ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-255

Nonparametric Bayesian Multi-Task Large-margin Classification

Changying Du[1,2], Jia He[1], Fuzhen Zhuang[1], Yuan Qi[3], Qing He[1]

Abstract. In this paper, we present a nonparametric Bayesian multi-task large-margin classification model which can cluster tasks into the most appropriate number of groups and induce flexible model sharing within each task group simultaneously. Specifically, we first show a very simple method to integrate large margin learning with hierarchical Bayesian models by employing an important variant of the standard SVM, i.e., proximal SVM (PSVM), whose loss function is used to define a novel likelihood function. And then we assume that the model parameter of each task consists of two parts: one is shared within each task group (group-level parameter) while the other is specific to each distinct task (task rescaling parameter). A Dirichlet process prior is imposed on the group-level parameter while the task rescaling parameter is assigned a one-mean Laplace prior. Finally the parameter of a task is the corresponding group parameter times its specific rescaling parameter. We give efficient Markov chain Monte Calo (MCMC) algorithm to conduct model inference. Experiments on the Landmine detection data and the UCI Yeast data demonstrate the effectiveness of our method.

1 INTRODUCTION

Machine learning lies in the heart of artificial intelligence, and has been extensively studied during the past decades. While traditional machine learning is approaching to its potential performance limit, a new learning scenario called multitask learning (MTL) [6] has attracted more and more attention in the community of machine learning and data mining [25, 2, 7, 26, 8, 15, 9, 21]. Multitask learning learns multiple related tasks together so as to improve the performance of each task relative to learning them separately. Over the past decade, MTL has been successfully applied to many important areas including computer vision [24, 15], natural language processing [1], bioinformatics [20, 26] and landmine detection [25, 14].

It has been shown that the performance boosting merit of MTL is mainly due to its information sharing among tasks, which is the key aspect in the design of MTL algorithms. To uncover latent task structure and alleviate harmful information sharing, task-grouping is a common practice in MTL [3, 25, 15, 16, 19]. Existing methods typically assume tasks in the same cluster share the same model [3, 25], though it is more reasonable to allow some flexibility in each task group. Meanwhile, large-margin classification models such as SVMs stand for the most popular classification models in traditional learning scenarios, but there are still not many successful multi-task large-margin classification models, especially those with the capability to find latent task groups automatically.

In this paper, we present a nonparametric Bayesian multitask large-margin classification model which can cluster tasks into the most appropriate number of groups and induce flexible model sharing within each group simultaneously. Specifically, we first show a very simple method to integrate large margin learning with hierarchical Bayesian models by employing an important variant of the standard SVM, i.e., proximal SVM (PSVM) [11], whose empirical loss function can be used to define a novel likelihood function. And then we assume that the model parameter of each task consists of two parts: one is shared within each task group (group-level parameter) while the other is specific to each distinct task (task rescaling parameter). A Dirichlet process (DP) [10, 23] prior is imposed on the group-level parameter while each dimension of the task rescaling parameter is assumed to have a one-mean Laplace prior. Due to the nonparametric clustering nature of DP, we can automatically cluster the tasks into separate groups without pre-specifying the group number, which is hard to determine in advance. In each group all tasks share the same group-level parameter while each task has its own small task-specific rescaling over the group parameter. By imposing a one-mean laplace prior, the rescaling is sparse, and finally the parameter of a task is the group parameter times its specific rescaling parameter. This corresponds to that in each task group, for most dimensions the multidimensional models are identical, but for special ones they may differ from each other, which is a flexible model sharing scheme.

We give efficient Markov chain Monte Calo (MCMC) algorithm to conduct model inference. Experiments on the Landmine detection data set and the UCI Yeast data set demonstrate our method can not only outperform state-of-the-art MTL algorithms but also discover the task-clustering structure very well.

The remainder is organized as follows. Section 2 briefly covers the necessary preliminaries. Then in Section 3 we propose our nonparametric Bayesian multi-task large-margin classification model by first defining a novel likelihood function. The experimental results are demonstrated in Section 4 and related works are given in Section 5. Finally we conclude the paper in Section 6.

[1] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China, email: ducy@ics.ict.ac.cn
[3] Departments of CS and Statistics, Purdue University, IN, USA

2 PRELIMINARIES

2.1 Dirichlet Process

Ferguson [10] first introduced the Dirichlet process (DP), which is a distribution over distributions and widely used in Bayesian nonparametric models of data. Sethuraman [22] explicitly showed that measures drawn from a Dirichlet process are discrete with probability one in the stick-breaking representation of DP, that is, the random measure G distributed according to a Dirichlet process $\mathrm{DP}(\alpha, G_0)$ with base distribution G_0 and concentration parameter α can be written as

$$G = \sum_{i=1}^{\infty} \pi_i(\mathbf{v})\delta_{\phi_i},\ \pi_i(\mathbf{v}) = v_i \prod_{j=1}^{i-1}(1-v_j)$$

where $\phi_i \sim G_0$, $v_i \sim \mathrm{Beta}(1,\alpha)$, and δ_ϕ is an atom at ϕ. It is clear from this formulation that the support of G consists of a countably infinite set of atoms, which are drawn independently from G_0.

The Pólya urn scheme due to Blackwell and MacQueen [4] shows that not only are draws from the Dirichlet process discrete, but also they exhibit a clustering property. Let $\theta_1, \theta_2, \cdots$ be a sequence of i.i.d. random variables distributed according to G. That is, the variables $\theta_1, \theta_2, \cdots$ are conditionally independent given G, and hence exchangeable. Blackwell and MacQueen showed that the successive conditional distributions of θ_i given $\theta_1, \cdots, \theta_{i-1}$ have the following simple form:

$$\theta_i|\theta_1, \cdots, \theta_{i-1}, \alpha, G_0 \sim \sum_{l=1}^{i-1} \frac{1}{i-1+\alpha}\delta_{\theta_l} + \frac{\alpha}{i-1+\alpha}G_0 \quad (1)$$

where G has been integrated out, and δ_θ represents an atom at θ.

In a DP model, only a few of the countably infinite set of atoms will dominate in the posterior, and the actual number of clusters used to model data can be automatically inferred from data using popular Bayesian inference framework, e.g., MCMC [17] and Variational inference [5].

2.2 The Proximal SVM Classifier

Consider the 2-class classification problem of classifying N points in d-dimensional real space R^d, represented by the $N \times d$ matrix A. A diagonal matrix D with $+1$ or -1 along its diagonal specifies whether the membership of each point A_i in the class $A+$ or $A-$.

For the classification problem stated above, the PSVM with a linear kernel [11] tries to find the proximal planes: $x'w - r = \pm 1$ where w, r are the orientation and the relative location to the origin respectively. And it can be formulated by the following quadratic program with equality constraints and the regularization parameter ν:

$$\begin{aligned} \min_{(w,r,y)\in R^{d+1+N}} \ & \frac{\nu}{2}\|y\|^2 + \frac{1}{2}(w'w + r^2) \\ s.t. \quad & D(Aw - er) + y = e \end{aligned} \quad (2)$$

where y and e are N-dimensional column vectors and each element of e is equal to one. The resulting separating plane acts like below:

$$x'w - r \begin{cases} > 0, & \text{then } x \in A+, \\ < 0, & \text{then } x \in A-, \\ = 0, & \text{then } x \in A+ \text{ or } x \in A- . \end{cases} \quad (3)$$

3 THE MODEL

3.1 Proximal SVM and a Novel Likelihood Function

PSVM aims to minimize the loss function $L(A;\ w, r) = \frac{\nu}{2}\|D(Aw - er) - e\|^2 + \frac{1}{2}(w'w + r^2)$, where the first term is the empirical loss on training data A while the second being structure loss. Let $\widetilde{w} = (w, r)$ denotes model parameter and $L_1(A;\ \widetilde{w}) = \|D(Aw - er) - e\|^2$ denotes the empirical loss on training data A, then we define the likelihood of parameter $\widetilde{w}$ on A as[4]:

$$F(A;\ \widetilde{w}) = exp(-L_1(A;\ \widetilde{w})) = exp(-\|D(Aw - er) - e\|^2) \quad (4)$$

Note that the new likelihood function is directly defined on the entire training data set rather than on a single point. By imposing prior distributions such as Gaussian or Laplacian on the model parameter $\widetilde{w}$, we can convert the regularized optimization formulation in PSVM into a Bayesian inference problem. Under Bayesian framework, many useful prior distributions can be incorporated to fit the learning scenarios, e.g., a Dirichlet process prior can cluster the model parameters of multiple learning tasks.

3.2 Bayesian Multi-Task large-margin Classification

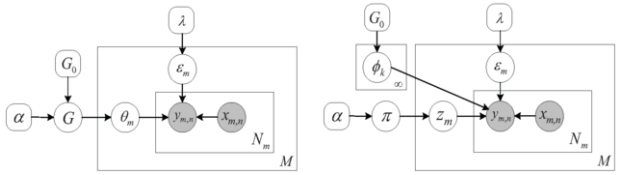
In the multitask learning scenario, we assume that the PSVM model parameter $\widetilde{w}$ of each task consist of two parts, i.e., the group-level parameter θ and the task rescaling parameter ε, and we have $\widetilde{w} = \theta \circ \varepsilon$, where $\circ$ is the Hadamard product (or element-wise product). Note a group-level parameter is shared within its corresponding task group while a task rescaling parameter is specific to its corresponding distinct task.

Given M learning tasks, we assume their group-level parameters $\theta_i,\ i = 1, ..., M$ are drawn independently from a Dirichlet process: $\theta_i|G \sim G,\ G|\alpha, G_0 \sim \mathrm{DP}(\alpha, G_0)$, where the base distribution G_0 is a zero-mean multi-variate Gaussian distribution $N(0, \Sigma)$, and the concentration parameter α is a positive real number. Each dimension of $\varepsilon_i,\ i = 1, ..., M$ is drawn independently from a one-mean Laplace distribution: $\varepsilon_{ik}|\lambda \sim \mathrm{Laplace}(1, \lambda)$, where the scale parameter λ is a positive real number, and $k = 1, ..., d+1$ is the dimension index[5]. Then given the training data set $\mathcal{D}_i = \{(x_{i,n}, y_{i,n})\}_{n=1}^{N_i}$ and the model parameters θ_i and ε_i in the i-th task, we have the likelihood of the i-th task as follows:

$$F(\mathcal{D}_i;\ \theta_i, \varepsilon_i) = exp(-L_1(\mathcal{D}_i;\ \theta_i \circ \varepsilon_i)), \quad (5)$$

with which we can integrate the large-margin principle into our Bayesian Model. Due to the nonparametric clustering nature of DP, we can automatically cluster the tasks into separate groups without pre-specifying the group number, which is hard to determine in advance. In each group all tasks share the same group-level parameter while each task has its own small task-specific rescaling over the group parameter. With the one-mean laplace prior, the rescaling is sparse, and finally

[4] Note, a likelihood function is not necessarily a probability density.
[5] The parameter $\widetilde{w}$ in PSVM has $d+1$ dimensions for d-dimensional data.



(a) (b)

Figure 1. (a) Graphical representation of the nonparametric Bayesian Multi-Task Large-margin Classification model (dpMTLC); (b) An equivalent representation of dpMTLC model in terms of the stick-breaking construction.

the parameter of a task is the group parameter times its specific rescaling parameter. This correspond to in each group all tasks have (almost) the same model coefficients on most features but differ from each other on some special ones, which is a flexible model sharing scheme.

The graphical model of our nonparametric Bayesian Multi-Task Large-margin Classification model (dpMTLC) is shown in Figure 1 (a). Note that if we set the Laplace scale parameter λ to a small value that is very close to zero, then dpMTLC reduces to a model without flexible model differences in each task group, which we will refer to as dpMTLC0 in the sequel. To facilitate model inference, we have re-represented dpMTLC using the stick-breaking construction method of DP, as is shown in Figure 1 (b), where ϕ_k is drawn from the base distribution G_0, and $z_m \sim \pi(\mathbf{v})$, where $\pi_i(\mathbf{v}) = v_i \prod_{j=1}^{i-1}(1-v_j),\ v_i|\alpha \sim \text{Beta}(1,\alpha),\ i = 1, ..., \infty$. Note that $z_m,\ i = 1, ..., M$ also have the exchangeability similar as that of $\theta_i,\ i = 1, ..., M$ according to [22], which is important in our model inference. Though we only considered linear classification here, it is easy to extend dpMTLC to nonlinear case using the kernel methods similar as in PSVM [11].

3.3 Model Inference

We are interested in the posterior distribution of latent variables θ and ε, however, exact inference by analytically computing their posterior is intractable. A popular kind of approximate posterior inference methods is the Markov chain Monte Calo (MCMC) sampling, where a Markov chain is constructed to converge to the target posterior distribution, and samples are then taken from that Markov chain. Specifically, here we employ Gibbs sampling which reaches the next state of a Markov chain by sequentially sampling all variables from their conditional distributions based on the current values of all other variables and the data. Thus in each MCMC iteration we alternately sample θ, ε and hyperparameters as follows:

Sample θ. Conditioned on ε and hyperparameters, we could use Gibbs sampling for θ if G_0 is the conjugate prior for the likelihood given by F. That is, we would repeatedly draw samples from $\theta_i|\theta_{-i}, \mathcal{D}_i$ (where $i = 1, ..., M$) using the conditional distribution (1) and the likelihood (6). Neal [17] presented several algorithms for sampling from the posterior distribution of Dirichlet process mixtures when non-conjugate priors are used. Here, we use Gibbs sampling with auxiliary parameters (Neal's algorithm 8). This algorithm uses the stick-breaking representation of DP and constructs a Markov chain whose state consists of $z_1, ..., z_M$ and $\phi = (\phi_z : z \in \{z_1, ..., z_M\})$, so that $\theta_i = \phi_{z_i}$. To allow the creation of new clusters, it temporarily supplements the ϕ_z parameters of existing clusters with m (or $m-1$) additional parameter values drawn from the prior. Each iteration of the Markov chain simulation operates as follows:

- For $i = 1, ..., M$: Let k^- be the number of distinct z_j for $j \neq i$ and let $h = k^- + m$. Label these z_j with values in $\{1, ..., k^-\}$. If $z_i = z_j$ for some $j \neq i$, draw values independently from G_0 for those $\phi_z,\ k^- < z \leq h$. If $z_i \neq z_j$ for all $j \neq i$, let $z_{k^-+1} = z_i$, and draw values independently from G_0 for those $\phi_z,\ k^- + 1 < z \leq h$. Draw a new value for z_i from $\{1, ..., h\}$ according to:

$$P(z_i = z|z_{-i}, \mathcal{D}_i, \phi_1, ..., \phi_h) = \begin{cases} b\frac{n_{-i,c}}{M-1+\alpha}F(\mathcal{D}_i;\ \phi_z, \varepsilon_i), & 1 \leq z \leq k^-, \\ b\frac{\alpha/m}{M-1+\alpha}F(\mathcal{D}_i;\ \phi_z, \varepsilon_i), & k^- < z \leq h, \end{cases} \tag{6}$$

- For all $z \in \{z_1, ..., z_M\}$ draw a new value from the distribution $\phi_z|\{\mathcal{D}_i$ such that $z_i = z\}$, or perform some update that leaves this distribution invariant[6].

Throughout this paper, we set $m = 5$.

Sample ε. When θ and hyperparameters are fixed, we could not analytically compute the posterior of ε as the Laplace prior is not conjugate to the likelihood function F, but it is easy to get the prior times the likelihood, so we appeal to slice sampling [18] which can sample from a distribution when we only have a function proportional to its density. Concretely, we use the single-variable slice sampling with the "stepping out" procedure to find an interval around the current point, and then the "shrinkage" procedure to sample from this interval. Due to space limit we refer the reader to Section 4 of [18] for the details of these procedures.

Sample hyperparameters. For hyperparameters, we only consider the sampling of concentration parameter α in DP. To adequately share knowledge among tasks, we encourage smaller values of α, and hence a relatively small number of task groups. As shown in [17], the likelihood of α only depends on the number of distinct θ's, so it is easy to get the product of its prior and likelihood. Thus we also use slice sampling.

3.4 Testing on Unseen Data

Samples simulated from the posterior distribution are used to estimate posterior predictive probabilities. For a unseen data in task i with covariates x', the posterior predictive probability of the response variable, y', is estimated as follows:

$$P(y' = c|x') = \frac{1}{S}\sum_{s=1}^{S} P(y' = c|x', \Theta^{(s)}), \tag{7}$$
$$P(y' = c|x', \Theta^{(s)}) = 1/(1 + exp(-c \cdot \theta_i^{(s)} \circ \varepsilon_i^{(s)} \cdot x')),$$

where $c = -1, +1$ is the class label, S is the number of post-convergence samples from MCMC, and $\Theta^{(s)}$ represents the set of parameters obtained at iteration s.

[6] Here we use slice sampling to update ϕ_z, which is similar as described in the Sampling of ε.

4 EXPERIMENTS

In this section, our algorithm is evaluated on the widely used Landmine detection data set[7] and the UCI Yeast data set[8]. We compare our algorithm with the following algorithms:

- Single Task Learning (STL, each task learns its own classifier separately), and Pooling (pool all data together and learn a single classifier);
- Xue et al.'s method [25]: directly clusters tasks with Dirichlet process prior and logistic regression likelihood;
- Group multi-task feature learning (GMTFL) [15]: learns the task grouping structure (with pre-specified number of groups) and encourages the tasks within each group to share features by an integer programming;
- Multi-Task Sparsity learning [14]: combines feature selection with both polynomial and RBF kernel selection through the maximum entropy discrimination (MED) [13] framework, which subsumes SVM as the special case.

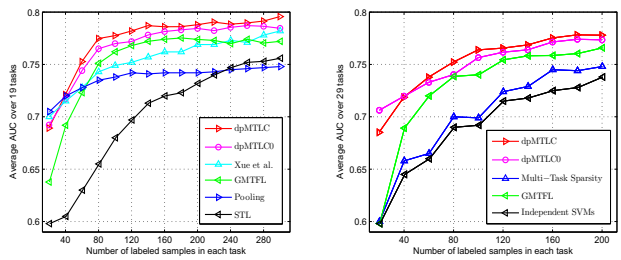
To replicate the experiments in previous work [25, 14], the averaged AUC over all involved tasks is adopted as our performance evaluation metric throughout this paper, where AUC denotes area under the Receiver Operation Characteristic (ROC) curve. A larger AUC value indicates a better classification performance. By using Hinton diagram to visualize the between-task similarity matrix we also demonstrate the learned task clustering structure of our algorithm.

For all experiments of the proposed dpMTLC and dpMTLC0, we run 1000 MCMC iterations to sample from the posterior distributions of latent variables, then discard the initial 500 samples as burn-in and use the rest for prediction. The base distribution G_0 is chosen as a zero-mean Gaussian distribution with identity covariance matrix $N(0, I)$, while the concentration parameter α of DP is assumed to have prior $\log(\alpha) \sim N(-2, 1)$. Finally the Laplace scale parameter λ is set to 0.01. The parameter settings of all compared algorithms follow instructions in their original papers and are carefully tuned on our data sets. For GMTFL the parameter γ is set to 0.05 for all experiments conducted in this paper.

4.1 Landmine Detection Data

The Landmine detection data [25, 14] are collected from 29 different landmine fields with each instance in it represented by a 9-dimensional feature vector and a corresponding binary label. Thus this data is naturally modeled as a multi-task binary classification problem where we have 29 binary classification tasks. The numbers of mines and total data points in each task are listed in Table 1. Among these 29 sub-sets, 1-15 correspond to regions that are relatively highly foliated and 16-29 correspond to regions that are bare earth or desert. Thus we expect that there are approximately two clusters of tasks corresponding to two classes of ground surface conditions.

To replicate and directly compare with the results in [25], we first evaluate our models using tasks 1-10 and 16-24. The number of training samples in each task is varied from 20 to 300 as in [25]. For each task, the training samples are randomly chosen from the corresponding data set and the remaining samples are used for testing. The results of Xue et al.'s method, Pooling and STL are directly cited from [25]. The task group number in GMTFL is set to the expected number 2. We independently repeat 50 trials of dpMTLC, dpMTLC0 and GMTFL, and the average results are shown in Figure 2 (a), from which we can see clear advantage of dpMTLC over Xue et al.'s method, GMTFL, Pooling and STL which is supposed to be owing to the large-margin classification principle and the more flexile model sharing of dpMTLC. Specially, we can find dpMTLC is also superior to dpMTLC0 because dpMTLC0 doesn't have the task rescaling flexibility in each task group compared with dpMTLC. Besides, we observed that each MCMC iteration of dpMTLC takes about 0.5 second using a MATLAB implementation on a Laptop with 2.6 GHz CPU (approximately 10 minutes for each trial on the selected 19 tasks).



(a) (b)

Figure 2. Performance comparison on the Landmine data: (a) averaged results over 50 independent trials on the selected 19 tasks with the number of training samples per task varied from 20 to 300 in each trial; (b) averaged results over 30 independent trials on all 29 tasks with the number of training samples per task varied from 20 to 200 in each trial.

Then we conduct experiments on all 29 tasks and compare with the Multi-Task Sparsity learning model in [14]. All algorithmic settings here are the same as above except that the number of training samples per task is varied from 20 to 200 as in [14]. The results of Multi-Task Sparsity learning and Independent SVMs (STL by SVM) are directly cited from [14]. We independently repeat 30 trials of dpMTLC, dpMTLC0 and GMTFL, and the mean results are shown in Figure 2 (b), from which we see dpMTLC is consistently superior to all competitors again. Note that, both the Independent SVMs and the Multi-Task Sparsity learning model have performed 5 folds cross-validation to choose their parameters, i.e., when the number of training samples is varied (from 20 to 200) for each task, the remaining samples (with labels kept unobserved) are split in half for cross-validation and testing.

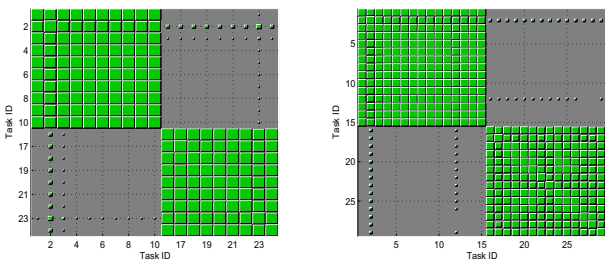
To demonstrate the task clustering effect of dpMTLC, we first compute the between-task similarity matrix and then use Hinton diagram [12] to visualize it. The i-th row and j-th column element of the between-task similarity matrix simply records the number of occurrences that task i and task j are grouped into the same cluster by dpMTLC during the post-burn-in iteration for all random and independent trials. Figure 3 shows the Hinton diagrams for the between-task similarity matrices corresponding to different experiment settings: (a)

[7] http://www.ee.duke.edu/~lcarin/LandmineData.zip/
[8] http://archive.ics.uci.edu/ml/datasets/Yeast/

Table 1. Number of Mines and Total Data Points in Each Task for the Landmine Data

Task ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
# Mines	40	48	39	19	22	29	29	26	31	15	38	35	27	31	17	32	27	31	30	37	33	39	17	41	31	36	36	26	42
# Data	690	690	689	508	509	509	510	511	508	509	689	688	508	510	507	445	449	448	449	449	451	454	447	449	445	448	448	454	449

50 independent trials of dpMTLC on selected 19 tasks with 300 random training samples per task in each trial; (b) 30 independent trials of dpMTLC on all 29 tasks with 200 random training samples per task in each trial. Note that in a Hinton diagram, the size of a block is proportional to the value of its corresponding matrix element. Thus from the figure we can easily discover the expected task-clustering structure: there are approximately two clusters, with tasks 1-15 corresponding to highly foliated regions and tasks 16-29 corresponding to bare earth or desert regions. These results are significantly better than that of Xue et al.'s method [25]. We believe this is owing to the more flexile model sharing scheme of dpMTLC.



(a) (b)

Figure 3. Hinton diagrams for the between-task similarity matrices on the Landmine detection problem: (a) 50 independent trials of dpMTLC on selected 19 tasks with 300 random training samples per task in each trial; (b) 30 independent trials of dpMTLC on all 29 tasks with 200 random training samples per task in each trial. In a Hinton diagram, the size of a block is proportional to the value of its corresponding matrix element.

4.2 Yeast Data

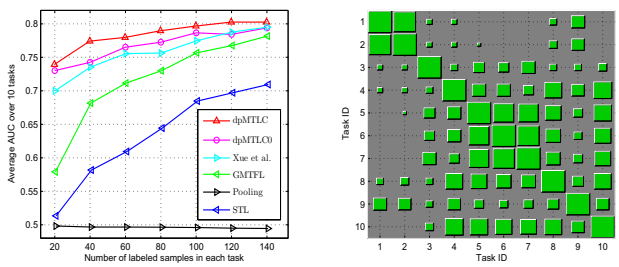
The Yeast data is a public available protein localization sites classification data set from the UCI Machine Learning Repository. There are 8 protein features for each instance in it and the total 1484 instances are distributed in 10 classes as shown in Table 2. Here following previous construction method [14] we treat the classification problem of distinguishing one class from other 9 classes as a task, thus totally we have 10 binary classification tasks. The differences among these tasks are obvious while their relatedness is more subtle. We compare dpMTLC with Xue et al.'s method, GMTFL, dpMTLC0, Pooling and STL. Here the task group number for GMTFL is set to 3 as the DP based methods typically find 2 to 4 clusters from the tasks. For Pooling and STL we adopt Minka's LG package[9]. We vary the number of training samples in each task from 20 to 140 and the averaged results over 30 independent trials are shown in Figure 4 (a), from which we see obvious advantage of MTL algorithms over Pooling and STL, and the superiority of dpMTLC over all compared algorithms again.

[9] `http://research.microsoft.com/~minka/papers/logreg/`

Note that here dpMTLC outperforms dpMTLC0 more than it does on the Landmine data because there are more differences among tasks in each task cluster here and dpMTLC0 cannot reflect these differences in learned model. Thus this verifies the effectiveness of the more flexile model sharing scheme and the large-margin classification principle of dpMTLC again. In Figure 4 (b) we also visualize the task clustering structure learned by dpMTLC from all 10 tasks with 30 independent trials and 100 random training samples per task in each trial. Here the task-clustering structure is not that clear, but one still can find tasks 1-2 and tasks 5-7 are roughly clustered together respectively. We think this is mainly due to the fact that the constructed MTL problem doesn't have very obvious task clusters or the differences between clusters are not very large.

Table 2. Number of Instances in Each Class for the Yeast Data

Class name	CYT	NUC	MIT	ME3	ME2	ME1	EXC	VAC	POX	ERL
# Instances	463	429	244	163	51	44	37	30	20	5



(a) (b)

Figure 4. Results on the Yeast data: (a) averaged performance over 30 independent trials with the number of training samples per task varied from 20 to 140 in a trial; (b) Hinton diagram learned by dpMTLC from all 10 tasks with 30 independent trials and 100 random training samples per task in each trial.

5 RELATED WORK

During the past decade, multi-task learning [6] has been the focus in the community of machine learning and data mining, and many algorithms on it have been proposed. For task grouping, in [25], Xue et al. proposed a multi-task classification method using the Dirichlet process prior where they directly cluster tasks into separate groups with tasks in the same group sharing a same logistic regression model. In [15], a group multi-task feature learning method is proposed, which assumes the tasks exist in groups and the tasks within each group share features. Just recently, Kumar and Daumé [16] proposed a method to simultaneously learn task grouping and

overlap in multi-task learning based on the assumption that task parameters within a group lie in a low dimensional subspace but allows the tasks in different groups to overlap with each other in one or more bases.

For the multi-task large-margin learning, Jebara [14] provided a method to combines feature selection with kernel selection through the maximum entropy discrimination (MED) [13] framework, which subsumes SVM as the special case. However, this method cannot find the latent task groups and thus have more risk of harmful information sharing.

The work most related to ours is [25], which focuses on the problem of learning logistic regression models for multiple classification tasks. Our model has two main advantages compared with it. First, we replaced logistic regression by large-margin machines through employing an important variant of the standard SVM, i.e., proximal SVM (PSVM) [11]. Second, rather than directly clustering tasks we assumed the model parameter of each task consists of two parts, i.e., the group-level parameter and the task rescaling parameter, which allows the model parameters of tasks in the same group to differ on some features. This way, our model is more flexible in model sharing among tasks while having even better task clustering effect.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a nonparametric Bayesian multi-task large-margin classification model which can cluster tasks into the most appropriate number of groups and induce flexible model sharing within each task group simultaneously. We assume the model parameter of each task consists of two parts, i.e., the group-level parameter and the task rescaling parameter. A Dirichlet process prior is imposed on the group-level parameter while the task rescaling parameter is assigned a one-mean Laplace prior. Experiments on the Landmine detection data set and the UCI Yeast data set demonstrate our method can not only outperform state-of-the-art MTL algorithms but also discover the task-clustering structure very well owing to the large-margin classification principle and the more flexile model sharing scheme of dpMTLC.

Though we only considered linear classification, it is easy to extend our model to nonlinear case with kernel method similar as in PSVM, which remains our future work.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (No. 61175052, 61203297, 61035003), National High-tech R&D Program of China (863 Program) (No. 2014AA012205, 2013AA01A606, 2012AA011003). Changying Du is also supported by the China Scholarship Council for one year study at Purdue University, West Lafayette, USA.

REFERENCES

[1] R.K. Ando and T. Zhang, 'A framework for learning predictive structures from multiple tasks and unlabeled data', *The Journal of Machine Learning Research*, **6**, 1817–1853, (2005).

[2] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil, 'Convex multi-task feature learning', *Machine Learning*, **73**(3), 243–272, (2008).

[3] B. Bakker and T. Heskes, 'Task clustering and gating for bayesian multitask learning', *The Journal of Machine Learning Research*, **4**, 83–99, (2003).

[4] D. Blackwell and J.B. MacQueen, 'Ferguson distributions via pólya urn schemes', *The annals of statistics*, **1**(2), 353–355, (1973).

[5] D.M. Blei and M.I. Jordan, 'Variational inference for dirichlet process mixtures', *Bayesian Analysis*, **1**(1), 121–144, (2006).

[6] R. Caruana, 'Multitask learning', *Machine Learning*, **28**(1), 41–75, (1997).

[7] Jianhui Chen, Lei Tang, Jun Liu, and Jieping Ye, 'A convex formulation for learning shared structures from multiple tasks', in *Proceedings of the 26th International Conference on Machine Learning*, pp. 137–144. ACM, (2009).

[8] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa, 'Natural language processing (almost) from scratch', *The Journal of Machine Learning Research*, **12**, 2493–2537, (2011).

[9] Changying Du, Fuzhen Zhuang, Qing He, and Zhongzhi Shi, 'Multi-task semi-supervised semantic feature learning for classification', in *12th IEEE International Conference on Data Mining (ICDM)*, pp. 191–200. IEEE, (2012).

[10] T.S. Ferguson, 'A bayesian analysis of some nonparametric problems', *The annals of statistics*, **1**(2), 209–230, (1973).

[11] Glenn Fung and Olvi L Mangasarian, 'Proximal support vector machine classifiers', in *Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 77–86. ACM, (2001).

[12] Geoffrey E Hinton and Terrance J Sejnowski, 'Learning and relearning in boltzmann machines', *MIT Press, Cambridge, Mass*, **1**, 282–317, (1986).

[13] T. Jaakkola, M. Meila, and T. Jebara, 'Maximum entropy discrimination', in *Proceedings of advances in Neural Information Processing Systems*, (1999).

[14] Tony Jebara, 'Multitask sparsity via maximum entropy discrimination', *The Journal of Machine Learning Research*, **12**, 75–110, (2011).

[15] Z. Kang, K. Grauman, and F. Sha, 'Learning with whom to share in multi-task feature learning', in *Proceedings of the 28rd International Conference on Machine Learning*, (2011).

[16] Abhishek Kumar and Hal Daumé III, 'Learning task grouping and overlap in multi-task learning', in *Proceedings of The 29th International Conference on Machine Learning*, (2012).

[17] R.M. Neal, 'Markov chain sampling methods for dirichlet process mixture models', *Journal of computational and graphical statistics*, 249–265, (2000).

[18] R.M. Neal, 'Slice sampling', *The annals of Statistics*, 705–741, (2003).

[19] Alexandre Passos, Piyush Rai, Jacques Wainer, and Hal Daumé III, 'Flexible modeling of latent task structures in multitask learning', in *Proceedings of The 29th International Conference on Machine Learning*, (2012).

[20] Y. Qi, O. Tastan, J.G. Carbonell, J. Klein-Seetharaman, and J. Weston, 'Semi-supervised multi-task learning for predicting interactions between hiv-1 and human proteins', *Bioinformatics*, **26**(18), i645–i652, (2010).

[21] Bernardino Romera-Paredes, Hane Aung, Nadia Bianchi-Berthouze, and Massimiliano Pontil, 'Multilinear multitask learning', in *Proceedings of The 30th International Conference on Machine Learning*, pp. 1444–1452, (2013).

[22] J. Sethuraman, 'A constructive definition of dirichlet priors', Technical report, DTIC Document, (1991).

[23] Yee Whye Teh, 'Dirichlet process', in *Encyclopedia of machine learning*, 280–287, Springer, (2010).

[24] A. Torralba, K.P. Murphy, and W.T. Freeman, 'Sharing visual features for multiclass and multiview object detection', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**(5), 854–869, (2007).

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

[26] Y. Zhang, D.Y. Yeung, and Q. Xu, 'Probabilistic multi-task feature selection', in *Proceedings of advances in Neural Information Processing Systems*, pp. 2559–2567, (2010).