Sparsity Models for Multi-Task Learning

Jian Zhang
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213, USA

December 1, 2005

Abstract

Many real world machine learning problems can be recast as multi-task learning problems, whose objective is to utilize the relations among those tasks in order to obtain a better generalization performance than learning them individually. In this paper we present two probabilistic models for solving multi-task learning problems which have a sparsity underlying assumption. In particular, our models are special cases of hierarchical Bayesian models which associate the generation of task parameters of each prediction function with a set of latent variables. By exploring different statistical assumptions of distributions of latent variables and the linear mixing matrix, we are able to achieve two types of sparsities: (1) each prediction function is a sparse linear combination of a set of basis functions; (2) each prediction function is a linear combination of a set of basis functions which are sparse themselves. In this paper we focus on the second type of sparsity models. Experiments on multi-labeled text classification demonstrate the effectiveness of the proposed models over the traditional single task learning approach.

1 Introduction

The traditional supervised learning problem tries to estimate a function $f : \mathcal{X} \mapsto \mathcal{Y}$, where $\mathcal{X}$ is the input space and $\mathcal{Y} = \mathbb{R}$ for regression or $\mathcal{Y} = \{C_1, C_2, \ldots, C_M\}$ for classification, given a training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$. Given K tasks, the objective of multi-task learning is to estimate K prediction functions $f^{(1)}, f^{(2)}, \ldots, f^{(K)}$ jointly (as opposed to individually) so that a better generalization performance can be achieved compared to learning each task independently. It is often the case that each prediction function $f^{(k)} : \mathcal{X} \mapsto \mathcal{Y}^{(k)}$, e.g. they share the same input space $\mathcal{X}$ but not necessarily the output space. Multi-task learning has been both an relatively old [7, 5] and new research topic [9, 8, 2, 6, 1]. More or less, multi-task learning approaches are based on the assumption that tasks are related in some way so that information can be "borrowed". The necessity of this assumption is also easy to be seen as it is unlikely to gain by learning jointly from K totally irrelevant tasks. Therefore, it is natural to explore different assumptions about how tasks are related, and design corresponding models which are suitable for those multi-task learning "scenarios".

There are many potential interesting applications of multi-task learning. For example, in text classification it is often the case that each document can belong to multiple categories, which is also known as "multi-label" text classification. Due to the relatedness among those categories, we can treat the classification problem with respect to each category as a task and formulate a multi-task learning problem. Similarly, we can also formulate the multiple-user anti-spam email filtering as a multi-task learning problem where each task is the anti-spam email filtering problem with respect to a particular user. Multi-task learning approaches are advantages for this problem as those individual tasks are different but very closely related. Other possible applications include predicting many stock prices, conjoint analysis, etc.

In this paper we present a new approach for sparse formulations of multi-task learning models. Our models are based on a clean, well-motivated latent variable generative model, in which tasks parameters are assumed to be generated from a linear mixing of a set of latent variables plus some random noise. In particular, we can achieve two types of sparsities within this formulation, by imposing different statistical assumptions on the model: (1) each resulting classifier is a sparse linear combination of some basis classifiers; (2) each resulting classifier is a linear combination of a set of basis classifiers which are sparse. In this paper we will focus on the latter as the former is already presented in detail in our previous work [9].

2 Probabilistic Models

Suppose that we use θ_k to index the prediction function for the k-th task, and in this paper we limit our discussion to linear methods (e.g. $f^{(k)}_{\theta_k}(\mathbf{x}) = \theta_k^T\mathbf{x}$) since the generalization to non-linear case is straightforward. We assume the following generative model for the θ_k's [10]:

$$\begin{aligned} \theta_k &= \Lambda\mathbf{s}_k + \mathbf{e}_k \\ \mathbf{s}_1, \ldots, \mathbf{s}_K &\sim p(\mathbf{s}_1, \ldots, \mathbf{s}_K|\Phi) \\ \mathbf{e}_k &\sim \mathcal{N}(\mathbf{0}, \Psi) \end{aligned} \tag{1}$$

where $\mathbf{s}_k \in \mathbb{R}^{H\times 1}$ $(k = 1, 2, \ldots, K)$ are latent variables which follow a distribution parametrized by Φ; $\Lambda \in \mathbb{R}^{F\times H}$ is a linear transformation matrix on $\mathbf{s}_k$'s; $\mathbf{e}_k \in \mathbb{R}^{F\times 1}$ is usually assumed to be Gaussian random noise. Note that θ_k is composed of two parts: the common, sharing component $\Lambda\mathbf{s}_k$ and the task specific component $\mathbf{e}_k$. This is important since it allows to have a good generalization power when the number of training examples per task goes to infinity, in which case we would like to give each task enough freedom to grow respectively. Also notice that when $\Lambda\mathbf{s}_k = \mathbf{0}$, this framework degenerates trivially to the traditional model for single task learning (such as logistic regression). We can fully specify the generative model for multi-task learning by assuming the logistic regression as the classification model (or more generally any suitable generalized linear model [4])

$$y \sim \mathcal{B}(\sigma(\theta_k^T\mathbf{x}))$$

where $\mathcal{B}(.)$ denotes Bernoulli distribution, and $\sigma(t) = (1 + \exp(-t))^{-1}$ is the logistic function. There are at least two possible ways to achieve sparsity models based on the above generative model framework:

1. Assume a sparse prior for $\mathbf{s}_k$'s such as Laplace, e.g.

$$p(\mathbf{s}_k) \propto \prod_{h=1}^{H} \exp(-|s_{k,h}|).$$

   This essentially is assuming that each target classifier is a sparse linear combination of basis classifiers, and details for this model can be found in [9].

2. Instead of assuming Λ to be fixed as in equation (1), we would assume it to be random such that

$$p(\Lambda_{.j}) \propto \prod_{f=1}^{F} \exp(-|\Lambda_{f,j}|),$$

   where $\Lambda_{.j}$ denotes the j^{th} column of matrix Λ. That is, we assume that each column vector of Λ follows a sparse prior distribution such as Laplace. By performing a point estimation $\hat{\Lambda}$, this model will lead to a set of basis classifiers (the set of column vectors of Λ can be thought as basis classifiers) that are sparse.

After specifying the probabilistic model, we can apply either empirical Bayes approach or point estimation approach to learn model parameters. In either case, the basic intuition is to realize that we only need to

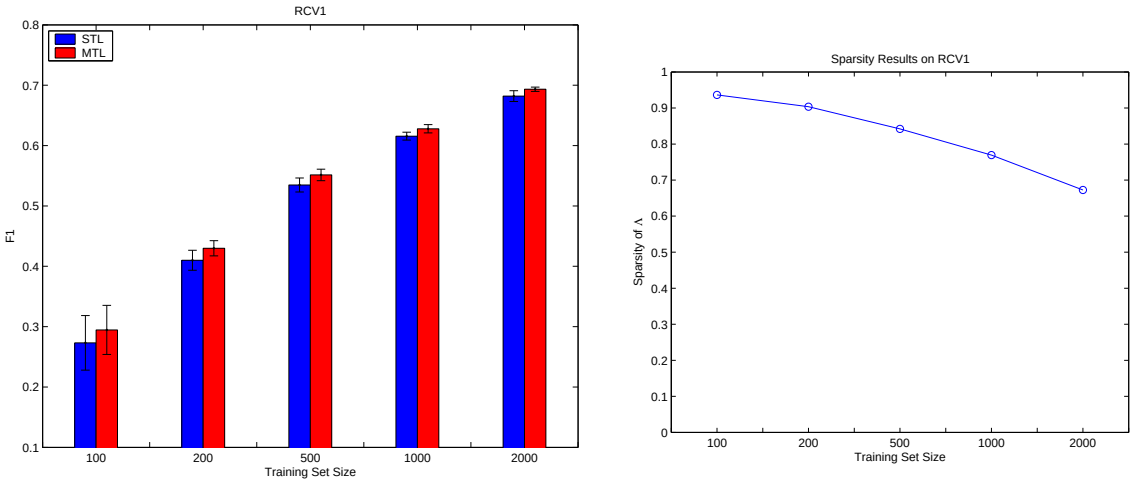

Figure 1: LEFT: Text classification results on RCV1; RIGHT: Average sparsity rate of elements in $\hat{\Lambda}$.

iteratively estimate θ_k's and $\mathbf{s}_k$'s for each task and then Λ and Ψ for all tasks. Here we focus on point estimation method which is more efficnet for dealing with high-dimensional data like text. However, point estimation for the covariance matrix Ψ of $\mathbf{e}_k$'s is not well-behaved as it goes to $\mathbf{0}$ and thus we cannot get a sensible estimation $\hat{\Psi}$. Instead, we restrict the form of $\Psi = \lambda \mathbf{I}$ to be diagonal and isotropic in our algorithm, and use cross-validation as an outer loop to tune the scalar parameter λ just as in traditional single task learning problems. Procedurely

1. Update the estimates $\hat{\theta}_k$ and $\hat{\mathbf{s}}_k$ given $\hat{\Lambda}$ computed in the previous step (conditioned on Λ and Ψ, task parameters will decouple and we can conduct this step per task):

$$\{\hat{\theta_k}, \hat{\mathbf{s}}_k\} = \arg\max_{\theta_k, \mathbf{s}_k} \left\{ \frac{1}{n_k} \sum_{i=1}^{n_k} L(y_i^{(k)}, \theta_k^T \mathbf{x}_i^{(k)}) + \log \mathcal{N}(\theta_k \mid \hat{\Lambda}\mathbf{s}_k, \Psi) \right\}, \quad k = 1, 2, \ldots, K$$

   This is essentially equivalent to the regularized linear methods for classification, and we can apply any suitable optimization algorithm to solve it, such as conjugate gradient or quasi-Newton method.

2. Given the updated $\hat{\theta}_k$'s and $\hat{\mathbf{s}}_k$'s:

$$\hat{\Lambda} = \arg\max_{\Lambda} \left\{ \sum_{k=1}^{K} \log \mathcal{N}(\hat{\theta}_k \mid \Lambda \hat{\mathbf{s}}_k, \Psi) + \log p(\Lambda) \right\}$$

   Plugging in the prior of Λ, it leads to a set of Lasso-style problems:

$$\hat{\Lambda} = \arg\min_{\Lambda} \left\{ \sum_{k=1}^{K} (\hat{\theta}_k - \Lambda \hat{\mathbf{s}}_k)^T (\hat{\theta}_k - \Lambda \hat{\mathbf{s}}_k) + \gamma \sum_{h=1}^{H} \sum_{f=1}^{F} |\Lambda_{f,h}| \right\}$$

   where γ controls how sparse the solution $\hat{\Lambda}$ is.

3 Experimental Results

To demonstrate the effectiveness of the model, we conducted experiments on multi-label text classification tasks. Here we use the RCV1 dataset [3], which is one of the new benchmark collection for text classification. The TOPIC code hierarchy contains more than one hundred categories (H, the dimensionality of

latent variable, is set to 10 in all our experiments), and we treat the classification problem with respect to each category as a task. Since multi-task learning is mostly effective when the number of training examples per task is small (as can be seen from the fact that MLE is asymptotically optimal), we conduct experiments by varying the number of training examples. We randomly select 10k test documents as our testset, and classification results are evaluated using the $F1$ measure, which is the typically used evaluation measure for text classification. The classification loss function we used here is the logistic loss, and we compare our algorithm (MTL) with the standard regularized logistic regression for single task learning (STL). Regularization parameter of STL was chosen through cross validation, and for MTL it was chosen to match the prior variance of θ_k's. Results are shown in the left graph of Figure 1, from which we can see that our MTL model is more effective than the corresponding STL algorithm in terms of prediction accuracy.

Furthermore, our model also has the sparsity property[1]. In some sense, sparsity reflects the degree of freedom of the fitted model and thus measures the model complexity. Although the actual answer depends on the number of training examples, investigation on the average number of non-zero elements of $\hat{\Lambda}$ does suggest that we achieved a sparse solution, as also shown in Figure 1.

4 Concluding Remarks

In this paper we present a probabilistic framework which can be used for a variety of multi-task learning scenarios [10], and focus on models which can lead to sparse solutions. We conducted experiments on multi-label text classification, and results show the advantages of the proposed models over single task learning methods. In the future we would like to investigate more flexible models for different multi-task learning scenarios, as well as a systematic way of automatically choosing the dimension H of the latent variable $\mathbf{s}_k$. Furthermore, we would like to explore more interesting applications of various multi-task scenarios.

References

[1] R. Ando and T. Zhang. A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data. Technical Report RC23462, IBM T.J. Watson Research Center, 2004.

[2] T. Evgeniou and M. Pontil. Regularized Multitask Learning. Proc. of 17th SIGKDD Conference on Knowledge Discovery and Data Mining, 2004.

[3] D. Lewis, Y. Yang, T. Rose and F. Li. RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research 5:361-397, 2004.

[4] P. McCullagh and J. A. Nelder. Generalized Linear Models (2nd edition), Chapman & Hall/CRC, 1989.

[5] D. L. Silver. The Selective Transfer of Neural Network Task Knowledge. Ph.D. Thesis, The University of Western Ontario, London, Ontario, Canada.

[6] T. W. Teh, M. Seeger and M. I. Jordan. Semiparametric Latent Factor Models. AISTATS 2005.

[7] S. Thrun and L. Pratt (eds). Learning to Learn, Kluwer Academic Publishers, 1998.

[8] K. Yu, V. Tresp and A. Schwaighofer. Learning Gaussian Processes from Multiple Tasks. In Proceedings of 22nd ICML, Bonn, Germany, 2005.

[9] J. Zhang, Z. Ghahramani and Y. Yang. Learning Multiple Related Tasks using Latent Independent Component Analysis. NIPS 2005.

[10] J. Zhang. A Probabilistic Framework for Multi-Task Learning. Ph.D. Thesis Proposal, Carnegie Mellon University.

[1] In our experiments γ was set so that half of the prior variance of θ_k comes from $\Lambda\mathbf{s}_k$ and half comes from $\mathbf{e}_k$.