An Improved Multi-task Learning Approach with Applications in Medical Diagnosis

Jinbo Bi[1], Tao Xiong[2], Shipeng Yu[1], Murat Dundar[1], and R. Bharat Rao[1]

[1] CAD and Knowledge Solutions, Siemens Medical Solutions
20 Valley Stream Parkway, Malvern, PA 19355, USA
jinbo.bi@siemens.com
[2] Risk Management, Applied Research, eBay Inc.
2145 Hamilton Avenue, San Jose, CA 95125, USA

Abstract. We propose a family of multi-task learning algorithms for collaborative computer aided diagnosis which aims to diagnose multiple clinically-related abnormal structures from medical images. Our formulations eliminate features irrelevant to all tasks, and identify discriminative features for each of the tasks. A probabilistic model is derived to justify the proposed learning formulations. By equivalence proof, some existing regularization-based methods can also be interpreted by our probabilistic model as imposing a Wishart hyperprior. Convergence analysis highlights the conditions under which the formulations achieve convexity and global convergence. Two real-world medical problems: lung cancer prognosis and heart wall motion analysis, are used to validate the proposed algorithms.

1 Introduction

Physicians routinely use computer aided diagnosis (CAD) systems in clinical practice [1]. It is well accepted that CAD systems decrease detection and recognition errors when used as a second reader [2]. Typically, the goal of a CAD system is to detect potentially abnormal structures in medical images. However, most CAD systems focus on the diagnosis of a single isolated abnormality using images taken only for the specific disease, which neglects a fundamental aspect of physicians diagnostic workflow where they examine not only primary abnormalities but also symptoms of related diseases.

For instance, an automated lung cancer CAD system can be built to separately identify solid nodules and ground glass opacities (GGOs). (Patients can have both structures, or GGOs can later become calcified GGOs which become solid or partly-solid nodules.) Radiologic classification of small adenocarcinoma of lung by means of thoracic thin-section CT discriminates between solid nodules and GGOs. Fig. 1 shows two CT slices with a nodule and a GGO respectively. A solid nodule is defined as an area of increased opacification more than 5mm in diameter, which completely obscures underlying vascular markings. A ground-glass opacity (GGO) is defined as an area of a slight homogeneous increase in density, which does not obscure underlying vascular markings [3]. Detecting

W. Daelemans et al. (Eds.): ECML PKDD 2008, Part I, LNAI 5211, pp. 117–132, 2008.
© Springer-Verlag Berlin Heidelberg 2008

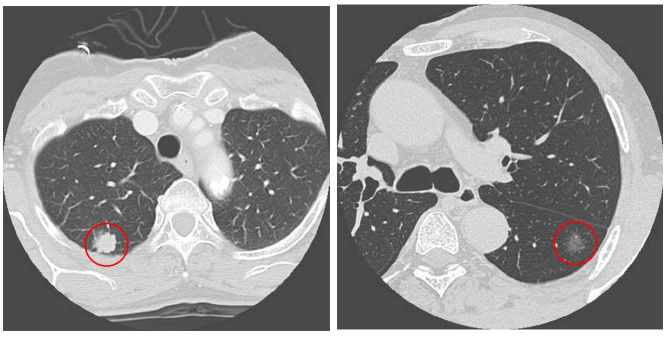

Fig. 1. Lung CT images: left, solid nodule; right, ground glass opacity (GGO)

nodules and detecting GGOs are two dependent tasks with their own respective characteristics, and is thus more sensible to be tackled jointly.

Another example is the wall motion analysis of the left ventricle which is used to diagnose ischemia diseases. The left ventricle wall is medically segmented into 16 segments. Fig. 2 shows an ultrasound image of left ventricle in the apical four chamber (A4C) view and the six segments seen from this view. The task is to predict the wall motion abnormality of each segment by extracting features from cardiac ultrasound images and classifying each segment as being normal versus abnormal. Left ventricle segments are physically connected, and if any segment has abnormalities, the neighboring segments are affected, which makes jointly learning the classifiers both necessary and beneficial.

We introduce a concept – "collaborative" computer aided diagnosis (CCAD) – that aims to improve the diagnosis of a single abnormality by fusing information, knowledge or data from various related sources, such as detecting nodules not only by itself but also by learning from multiple related abnormal structures simultaneously. In the machine learning field, the collaborative learning problem

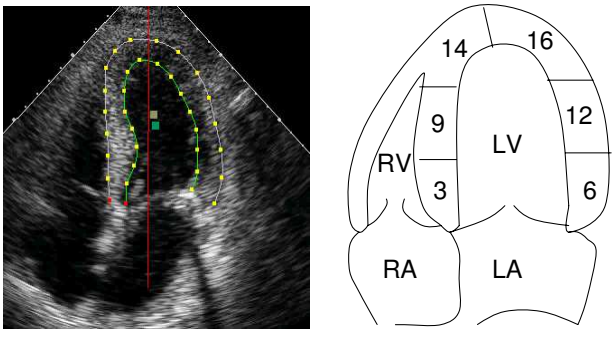

Fig. 2. Ultrasound image of heart: left, ultrasound image of A4C view; right, segments of left ventricle seen from A4C view

has been cast as multi-task learning (MTL), collaborative filtering or collaborative prediction. Multi-task learning is a learning methodology that estimates models for several tasks in a joint manner. Although almost all existing multi-task learning methods assume some relatedness among tasks, the definition of relatedness varies [4,5,6]. From the hierarchical Bayesian viewpoint [7], multi-task learning essentially seeks to learn a good prior over all tasks to capture task dependencies.

We model the across-task relatedness as sharing a common feature or kernel representation. Dimension reduction or sparse kernel representation is essential for CAD applications. Previous work on selecting features for multiple related tasks include the work in [8] that is based on maximum entropy discrimination, and the regularization-based methods [9,10] by applying a joint regularization of the model parameters. We derive a family of effective approaches, which generalizes our early multi-task learning study [12], by directly maximizing the joint *a posterior* distribution across tasks. By imposing a hyperprior that corresponds to a trace norm constraint [11] on model parameter variance, we are able to eliminate features irrelevant to all tasks as well as select discriminative features for each individual task.

2 MTL Algorithms

Assume that we have T tasks in total, for each task t, we have sample set $(\mathbf{X}_t \in R^{\ell_t \times d}, \mathbf{y}_t \in R^{\ell_t})$. The matrix $\mathbf{X}_t$ is the feature matrix or kernel matrix where the i-th row corresponds to the i-th example $\mathbf{x}_i^t$ of task t, and each column represents a feature or a kernel basis, and $\mathbf{y}_t$ denotes the label vector where the i-th component is y_i^t. We consider functions of the form $\mathbf{x}^\top \boldsymbol{\alpha}$ which is linear in terms of the model parameter $\boldsymbol{\alpha}$. We focus on models where $\mathbf{x}$ is in the original feature space but many discussions in this article can be extended to kernel spaces.

To learn the parameter vector α_t, single task learning methods minimize a regularized risk $L(\boldsymbol{\alpha}_t, \mathbf{X}_t, \mathbf{y}_t) + \lambda P(\boldsymbol{\alpha}_t)$ for an optimal $\boldsymbol{\alpha}_t$ where P is a regularization operator, such as a 2-norm penalty on $\boldsymbol{\alpha}_t$, i.e., $\sum_{j=1}^d (\alpha_{tj})^2$, or a 1-norm penalty, $\sum_{j=1}^d |\alpha_{tj}|$, L defines the loss term, and λ balances between L and P. For example, the logistic regression loss

$$L(\boldsymbol{\alpha}_t, \mathbf{X}_t, \mathbf{y}_t) = \sum_{i=1}^{\ell_t} \log(1 + \exp(-\sum_{j=1}^{d} \alpha_{tj} x_{ij}^t y_i^t)) \tag{1}$$

and the least squares loss

$$L(\boldsymbol{\alpha}_t, \mathbf{X}_t, \mathbf{y}_t) = \sum_{i=1}^{\ell_t} (\sum_{j=1}^{d} \alpha_{tj} x_{ij}^t - y_i^t)^2 \tag{2}$$

are both strictly convex functions in terms of $\boldsymbol{\alpha}_t$.

A family of joint learning algorithms can be derived by rewriting $\boldsymbol{\alpha}_t = \mathbf{C}\boldsymbol{\beta}_t$ where $\mathbf{C}$ is a diagonal matrix with diagonal vector equal to $\mathbf{c} \geq 0$, and we solve the following problem over all tasks:

$$\begin{array}{ll} \min\limits_{\boldsymbol{\beta}_t, t=1,\cdots,T, \mathbf{c}\geq 0} & \sum\limits_{t=1}^{T} \left(L(\mathbf{C}\boldsymbol{\beta}_t, \mathbf{X}_t, \mathbf{y}_t) + P_1(\boldsymbol{\beta}_t)\right) \\ \text{subject to} & P_2(\mathbf{c}) \leq \gamma, \end{array} \tag{3}$$

where P_1 and P_2 are any suitable regularization operators. For each task t, solving problem (3) constructs a function $f(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\alpha}_t = \mathbf{x}^\top \mathbf{C}\boldsymbol{\beta}_t = \sum_j x_j c_j \beta_{tj}$ where $\boldsymbol{\beta}_t$ is task-specific while the same $\mathbf{c}$ is used across different tasks. We call $\mathbf{c}$ an indicator vector indicating if an according feature is used in the model. Typically $\mathbf{c}$ comprises entries that are equal to 0 or 1, which leads to difficult combinatorial optimization problems, and thus has been relaxed to non-negative real values in Problem (3). If $c_j = 0$, the j-th variable is not used in any model for all tasks regardless of the value of a specific $\boldsymbol{\beta}$. Otherwise if $c_j > 0$, the j-th variable appears in all models but an appropriate $\boldsymbol{\beta}$ vector can rule out this feature for a particular task. In other words, $\mathbf{c}$ is used to eliminate any irrelevant features, and $\boldsymbol{\beta}_t$ selects the best suitable features for each individual task.

Many regularization terms can be considered for the choices of P_1 and P_2. For example, if the 2-norm regularization is employed for both P_1 and P_2, the problem (3) becomes

$$\begin{array}{ll} \min\limits_{\boldsymbol{\beta}_t, t=1,\cdots,T, \mathbf{c}\geq 0} & \sum\limits_{t=1}^{T} \left(L(\mathbf{C}\boldsymbol{\beta}_t, \mathbf{X}_t, \mathbf{y}_t) + \sum_{j=1}^{d} \beta_{tj}^2\right) \\ \text{subject to} & \sum_{j=1}^{d} c_j^2 \leq \gamma, \end{array} \tag{4}$$

where $\gamma > 0$ is a tuning parameter. Empirical results included in [12] demonstrate the effectiveness of the formulation (3) with $P_1(\cdot) = \sum_{j=1}^{d} \beta_{tj}^2$ and $P_2(\cdot) = \sum_{j=1}^{d} |c_j|$.

To effectively optimize (3), we design an alternating optimization algorithm, which is, in spirit, similar to the Expectation-Maximization approach. At iteration s, the "E" step estimates the optimal $\mathbf{c}^s$, which serves the common prior, based on $\boldsymbol{\beta}^{s-1}$. The "M" step estimates a new $\boldsymbol{\beta}_t^s$ for each t by maximizing the posterior based on $\mathbf{c}^s$. The algorithm does the following steps at the s-th iteration:

Algorithm $\mathcal{A}(\mathbf{C}^{s-1}, \ \boldsymbol{\beta}_t^{s-1}, t = 1, \cdots, T)$

– Fix $\mathbf{C} = \mathbf{C}^{s-1}$ (initially, to $\mathbf{I}$), convert $\tilde{\mathbf{X}}_t \leftarrow \mathbf{X}_t\mathbf{C}$, solve (5) for $\boldsymbol{\beta}_t^s$,

$$\forall\ t = 1, \cdots, T, \min_{\boldsymbol{\beta}_t} L(\boldsymbol{\beta}_t, \tilde{\mathbf{X}}_t, \mathbf{y}_t) + P_1(\boldsymbol{\beta}_t). \tag{5}$$

– Fix $\boldsymbol{\beta}_t = \boldsymbol{\beta}_t^s$, convert $\hat{\mathbf{X}}_t \leftarrow \mathbf{X}_t\mathbf{B}_t$ where $\mathbf{B}_t$ is a diagonal matrix with diagonal elements equal to $\boldsymbol{\beta}_t^s$, solve problem (6) for $\boldsymbol{c}^s$,

$$\min_{\mathbf{c}\geq 0} \ L(\mathbf{c}, \hat{\mathbf{X}}_t, \mathbf{y}_t), \qquad \text{subject to} \ \ P_2(\mathbf{c}) \leq \gamma. \tag{6}$$

---

3 A Statistical Justification

Let $p(\mathbf{y}_t|\mathbf{X}_t, \mathbf{w}_t)$ specify the likelihood for task t, with a noise model which is independent of other tasks. Here $\mathbf{w}_t$ is the model parameter to be determined. In a hierarchical Bayesian framework, we assume all the function weights $\mathbf{w}_t$ are *i.i.d.* sampled from a common prior $p(\cdot)$, which accounts for the dependencies between different tasks. Typically a zero mean Gaussian prior with covariance $\mathbf{\Sigma}$ is assigned to the weights $\mathbf{w}_t$, i.e., $\mathbf{w}_t \sim N(\mathbf{0}, \mathbf{\Sigma})$. Then the *a posteriori* distribution of all function coefficients $\{\mathbf{w}_t\}$ can be calculated via Bayes rule as, $p(\mathbf{W}|\mathbf{X}, \mathbf{y}, \mathbf{\Sigma}) \propto \prod_t p(\mathbf{y}_t|\mathbf{X}_t, \mathbf{w}_t)p(\mathbf{w}_t|\mathbf{\Sigma})$ where $\mathbf{W}$ is a matrix containing all weight vectors $\mathbf{w}_t$ as rows, and $\mathbf{X}$, $\mathbf{y}$ contain data from all tasks.

We are interested in learning the shared covariance matrix $\mathbf{\Sigma}$ rather than fixing it. A Bayesian treatment would be to assign a hyperprior to $\mathbf{\Sigma}$ and learn $\mathbf{W}$ and $\mathbf{\Sigma}$ jointly. Since $\mathbf{\Sigma}$ is symmetric and positive definite, one choice of the prior is $p(\mathbf{\Sigma}) \propto |\mathbf{\Sigma}|^{T/2} \exp\left(-\frac{1}{2}\operatorname{tr}(\mathbf{\Sigma})\right)$, with $\operatorname{tr}(\cdot)$ the matrix trace. This is essentially a Wishart distribution with degrees of freedom $d + T + 1$ and scale matrix $\mathbf{I}$. Now the joint *a posteriori* distribution of $(\mathbf{W}, \mathbf{\Sigma})$ is

$$p(\mathbf{W}, \mathbf{\Sigma}|\mathbf{X}, \mathbf{y}) \propto p(\mathbf{\Sigma}) \prod_t p(\mathbf{y}_t|\mathbf{X}_t, \mathbf{w}_t)p(\mathbf{w}_t|\mathbf{\Sigma}). \tag{7}$$

The *maximum a posteriori* (MAP) estimate is to find a point estimate that maximizes the posterior (7). This is equivalent to solving the following optimization problem $\min_{\mathbf{w}_t, \mathbf{\Sigma}} \quad \sum_{t=1}^{T} \left(L(\mathbf{w}_t, \mathbf{X}_t, \mathbf{y}_t) + \mathbf{w}_t^\top \mathbf{\Sigma}^{-1} \mathbf{w}_t \right) + \operatorname{tr}(\mathbf{\Sigma})$ by taking the negation of the logarithm of (7) and removing the normalization constant. Here the loss function for each task t is $L(\mathbf{w}_t, \mathbf{X}_t, \mathbf{y}_t) \propto -\log p(\mathbf{y}_t|\mathbf{X}_t, \mathbf{w}_t)$. This can also be equivalently written as

$$\begin{array}{ll} \min\limits_{\mathbf{w}_t, t=1,\cdots,T, \mathbf{\Sigma}} & \sum\limits_{t=1}^{T} L(\mathbf{w}_t, \mathbf{X}_t, \mathbf{y}_t) + \mathbf{w}_t^\top \mathbf{\Sigma}^{-1} \mathbf{w}_t, \\ \text{subject to} & \operatorname{tr}(\mathbf{\Sigma}) \leq \gamma \end{array} \tag{8}$$

with an appropriately chosen $\gamma > 0$. For each of the task t, this trace condition essentially requires that the expected variance of each weight component w_t^j of $\mathbf{w}_t, \forall t$, is proportional to γ. With a small γ, some components will become small to achieve *sparse estimates* of $\mathbf{w}_t$. Thus this formulation leads to a *jointly sparse structure* of the weight matrix $\mathbf{W}$.

If we decompose the matrix $\mathbf{\Sigma}$ to its eigen-form $\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ where $\mathbf{U}$ is an orthonormal matrix and the diagonal matrix $\mathbf{\Lambda}$ contains eigen-values $\sigma_j \geq 0$, the problem (8) becomes an equivalent form with $\boldsymbol{\alpha}_t = \mathbf{U}^\top \mathbf{w}_t$:

$$\begin{array}{ll} \min\limits_{\boldsymbol{\alpha}_t, t=1,\cdots,T, \mathbf{U}, \boldsymbol{\sigma} \geq 0} & \sum\limits_{t=1}^{T} \left(L(\boldsymbol{\alpha}_t, \mathbf{X}_t\mathbf{U}, \mathbf{y}_t) + \sum\limits_{j=1}^{d} \frac{1}{\sigma_j} \alpha_{tj}^2 \right), \\ \text{subject to} & \sum\limits_{j=1}^{d} \sigma_j \leq \gamma. \end{array} \tag{9}$$

Problem (9) implies that the original input $\mathbf{x}$ has been transformed to $\mathbf{U}^\top\mathbf{x}$ and then a linear function is constructed in the transformed space where features are independent.

Many image features in CAD applications are computationally expensive, so one of the major goals is to reduce the number of image features in the models. Since the resulting orthonormal $\mathbf{U}$ may not be sparse, we assume $\mathbf{U} = \mathbf{I}$ to enforce the sparsity on the original image features instead of sparse representations in the transformed space. By showing equivalence between (4) and (9), the probabilistic model in this section provides a statistical justification for our algorithms.

Theorem 1. *For any optimal solution of Problem (9) where* $\mathbf{U} = \mathbf{I}$*, there is a corresponding optimal solution to Problem (4), and vice versa.*

The proof can be obtained by change of variables as follows: $\beta_{tj} = \alpha_{tj}/\sqrt{\sigma_j}$, $\forall t = 1, \ldots, T$, $c_j = \sqrt{\sigma_j}$, $j = 1, \ldots, d$. Correspondingly, $\alpha_{tj} = c_j\beta_{tj}$ and $\sum_j c_j^2 = \sum_j \sigma_j \leq \gamma$.

4 Connection to Existing Methods

Feature selection for multi-task learning using a joint regularization has been recently proposed in [9] where a so-called ℓ_1/ℓ_2 norm is applied to the weight matrix $\mathbf{A}$ formed by all $\boldsymbol{\alpha}_t$ as rows. A more recent work [10] dedicated to multi-task feature learning has defined a new norm as $\|\mathbf{A}\|_{2,1} = \sum_{j=1}^{d} \sqrt{\sum_{t=1}^{T} \alpha_{tj}^2}$, which is the same as the ℓ_1/ℓ_2 norm in [9]. Assuming $\mathbf{U} = \mathbf{I}$, both work essentially solves the following optimization problem

$$\begin{aligned} &\min_{\boldsymbol{\alpha}_t, t=1,\cdots,T} \sum_{t=1}^{T} L(\boldsymbol{\alpha}_t, \mathbf{X}_t, \mathbf{y}_t), \\ &\text{subject to } \sum_{j=1}^{d} \sqrt{\sum_{t=1}^{T} \alpha_{tj}^2} \leq \kappa, \end{aligned} \tag{10}$$

or an equivalent problem as follows

$$\min_{\boldsymbol{\alpha}_t, t=1,\cdots,T} \sum_{t=1}^{T} L(\boldsymbol{\alpha}_t, \mathbf{X}_t, \mathbf{y}_t) + \lambda \left(\sum_{j=1}^{d} \sqrt{\sum_{t=1}^{T} \alpha_{tj}^2} \right)^2, \tag{11}$$

where κ and λ are pre-specified parameters, and these two problems are equivalent for properly chosen κ and λ. The problem in [9] does not use the squared regularized term as in problem (10) whereas the formulation in [10] uses the square of $\|\mathbf{A}\|_{2,1}$ as the second term in problem (11).

As shown in [9,10], these regularization-based algorithms advance the multi-task learning research, but there has been lack of investigation if a probabilistic interpretation exists for these methods. The following theorem characterizes the connection between our formulation and (11). Hence, our probabilistic model is also feasible to justify these approaches that these methods assume a Wishart prior on the common covariance $\boldsymbol{\Sigma}$ of function weights $\mathbf{w}_t, \forall t$.

Theorem 2. *The Karush-Kuhn-Tucker (KKT) conditions of Problem (9) with* $\mathbf{U} = \mathbf{I}$ *are identical to the KKT conditions of Problem (11) for any convex and continuously differentiable loss function* $L(\boldsymbol{\alpha}, \mathbf{X}, \mathbf{y})$.

Proof. The Lagrangian of the problem (9) is:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\alpha}_t, \boldsymbol{\sigma}, a, \mathbf{b}) = \textstyle\sum_{t=1}^{T} L(\boldsymbol{\alpha}_t, \mathbf{X}_t, \mathbf{y}_t) + \sum_{t=1}^{T}\sum_{j=1}^{d} \frac{\alpha_{tj}^2}{\sigma_j} \\ + a(\textstyle\sum_{j=1}^{d} \sigma_j - \gamma) - \mathbf{b}^{\mathbf{T}}\boldsymbol{\sigma},\end{aligned}$$

where a and $\mathbf{b}$ are Lagrangian multipliers, and $a \geq 0$ is a scalar and $\mathbf{b} \geq \mathbf{0}$ is a vector.

Problem (9) minimizes a convex objective over a convex feasible region, and thus is a convex program. Then the KKT necessary and sufficient conditions are as follows:

$$\frac{\partial \mathcal{L}}{\partial \sigma_j} = -\textstyle\sum_{t=1}^{T} \frac{\alpha_{tj}^2}{\sigma_j^2} + a - b_j = 0, \tag{12}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_{tj}} = \frac{\partial L(\boldsymbol{\alpha}_t, \mathbf{X}_t, \mathbf{y}_t)}{\partial \alpha_{tj}} + 2\frac{\alpha_{tj}}{\sigma_j} = 0, \tag{13}$$

$$\textstyle\sum_{j=1}^{d} \sigma_j \leq \gamma, \quad a \geq 0, \quad \mathbf{b} \geq 0, \quad \boldsymbol{\sigma} \geq 0 \tag{14}$$

$$a(\textstyle\sum_{j=1}^{d} \sigma_j - \gamma) = 0 \tag{15}$$

$$b_j \sigma_j = 0, \quad j = 1, \ldots, d \tag{16}$$

Now we discuss the various cases in terms of the Lagrange multipliers a and b_j.

(1) If $b_j > 0$, by the complementary condition (16), $\sigma_j = 0$. It implies $\alpha_{\cdot j} = 0$ which denotes that for a specific number j, $\alpha_{tj} = 0, \forall t = (1, \ldots, T)$.
(2) If $b_j = 0$ (implying $\sigma_j > 0$) and $a = 0$, by KKT condition (12), $\sum_{t=1}^{T} \alpha_{tj}^2 = 0$. Hence, $\alpha_{\cdot j} = 0$.
(3) If $b_j = 0$ and $a > 0$ (implying $\sum_j \sigma_j = \gamma$ by (15)), then $a = \frac{1}{\sigma_j^2} \sum_{t=1}^{T} \alpha_{tj}^2$, and further we have $\sigma_j = \gamma \sqrt{\sum_{t=1}^{T} \alpha_{tj}^2} / \sum_{j=1}^{d} \sqrt{\sum_{t=1}^{T} \alpha_{tj}^2}$. Substituting σ_j into KKT condition (13) yields the optimality condition, which can be summarized as follows:

$$\forall (t = 1, \ldots, T, j = 1, \ldots, d),$$
$$\begin{cases} \frac{\partial L(\boldsymbol{\alpha}_t, \mathbf{X}_t, \mathbf{y}_t)}{\partial \alpha_{tj}} + \frac{2}{\gamma} \left(\sum_{j=1}^{d} \sqrt{\sum_{t=1}^{T} \alpha_{tj}^2} \right) (\sum_{t=1}^{T} \alpha_{tj}^2)^{-\frac{1}{2}} \alpha_{tj} = 0 \\ \text{or } \alpha_{\cdot j} = 0, \end{cases}$$

Now let $\lambda = 1/\gamma$ in Problem (11). Due to the convexity of Problem (11), its KKT conditions are necessary and sufficient and can be shown as

$$\forall (t = 1, \ldots, T, j = 1, \ldots, d),$$
$$\begin{cases} \frac{\partial l(\boldsymbol{\alpha}_t, \mathbf{X}_t, \mathbf{y}_t)}{\partial \alpha_{tj}} + \frac{2}{\gamma} \left(\sum_{j=1}^{d} \sqrt{\sum_{t=1}^{T} \alpha_{tj}^2} \right) (\sum_{t=1}^{T} \alpha_{tj}^2)^{-\frac{1}{2}} \alpha_{tj} = 0 \\ \text{or } \alpha_{\cdot j} = 0, \text{ and } \alpha_{\cdot j} = 0 \in \partial g(\alpha_{\cdot j}), \end{cases}$$

where we use $g(\alpha_{\cdot j})$ to denote the objective function in (11) as a function of $\alpha_{\cdot j}$, and ∂f to denote the subgradient of function g. The use of subgradient is necessary since the objective g becomes nondifferentiable as its argument goes to zero. The equivalence is established by comparing the two sets of KKT conditions.

Theorem 2 establishes an equivalence between the learning formulations (10), (11) and (9) (more precisely (4)). Hence our probabilistic model in Section 3 can interpret all these formulations as assuming a common covariance matrix $\mathbf{\Sigma}$ across $\mathbf{w}_t$, $\forall t$ and employing a Wishart hyperprior on $\mathbf{\Sigma}$. Furthermore, as a byproduct of Theorem 2, a closed-form solution for $\mathbf{c}$ can be further derived to simplify calculation in Algorithm $\mathcal{A}$ specifically for the formulation (4).

Theorem 3. *Given $\hat{\boldsymbol{\beta}}_t$, $t = 1, \cdots, T$, that are optimal solutions of Problem (4),*

$$c_j = \sqrt{\frac{\gamma \sum_{t=1}^{T} \hat{\beta}_{tj}^2}{\sum_{j=1}^{d} \sum_{t=1}^{T} \hat{\beta}_{tj}^2}}, \quad j = 1, \cdots, d$$

are optimal to Problem (4).

Proof. The proof can be obtained by re-examining Theorem 1 from which we have $\alpha_{tj} = \sqrt{\sigma_j}\beta_{tj}$ and $c_j = \sqrt{\sigma_j}$, and Theorem 2 from which we have $\sigma_j = \gamma\sqrt{\sum_{t=1}^{T} \alpha_{tj}^2}/(\sum_{j=1}^{d} \sqrt{\sum_{t=1}^{T} \alpha_{tj}^2})$. Substituting α_{tj} into the formula for σ_j yields $\sigma_j = \gamma \sum_{t=1}^{T} \beta_{tj}^2/(\sum_{j=1}^{d} \sum_{t=1}^{T} \beta_{tj}^2)$. Taking the square root of σ_j yields the formula for c_j.

5 Convergence Analysis

Although alternating optimization has been used to develop many efficient algorithms, the convergence proof does not necessarily exist. Convergence analysis usually encloses local convergence and global convergence properties. Local convergence implies that the algorithm converges to a solution $(\hat{\boldsymbol{\beta}}_t, \hat{\mathbf{c}})$ if being initialized from a close neighborhood of $(\hat{\boldsymbol{\beta}}_t, \hat{\mathbf{c}})$. A global convergence proves that the algorithm converges when initialized at any arbitrary points in the feasible region $\mathcal{S}$.

The local convergence property of Algorithm $\mathcal{A}$ is analyzed for Formulation (4). The key point is the requirement of the local strict convexity of the loss function L with respect to $(\boldsymbol{\beta}_t, \mathbf{c})$.

Theorem 4. *Let $(\hat{\boldsymbol{\beta}}_t, \hat{\mathbf{c}})$ be a local minimizer of Problem (4). If $\exists$ a neighborhood $\mathcal{N}$ of $(\hat{\boldsymbol{\beta}}_t, \hat{\mathbf{c}})$, such that the loss function L has continuous second-order derivatives and is strictly convex in $\mathcal{N}$, then $\exists\, \hat{\mathcal{N}}((\hat{\boldsymbol{\beta}}_t, \hat{\mathbf{c}}))$ for any initial point in $\hat{\mathcal{N}}((\hat{\boldsymbol{\beta}}_t, \hat{\mathbf{c}}))$, Algorithm $\mathcal{A}$ converges q-linearly to $(\hat{\boldsymbol{\beta}}_t, \hat{\mathbf{c}})$.*

Proof. Solving Problem (4) is equivalent to minimizing (5) with a properly chosen $\bar{\gamma} > 0$

$$\min_{\boldsymbol{\beta}_t, t=1,\cdots,T, \mathbf{c}\geq 0} \quad g(\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_T, \mathbf{c}) = \sum_{t=1}^{T} L(\mathbf{C}\boldsymbol{\beta}_t, \mathbf{X}_t, \mathbf{y}_t) + \sum_{t=1}^{T}\sum_{j=1}^{d} \beta_{tj}^2 + \bar{\gamma}\sum_{j=1}^{d} c_j^2$$

The objective function g has continuous second-order derivatives with respect to $\boldsymbol{\beta}_t$ and $\mathbf{c}$ and is strict convex in $\mathcal{N}$ due to the local property of the loss function L. Then the local convergence result developed in [13] on unconstrained problems is applied to show our theorem.

Global convergence analysis is usually more difficult and requires stronger conditions. We use the results developed in the mathematical programming field [13,14] to obtain a global convergence analysis which requires that both sub-problems, (5) and (6), have an unique optimal solution. This condition highly relies on the property of loss functions. If strictly convex loss functions are employed, such as the logistic regression loss or least squares loss, the loss term $L(\mathbf{C}\boldsymbol{\beta}_t, \mathbf{X}_t, \mathbf{y}_t)$ is bi-convex with respect to $(\boldsymbol{\beta}_t, \mathbf{c})$, in other words, is strictly convex with respect to $\boldsymbol{\beta}_t$ if $\mathbf{c}$ is fixed, and vice versa. The strict bi-convexity guarantees that sub-problems have an unique solution.

Let us denote the feasible set of the problem (4) as $\mathcal{S}$. In Algorithm $\mathcal{A}$, $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2$ where $\mathcal{S}_1$ is the feasible region for $\mathbf{c}$, $\mathcal{S}_1 = \{\mathbf{c} \mid \|\mathbf{c}\|^2 \leq \gamma\}$, and $\mathcal{S}_2$ is the feasible region for $\boldsymbol{\beta}_t,\ t = 1, \cdots, T$. Problem (4) has a 2-norm regularization on $\boldsymbol{\beta}$, so each $\boldsymbol{\beta}_t$ has to remain in the set $\mathcal{S}_2 = \{(\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_T) | \sum_{t=1}^{T}\sum_{j=1}^{d} \beta_{tj}^2 \leq \tilde{\gamma}\}$ for a $\tilde{\gamma} > 0$.

Theorem 5. *Let Ω be the set of fixed points of $\mathcal{A}$ as*

$$\{(\mathbf{c}, \boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_T) \in \mathcal{S} \mid (\mathbf{c}, \boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_T) = \mathcal{A}(\mathbf{c}, \boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_T)\}.$$

If the loss function L is strictly convex in terms of $\boldsymbol{\beta}_t,\ \forall\ t$, for fixed $\mathbf{c}$ and is also strictly convex in terms of $\mathbf{c}$ for fixed $\boldsymbol{\beta}$, and the regularizers P_1 and P_2 are strictly convex respectively in terms of $\boldsymbol{\beta}_t$ and $\mathbf{c}$, then for any initial point in $\mathcal{S}$, Algorithm $\mathcal{A}$

(i) either converges to Ω;

(ii) or the limit of every convergence subsequence is in Ω.

Proof. To achieve the results (i) or (ii), the theorem shown in [13,14] requires the following conditions: (a) each sub-problem in $\mathcal{A}$ has an unique optimal solution; (b) $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2$, where each $\mathcal{S}_i$ is a compact subset in a real space of proper dimension. We thus validate the satisfaction of these conditions.
Since the objective function of the unconstrained equivalent form (5) is strictly convex in terms of one set of variables when fixing the other due to the strict convexity of L, P_1 and P_2, (a) holds. Obviously, $\mathcal{S}_1 = \{\mathbf{c} | P_2(\mathbf{c}) \leq \gamma\}$ is a closed and bounded ball in a d-dimensional space, and $\mathcal{S}_2 = \{\boldsymbol{\beta} | \sum_t P_1(\boldsymbol{\beta}_t) \leq \tilde{\gamma}\}$ defines a closed and bounded ball in a $(d \times T)$-dimensional space. Thus both sets are compact subsets of a real space.

Some common loss functions satisfy the conditions in Theorem 5. For example, in the logistic regression loss function (1) and the least squares loss function (2), $L(\boldsymbol{\alpha}_t, \mathbf{X}_t, \mathbf{y}_t)$ is strictly convex in terms of $\boldsymbol{\alpha}_t$. Hence $L(\mathbf{C}\boldsymbol{\beta}_t, \mathbf{X}_t, \mathbf{y}_t)$ is strictly convex in terms of $\mathbf{c}$ if all $\boldsymbol{\beta}_t$ are fixed, and in terms of $\boldsymbol{\beta}_t$ if $\mathbf{c}$ is fixed. For such a loss function, the global convergence holds.

Particularly, Problem (4) is equivalent to Problem (9) which is a convex program for any convex loss function, so any local optimal solution obtained by Algorithm $\mathcal{A}$ is also a global minimizer of (4). In our experiments, we implement Algorithm $\mathcal{A}$ with the logistic regression loss and the least squares loss with the 2-norm regularization for P_1 and P_2, and thus the algorithm globally converges.

6 Experiments

We validate the proposed collaborative learning approach by comparing it to standard single-task learning (STL) approaches where multiple tasks are tackled independently, and to two commonly-used multi-task learning (MTL) methods, a regularization-based MTL method in [15] and a Bayesian MTL method based on Gaussian processes (CGP) in [5]. Notice that Algorithm $\mathcal{A}$ with the 1-norm penalty term for both P_1 and P_2 in Problem (3) has been implemented in our early work [12] where a pooling method which trains a single model for all tasks has also been compared. Readers can consult [12] for corresponding results.

6.1 Synthetic Data

We generated synthetic data to verify the behavior of the developed algorithms regarding the selected features and the performance in comparison with single-task learning (STL) logistic regression. The synthetic data was generated as follows:

- generate $\mathbf{x} \in R^{10}$ with each component $x_i \sim \mathbf{Uniform}[-1, 1]$;
- set $T = 3$ and the coefficient vectors of the 3 tasks to

$$\begin{aligned} \alpha_1 &= [1, 1, 1, 0, 0, 0, 0, 0, 0, 0], \\ \alpha_2 &= [0, 1, 1, 1, 0, 0, 0, 0, 0, 0], \\ \alpha_3 &= [0, 0, 1, 1, 1, 0, 0, 0, 0, 0]; \end{aligned}$$

- $y = \text{sign}(\alpha_t^\top \mathbf{x})$ for every sample $\mathbf{x}$ of task t.

For each task, we generated training sets of sizes $\ell = [20, 40, 60, 80]$, each used in a different trial, 150 samples for tuning and 1000 samples for testing, and repeated each trial 20 times. The misclassification rates averaged over the

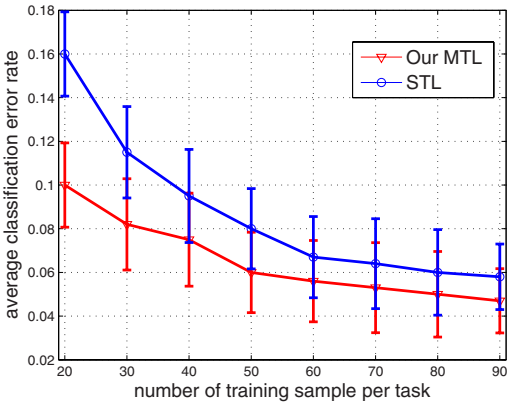

Fig. 3. Performance on synthetic data: error rates versus training sample sizes

3 tasks and 20 runs are shown in Fig. 3 for different training sample sizes, respectively by our approach and STL. Fig. 3 obviously shows the superiority of our approach. As expected, the difference of the two approaches becomes smaller as the sample size of each task becomes larger.

We show bar plots of the averaged estimated coefficient vectors by our approach in Fig. 4-left and the STL logistic regression in Fig. 4-right. Our approach successfully removed irrelevant features. For lucid presentation, each coefficient vector was normalized by its norm, averaged over all trials, and shown on Fig. 4. Although STL produced reasonable classifiers for each task, it could not delete all irrelevant features using data available for each single task.

6.2 Lung Cancer Data

A prototype version of our lungCAD system [16] (not commercially available) was applied on a proprietary anonymized patient data set collected from multiple

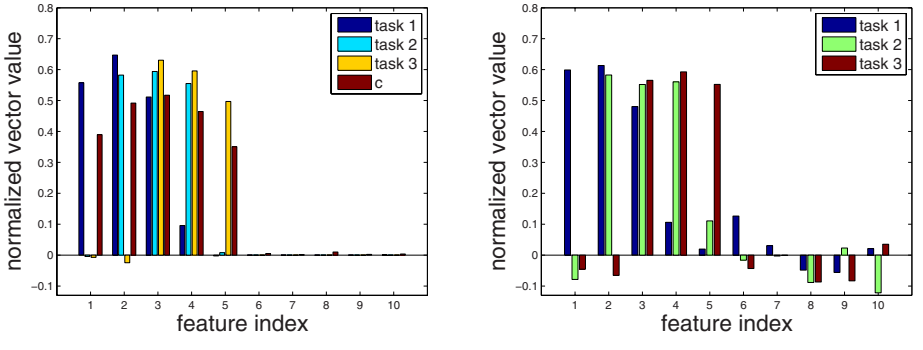

Fig. 4. Performance on synthetic data: left, coefficient vectors by our MTL; right, coefficient vectors by STL logistic regression

hospitals. The nodule dataset consisted of 176 CT images that were randomly partitioned into two groups by a third party agency for development and evaluation respectively: 90 volumes in training and 86 volumes in test. The GGO dataset consisted of 60 CT images. Due to the limited size of GGO set, GGO detection performance could not be measured reliably, so GGO cases were used only for improving nodule detection performance. In total, 129 nodules and 53 GGOs were labeled by radiologists, and 81 nodules appeared in the training set and 48 in the test set. The lungCAD system was independently applied to the training, test nodule sets and the GGO set, generating totally 11056, 13985 and 10265 suspicious candidates in the respective sets. Among them, 131, 81 and 87 candidates were true detections associated with nodules or GGOs. A total of 86 numerical image features were calculated. The statistics of the lungCAD data set is characterized in Table 1.

Table 1. Specifications of lungCAD data sets

	Nodule train	Nodule test	GGO
No. patients	90	86	60
No. candidates	11056	13985	10265
No. cancer	81	48	53
No. positives	131	81	87
No. False Positives /vol	121	161	169
No. feature	86	86	86

The first set of experiments were conducted as follows. We randomly sampled 50% (45 volumes) of the nodule training cases and 50% (30 volumes) of the GGO cases to train a classifier that was tested on 86 test nodule cases, and repeated 15 trials. In the first trial, we tuned the model parameter γ in Algorithm $\mathcal{A}$ and the regularized parameters in [15] according to a 3-fold cross validation performance within training, and we fixed them for other trials. Fig. 5 shows the test ROC curves averaged over the 15 trials with error variance bars. Our algorithm $\mathcal{A}$ produces a curve that dominates the ROC curves corresponding to other approaches. It also had a relatively small model variance by referencing the error bars. The regularized MTL and CGP were superior to STL learning, inferior to our method, and the regularized MTL also presented a relatively large error variance as shown by the error bars.

We conducted more complete performance comparisons using the AUC measure by randomly sampling p% of training nodule cases with a fixed amount of GGO cases where $p = 10,\ 25,\ 50,\ 75,\ 100$, and 15 trials were performed for each p. We averaged the AUC numbers over 15 trials for each value of p, and illustrated them in Fig. 6 together with error bars. The resulting models achieved better performance with less help from related tasks when more samples of the nodule detection task were used.

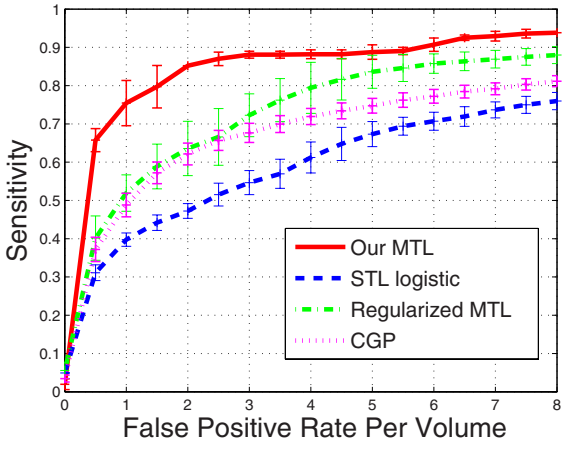

Fig. 5. On Lung Cancer Data: test ROC plots of models trained using 50% of nodule and GGO training patient volumes

6.3 Heart Wall Motion Data

Cardiac wall motion data has a different structure from lung cancer data (in lungCAD data, different patient data were provided for different tasks). Here we collected 220 ultrasound images of patients hearts, and 432 image features were extracted from the left ventricle of each heart to characterize the global motion and segment-level wall motion of the LV. Hence each heart was represented by a feature vector of 432 components. Overall 16 labels, one for each segment, were provided to a single feature vector. Hence the same set of patients were provided

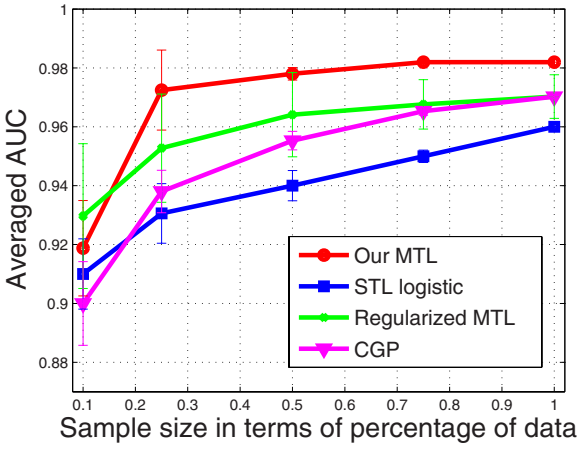

Fig. 6. On Lung Cancer Data: the plot of averaged AUC versus training sample size

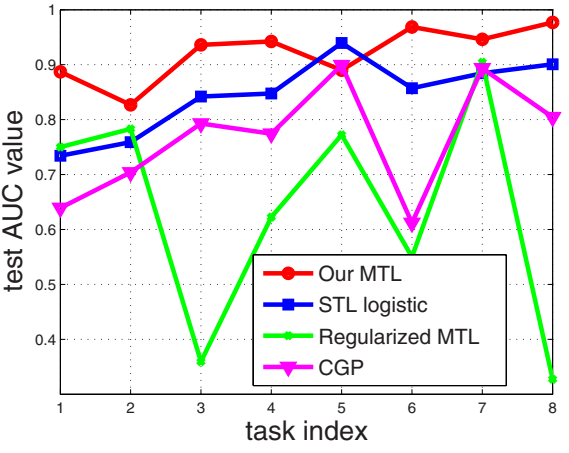

Fig. 7. On Heart Data: the plot of test AUC versus task index

for the different 16 tasks. This is sometimes referred to as multi-label prediction problems.

Although great efforts were made to collect a reasonable number of abnormal studies, the normal versus abnormal segment-level distribution was extremely unbalanced since most patients only have one or two abnormal segments. Many of the segments had fewer than 3 abnormal cases. Only 8 segments (out of 16 segments), for which enough abnormal cases (25 cases on average) were present in the 220 cases, were used in our experiments.

The 220 cases were randomly split 15 times into two sets of an equal size, one for training and one for test. We tuned model parameters such as γ using a 2-fold

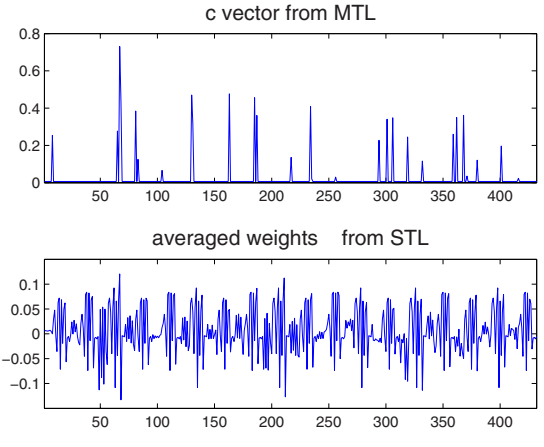

Fig. 8. On Heart Data: comparison of the selected features by our MTL and STL

cross validation within the training set. The average test AUC values for each of the 8 tasks are depicted in Figure 7 which clearly shows the effectiveness of our MTL approach. The regularized MTL and CGP were not originally proposed for sparse estimation, which may result in poor performance on data where dimension is much larger than the available sample size, such as the heart data.

The $\mathbf{c}$ vector learned by our approach was very sparse as shown in Fig. 8 which shows only 21 features were used by all the 8 classifiers combined. Notice that each classifier only chooses features from the features selected by $\mathbf{c}$. Whereas, STL logistic regression used much more features as the averaged weight vector $\boldsymbol{\alpha}$ in Fig. 8 was dense.

7 Conclusions

We have designed a series of approaches to learning multiple tasks jointly. Efficient algorithms have been developed through alternating optimization to find the optimal solutions of these approaches. Convergence analysis shows that the algorithms globally converge for strictly convex loss functions and regularization conditions. Our framework also provides a probabilistic interpretation for existing regularized multi-task learning methods. Although the proposed algorithms are general enough to be applied to any multi-task setting, they are motivated by the challenges of the real-world medical diagnostic problems. Computational results of the proposed approach on medical diagnostic problems demonstrate superiority to some early multi-task learning approaches. The proposed approach has been deployed in our lungCAD system which has received clinical approval from Food and Drug Administration. Possible extension of this work includes the examination of general feature representation without the independence assumption among features and the related algorithm design.

References

1. Roehrig, J.: The promise of CAD in digital mamography. European Journal of Radiology 31, 35–39 (1999)
2. Armato-III, S.G., Giger, M.L., MacMahon, H.: Automated detection of lung nodules in CT scans: preliminary results. Medical Physics 28(8), 1552–1561 (2001)
3. Suzuki, K., Kusumoto, M., Watanabe, S., Tsuchiya, R., Asamura, H.: Radiologic classfication of small adenocarcinoma of the lung: Radiologic-pathologic correlation and its prognostic impact. The Annals of Thoracic Surgery CME Program 81, 413–420 (2006)
4. Caruana, R.: Multitask learning. Machine Learning 28(1), 41–75 (1997)
5. Yu, K., Tresp, V., Schwaighofer, A.: Learning Gaussian processes from multiple tasks. In: ICML 2005 (2005)
6. Ando, R.K., Zhang, T.: A framework for learning predictive structures from multiple tasks and unlabeled data. Journal of Machine Learning Research 6, 1855–1887 (2005)
7. Heskes, T.: Empirical Bayes for learning to learn. In: Proc. 17th International Conf. on Machine Learning, pp. 367–374. Morgan Kaufmann, San Francisco (2000)

8. Jebara, T.: Multi-task feature and kernel selection for SVMs. In: Proceedings of the 21st International Conference on Machine learning (2004)
9. Obozinski, G., Taskar, B., Jordan, M.I.: Multi-task feature selection. Technical report, UC Berkeley (2006)
10. Argyriou, A., Evgeniou, T., Pontil, M.: Multi-task feature learning. In: Schölkopf, B., Platt, J., Hoffman, T. (eds.) Advances in Neural Information Processing Systems, vol. 19, pp. 41–48. MIT Press, Cambridge (2007)
11. Srebro, N., Rennie, J.D.M., Jaakola, T.S.: Maximum-margin matrix factorization. In: NIPS 2005 (2005)
12. Xiong, T., Bi, J., Rao, R.B., Cherkassky, V.: Probabilistic joint feature selection for multi-task learning. In: SIAM International Conference on Data Mining (2006)
13. Bezdek, J.C., Hathaway, R.J.: Convergence of alternating optimization. Neural, Parallel Sci. Comput. 11, 351–368 (2003)
14. Bezdek, J.C., Hathaway, R.J.: Some notes on alternating optimization. In: Pal, N.R., Sugeno, M. (eds.) AFSS 2002. LNCS (LNAI), vol. 2275, pp. 288–300. Springer, Heidelberg (2002)
15. Evegniou, T., Pontil, M.: Regularized multi–task learning. In: Proc. of 17–th SIGKDD Conf. on Knowledge Discovery and Data Mining (2004)
16. Rao, R.B., Bi, J., Fung, G., Salganicoff, M., Obuchowski, N., Naidich, D.: LungCAD: a clinically approved, machine learning system for lung cancer detection. In: ACM International Conference on Knowledge Discovery and Data Mining (2007)