opacity (GGO) is defined as an area of a slight, homogeneous increase in density, which does not obscure underlying vascular markings [22]. Figure 3 shows examples of a solid nodule and a GGO.

5.2.2 Data generation A prototype version of Siemens CAD system (not commercially available) was applied on a proprietary de-identified patient data set. The nodule dataset consisted of 176 high-resolution CT images (collected from multiple sites) that were randomly partitioned into two groups : a training set of 90 volumes and a test set of 86 volumes. The GGO dataset consisted of 60 CT images. Since there were only a limited number of GGO cases, they were not partitioned beforehand to have a test set. The original goal was to use the additional GGO cases to improve the nodule detection performance. In total, 129 nodules and 53 GGOs were identified and labeled by radiologists. Among the marked nodules, 81 appeared in the training set and 48 in the test set. The training set was then used to optimize the classification parameters, and construct the final classifier which was then tested on the independent test set of 86 volumes.

The candidate generation algorithm was independently applied to the training, test nodule sets and the GGO set, achieving 98.8% detection rate on the training set at 121 FPs per volume, 93.6% detection rate on the test set at 161 FPs per volume and 90.6% detection rate on the GGO set at 169 FPs per valume, resulting in totally 11056, 13985 and 10265 candidates in the respective nodule training, nodule test and GGO sets. There can exist multiple candidates pointing to one nodule or one GGO, so 131, 81 and 87 candidates were labeled as positive in the training set, test set and GGO set, respectively. A total of 86 numerical image features were designed to depict both nodules and GGOs. The feature set contained some low-level image features, such as size, shape, intensity, template matching features which required on average 15.6 millisec. cpu time per feature per candidate. Some sophisticated features were also designed and included in the feature set. For example, the multi-scale statistical features depicting higher-order intensity properties of nodules and GGOs. These features each on average need 2010 millisec. cpu time for a candidate with the current implementation. The specifications of all the related data sets are summarized in Table 5.2.2 for clarity.

5.2.3 Experimental setting and performance The first set of experiments were conducted as follows. We randomly sampled 25% (23 volumes) of the nodule patient data from the training set, 25% (15 volumes) of the GGO patient data. These samples were used in the

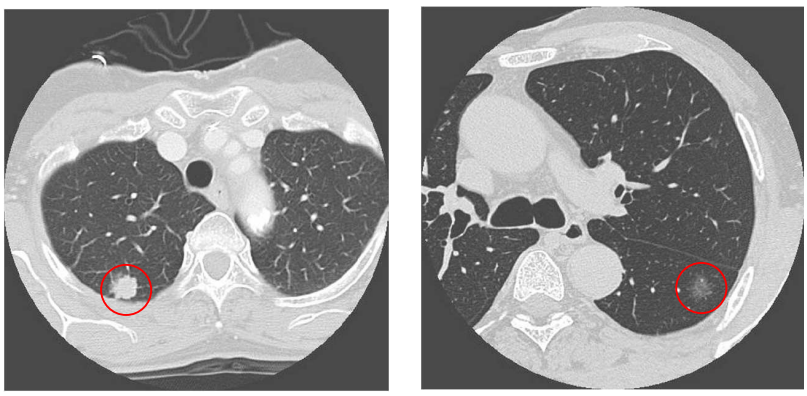

Figure 3: Examples of the slices in lung CT images: left – solid nodule; right – GGO.

	Nodule train	Nodule test	GGO
# patients	90	86	60
# cand.	11056	13985	10265
# cancer	81	48	53
# positives	131	81	87
# FP/vol	121	161	169
# feature	86	86	86

Table 1: Specifications of CAD datasets, '# cand'. means the number of candidates, '# cancer' means the number of cancerous tissues marked by radiologists, '# positives' means the number of candidates that are overlayed with cancerous tissues, and '# FP/vol' means the number of candidates that are not associated with any cancerous tissues, averaged over volumes.

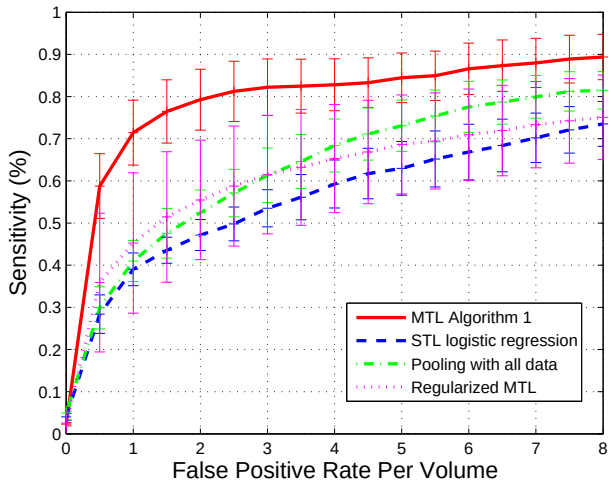

Figure 4: ROC plot of sensitivity versus false positive rate per volume using 25% of nodule and GGO training patient cases.

training phase. Notice that the random sampling can only take place at the patient level rather than the candidate level since otherwise information from a single patient will appear in both training and test sets, making the testing not independent. The nodule classifiers obtained by our approach and three other approaches were tested on the unseen test set of 86 patient cases. Since the GGO data was not partitioned, the resulting GGO classifiers were tested on the remaining set of cases after 15 volumes were sampled out from the set. We performed totally 15 trials by randomly re-sampling 15 times.

We compared our approach (3.10) with the implementation of Algorithm 1 to the single task learning with logistic regression, the pooling method where the two tasks are treated as identical task (which is called "pooling with all data" on the figures) and the regularized multi-task learning [5].

In the first trial, we tuned the model parameters such as γ in Algorithm 1 and the regularized parameters in [5] according to a 2-fold cross validation performance, and $\gamma = 100$ was the best choice for single task learning. Then we fixed $\gamma = 100$ for other trials. We use the same $\gamma = 100$ in the proposed multi-task learning formulation (3.10) for a fair comparison since the STL and MTL had the same parameter settings in this case. Note that the proposed MTL Algorithm 1 may produce better performance if we tune γ according to its own cross validation performance.

Figure 4 shows ROC curves averaged over the 15 trials together with test error variance (bars) drawn according to the standard deviation of detection rates of the 15 trials. Clearly, the plot shows that the proposed multi-task learning (3.10) generates a curve that dominates the ROC curves corresponding to other ap-

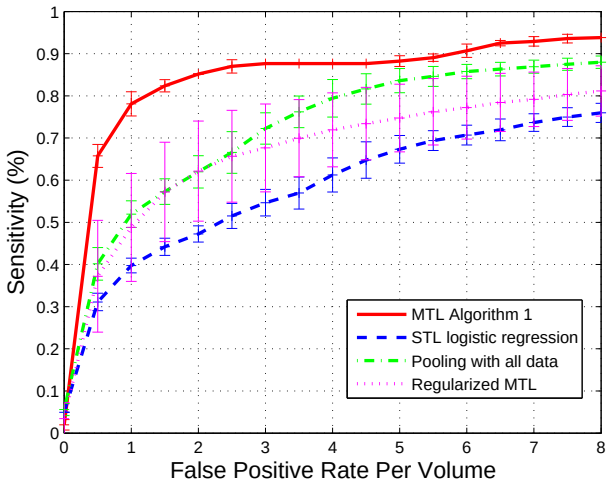

Figure 5: ROC plot of detection rate versus false positive per volume using 50% of nodule and GGO training patient volumes.

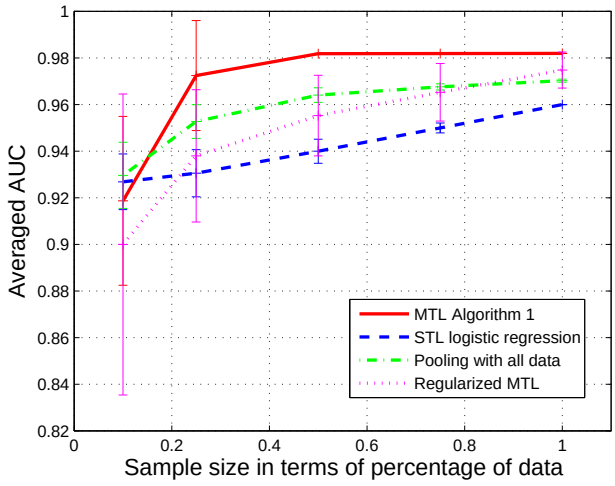

Figure 6: The AUC plot versus sample size.

proaches. It also had a relatively small model variance by referencing the error bars that are similar to those of STL with logistic regression and the pooling logistic regression where data from the two tasks were combined. The regularized MTL [5] did not always outperform the standard and pooling STL methods. Meanwhile, the classifier test error variance of the regularized MTL varied significantly with variations of samples as shown by the relatively large error bars in Figure 4. The same observation was also confirmed by performing the same experiments on an augmented sample (50%) of nodule training data and GGO data. Results obtained by sampling 50% of the nodule training and GGO data are shown in Figure 5. As expected, when more training data is presented, the detection rate (sensitivity) becomes higher and variance bars become smaller. We observed empirically that our Algorithm 1 often terminated within 15 alternating iterations. The common prior parameter $\mathbf{c}$ was stable to the variations of training data.

We also report the performance comparisons with area-under-the-ROC-curve (AUC) measure. AUC is a useful metric for classifier performance as it is independent of the decision criterion selected and prior probabilities [21]. We randomly sampled p% of training nodule set and the GGO set where $p = 10,\ 25,\ 50,\ 75,\ 100$. Obviously, when more and more data for a specific task is available, the resulting model achieves better performance, and accurate models can be learned with less help from other related tasks. We calculate the AUC for each ROC curve and the AUC numbers were averaged over 15 trials for each sample size choice p. Figure 6 illustrates the averaged AUC values and associated error bars whose length is proportional to standard deviation. We can see that the difference between the MTL and STL settings becomes less significant when more data is given for the nodule detection task. Again, our method presents relatively small model variance in comparison with the regularized MTL as shown in the error bars.

6 Conclusions

The joint feature selection problem was discussed in the multi-task learning setting. We proposed a probabilistic framework by imposing an automatic relevance determination prior on the hypothesis classes associated with each of the tasks. By regularizing the variance of the hypothesis parameters, similar feature patterns across different tasks are encouraged and features that are relevant to all (or most) of the tasks are identified. The proposed approach can be seen as a generalization of previous result from adaptive ridge regression to the multi-task learning setting. We prove that it is closely related to the joint feature selection approach in [20], and can produce solutions as sparse as the solutions obtained in [20]. Efficient algorithms are investigated to solve our formulation (Program 1) as described in Algorithm 1 and Algorithm 2. Our experimental results show that this approach outperforms the regularized multi-task learning approach [5] and the traditional single-task-learning methods. We also noticed in our experiments that not only the features that were shared by all tasks were selected, features that were particularly important to one specific task could also be selected in

the common prior c and then adjusted by α_t for each individual task. The exploration of the trade-off between the features commonly important to all tasks and features discriminative only for one or two tasks remains open for further research.

References

[1] R. K. Ando and T. Zhang, *A Framework for learning predictive structures from multiple tasks and unlabeled data*, Journal of Machine Learning Research, 2005.

[2] S. G. Armato-III, M. L. Giger, and H. MacMahon. *Automated detection of lung nodules in CT scans: preliminary results*, Medical Physics, 28(8):1552 – 1561, 2001.

[3] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[4] R. Caruana, *Multi-task learning*, Machine Learning, 28, p. 41-75, 1997.

[5] T. Evgeniou and M. Pontil, *Regularized multi-task learning*, Proceedings of the Tenth Conference on Knowledge Discovery and Data Mining, 2004.

[6] Y. Gandvalet, *Least absolute shrinkage is equivalent to quadratic penalization*, In Proceedings. of the International Conference on Artificial Neural Networks, Perspectives in Neural Computing, pages 201-206, 1998.

[7] Y. Gandvalet and S. Canu, *Outcomes of the equivalence of adaptive ridge with least absolute shrinkage*, In Advances in Neural Information Processing Systems, volume 11, MIT Press, 1999.

[8] W. Greene, *Econometric Analysis*, Prentice Hall, fifth edition, 2002.

[9] J. Goodman, *Exponential priors for maximum entropy models. In Proceedings of the Annual Meeting of the Association for Computational Linguistics*, In Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2004.

[10] I. Guyon and A. Ellisseeff, *A introduction to viable and feature selection*, Journal of Machine Learning Research, 3:1157-1182, 2003.

[11] T. Hastie, R. Tibshirani and J. Friedman, *The elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer-Verlag, 2001.

[12] T. Heskes, *Empirical Bayes for learning to learn*, Proceedings of ICML-2000.

[13] H. Liu and R. Setiono, *Incremental feature selection*, Applied Intelligence, 9(3):217:230, 1998.

[14] T. Jebara, *Multi-task feature and kernel selection for SVMs*, In Proceedings of the Twenty-Tirst International Conference on Machine learning (ICML), 2004.

[15] S. Lee, H. Lee, P. Abbeel and A. Y. Ng. *Efficient L1 regularized logistic regression*, In Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI 2006).

[16] D. P. Naidich, J. P. Ko, and J. Stoechek. *Computer aided diagnosis: Impact on nodule detection amongst community level radiologist. A multi-reader study*, In Proceedings of CARS 2004 Computer Assisted Radiology and Surgery, pages 902 – 907, 2004.

[17] D. J. C. MacKay, *Probable networks and plausible predictions - a review of practical Bayesian methods for supervised neural networks*, Network: Computation in Neural Systems, vol. 6, pp. 469-505, 1995.

[18] R. Malouf, *A comparison of algorithms for maximum entropy parameter estimation*, in Proceedings of the Sixth Conference on Natural Language Learning (CoNLL-2002).

[19] T. P. Minka, *A comparison of numerical optimizers for logistic regression*, http://research.microsoft.com/~minka/papers/logreg/.

[20] G. Obozinski, B. Taskar and M. I. Jordan *Multi-task Feature Selection*, Technical Report, UC Berkeley, 2006.

[21] F. Provost and T. Fawcett, *Robust Classification for Imprecise Environments*, Machine Learning, vol. 42/3, pp. 203-231, 2001.

[22] K. Suzuki, M. Kusumoto, S. Watanabe, R. Tsuchiya and H. Asamura, *Radiologic classfication of small adenocarcinoma of the Lung: Radiologic-Pathologic Correlation and Its Prognostic Impact*, The Annals of Thoracic Surgery CME Program, 81:413-20, 2006.

[23] R. Tibshirani, *Regression selection and shrinkage via the LASSO*, Journal of the Royal Statistics Society Series B, 58(1):267-288,1996.

[24] M. Tipping, *The relevance vector machine*, In Advances in Neural Information Processing Systems, volume 12, MIT Press, 2000.

[25] M. Tipping, *Sparse Bayesian learning and the relevance vector machine*, Journal of Machine Learning Research, 1:211-244, 2001.

[26] Y. Xue, X. Liao, L. Carin and B. Krishnapuram, *Learning multiple classifiers with Dirichlet process mixture priors*, In Workshop on Open Problems and Challenges for Nonparametric Bayesian Methods in Machine Learning at Neural Information Processing Systems, 2005.

[27] K. Yu, V. Tresp, and A. Schwaighofer, *Learning Gaussian Processes from Multiple Tasks*, Proceedings of the 22nd International Conference on Machine Learning (ICML), 2005.

[28] T. Zhang and F. Oles, *Text categorization based on regularized linear classifiers*, Information Retrieval, 4, 5-31.

[29] J. Zhang, *A Probabilistic framework for multi-task learning*, Ph.D thesis, Carnegie Mellon University, 2006.