NONPARAMETRIC BAYESIAN FEATURE SELECTION FOR MULTI-TASK LEARNING

Hui Li 1, Xuejun Liao 2, Lawrence Carin 1,2

1 Signal Innovations Group, Inc., Durham, NC 27703, USA
2 Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA

ABSTRACT

We present a nonparametric Bayesian model for multi-task learning, with a focus on feature selection in binary classification. The model jointly identifies groups of similar tasks and selects the subset of features relevant to the tasks within each group. The model employs a Dirichlet process with a beta-Bernoulli hierarchical base measure. The posterior inference is accomplished efficiently using a Gibbs sampler. Experimental results are presented on simulated as well as real data.

1. INTRODUCTION

Multi-task learning (MTL) is an approach to enhancing generalization by exploiting commonalities among related tasks [1]. Tasks could be related in various ways, necessitating different MTL methods. Recent MTL approaches have assumed that each task is sufficiently different from others to warrant distinct parameters [2]. These approaches, which focus on learning a common sparse feature representation shared across the tasks, are usually referred to as multi-task feature selection/learning [2, 3]. The key difference from many previous MTL studies (see [4] for an example) is that the parameter associated with each feature is task-specific. Removal of spurious features or the correlations between features is known to promote generalization. In an MTL setting, the benefits could be more prominent because information-sharing helps to identify the common sparse features.

In this paper we consider multi-task feature selection under the assumption that the tasks are partitioned into groups, and that the tasks in the same group share a common subset of relevant features. We propose a nonparametric Bayesian model for jointly inferring groups of similar tasks and selecting features relevant to the tasks in each group. The model employs a Dirichlet process (DP) [5] with the base measure specified by a beta-Bernoulli process (BBP) [6]. The DP partitions the tasks into groups, and the BBP selects the features relevant to the tasks in each group, with the parameter associated with each feature task-specific. The method is nonparametric in the sense that both the number of groups and the subset of features selected in each group are not specified a priori but are inferred from data. The proposed method extends the methods based on $L_{2,1}$ regularization [2, 3, 7], which assume that all tasks share the same subset of features.

It also extends the method in [4], in which the tasks in each group share the specific parameters. Performance comparisons based on both simulated and real data demonstrate the merits of the proposed method.

2. MODEL FORMULATION

We consider $M$ tasks. For $m = 1, 2, \cdots, M$, the goal in the $m$-th task is to learn a binary classifier based on the set of training examples $D_m = \{(x_1^m, y_1^m) : i = 1, 2, \cdots, n_m\}$, where $x_i^m \in \mathbb{R}^K$ is the feature vector and $y_i^m \in \{0, 1\}$ the associated class label in the $i$-th training example. The proposed multi-task feature selection model has a hierarchical structure, which is specified as follows.

$$
\begin{align*}
\pi_j &\sim \text{Beta}(1, \alpha), \\
\theta_{jm} &\sim \prod_{d=1}^{K} \text{Bernoulli}(\tau_d), \\
\tau &\sim \prod_{d=1}^{K} \text{Beta}(\alpha/2, \beta(K - 1)/2), \\
\alpha &\sim \text{Gamma}(\rho_1, \rho_2), \\
w_m &\sim \text{Normal}(0, \beta I_K),
\end{align*}
$$

for $i = 1, 2, \ldots, n_m$ and $m = 1, 2, \ldots, M$, where the symbol $\circ$ denotes element-wise multiplication of two vectors, $I_K$ is a $K \times K$ identity matrix, and $I(\cdot)$ is one if the argument is true and zero otherwise. The equations in (1) are explained as follows. The first two equations define a set of binary linear classifiers based on probit regression [8], with $\theta_{jm} \circ w_m$ giving the parameters of the classifier for task $m$, where $\theta_{jm} \in \{0, 1\}^K$ specifies the features selected in task $m$ and $w_m \in \mathbb{R}^K$ the parameters associated with the selected features. We assume that the first feature is the constant one, thus the first parameter is an intercept. Equations 3-7 in (1) specify that $\theta_{jm}$ is governed by a Dirichlet process with concentration $\alpha$ and the base specified by a beta-Bernoulli process (see equations 6-7). Equation 8 gives the hyper-prior for the DP concentration and equation 9 the prior distribution for $w_m$. The model in (1) is termed Bernoulli DP (BDP), to indicate it uses a Dirichlet process with a Bernoulli base distribution.
The Dirichlet process partitions \( \{ \theta_m \}_{m=1}^M \) into \( J \leq M \) groups \( \{ \theta^*_j \}_{j=1}^J \), with \( \theta_m = \theta^*_j \) if task \( m \) is in group \( j \). The ones in \( \theta^*_j \) index the features selected for the tasks in group \( j \), but each task has its own parameters associated with the selected features, which are defined through \( \{ w_m \}_{m=1}^M \). The methods based on \( L_{2,1} \) regularization [2, 3, 7] correspond to the special case of \( J = 1 \).

3. POSTERIOR INFERENCE

We first introduce latent variables \( \eta = \{ \eta_m \}_{m=1}^M \), with \( \eta_m \) indicating \( \theta_m = \theta^*_m \). For notational convenience, let a variable with its superscript and/or subscript omitted denote the set of variables obtained by taking the superscript and/or subscript of this variable over all possible values. For example, \( y = \{ y_i^m : i = 1, 2, \ldots, n_m \}_{m=1}^M \). We are interested in the posterior distribution of \( (\theta^*, \eta, w) \) given data \((y, x)\) and root hyper-parameters \((p_1, p_2, a, b, \beta)\). Given \((\theta^*, \eta, w)\), the model parameters are determined as \( \theta^*_m \circ w_m \), for \( m = 1, 2, \ldots, M \). To sample from this posterior, one needs to integrate out latent variables \((z, \tau, V, \alpha)\), which is intractable. Thus, we instead sample from the joint posterior,

\[
p(\theta^*, \eta, w, z, \tau, V, \alpha|y, x, p_1, p_2, a, b, \beta) \\
\propto \prod_{m=1}^M p(\eta_m|V)p(w_m|\beta)\prod_{i=1}^{n_m} p(y_i^m|z_i^m)p(z_i^m|x_i^m, \eta_m, \theta^*, w_m) \\
\times p(\alpha|p_1, p_2)p(\tau|a, b)\prod_{j=1}^N p(V_j|\alpha)\prod_{d=1}^L p(\theta^*_d|\tau_d) \\
\tag{2}
\]

where the infinite mixture \((G)\) in equation 3 of (1) has been truncated to have a finite number of, i.e., \( N \), components. The joint posterior can be analyzed by Gibbs sampling, based on the full conditional distributions derived from (1) and (2). The full conditionals all turn out to be standard distributions, therefore each iteration of the Gibbs sampler is easily implemented. After the Gibbs sampler converges, the samples fully characterize the statistical properties of the joint distribution in (2) as well as the marginal distribution \( p(\theta^*, \eta, w|y, x, p_1, p_2, a, b, \beta) \). The details of Gibbs sampling are omitted here for brevity.

4. EXPERIMENTAL RESULTS

We consider a synthetic example and a real remote-sensing application. The performance of the proposed method is evaluated in comparison with five other methods. The data sets considered here all involve binary classification problems, with the AUC employed to measure algorithmic performances on these data, where AUC stands for area under the receiver operation curve (ROC) [9].

For each data set, the results on the performance comparison are presented as the AUC averaged over the tasks and the Monte Carlo runs, as function of the number of training examples per task. Given the number of training examples, we perform multiple Monte Carlo trials, in each of which the data for each task are randomly split into a training subset and a test subset. The training subset (labeled examples) is used to train each of the six methods, and the test subset is used to test the generalization performance of each method. The AUC is calculated for each method in each task by ranking the test examples based on the class probabilities predicted by the method. The number of Monte Carlo trials is 20 for examples 1-2 and 50 for the third example.

The methods being compared in the experiments are summarized as follows.

1. MTL-BDP: multi-task learning (MTL) with BDP, based on the model in (1).
2. MTL-BBP: MTL with beta-Bernoulli process, which is a special case of (1) when \( J = M \), i.e., \( \theta_m = \theta^*_m \), \( m = 1, 2, \ldots, M \).
3. MTL-DP: the method referred to SMTL-2 in [4], which is MTL with the Dirichlet process (DP) prior, where a sparseness-promoting DP-base is constituted by \( K \) independent zero-mean normal distributions each with an independent gamma hyper-prior on its precision.
4. MTL-\( L_{2,1} \): multi-task feature selection with \( L_{2,1} \)-NORM [2], implemented in a Bayesian manner. This method is essentially an extension of the method in [10] and is of independent interest.
5. STL: single-task learning, where a probit model is learned independently in each task, using the method in [10].
6. Pooling: a single probit model is learned for all tasks, using the method in [10].

All six methods are based on Bayesian hierarchical models, using sparseness-promoting priors for selecting relevant features to be used in the probit models. The methods differ in the way they handle the relations between the feature-selections across different tasks. The method of MTL-\( L_{2,1} \) imposes that all tasks share the same subset of features, while MTL-DP relaxes this by letting only the tasks within a cluster share the same features (feature-selection is different across the clusters). However, MTL-DP also imposes that the tasks in a cluster have the same probit parameters associated with the selected features. This stringent constraint is relaxed in MTL-BDP (the proposed method) by allowing each task to have its own parameters and sharing only the feature-selections across different tasks. The method of MTL-BDP imposes a weaker prior than MTL-\( L_{2,1} \) by allowing the features in different tasks to be overlapped but not exactly the same. Since there are no clusters formed in MTL-BBP, any two tasks will have little chance of sharing exactly the same features. The methods of STL and pooling lie at two extremities in the spectrum spanning the six methods, the former imposing the weakest prior while the latter imposes the strongest.

Throughout this section, the root-layer hyper-parameters involved in (1) are specified as follows: \( \beta = a = b = 1 \), \( \rho_1 = \rho_2 = 0.05 \), the infinite mixture constituting \( G \) in (1).
is truncated to have $N = M$ components, where $M$ is the number of tasks, and $K$ is the original feature dimensionality.

The method of MTL-BBP, which is implemented as a special case of MTL-BDP as discussed above, uses a subset (involving only $\beta, a, b$) of the above setting. The setting of MTL-DP is the same as in [4]. The methods of MTL-$L_{2,1}$, STL, and pooling have a single root-layer parameter $\gamma$, which is set to one for all of these methods.

4.1. Results on synthetic data

We first use a synthetic example to demonstrate the performance of the proposed model and illustrate its working mechanism. We assume six tasks, respectively associated with data sets $\{x_i^m, y_i^m\}_{i=1}^{n_m}$, $m = 1, 2, \ldots, 6$, where $n_m \equiv 500$. The feature vectors $\{x_i^m, i = 1, \ldots, n_m\}_{m=1}^6$, each with a dimensionality $K = 50$, are independently drawn from a zero-mean Gaussian distribution with identity covariance matrix. The class labels are determined as $y_i^m = I(x_i^m \theta_m \mathbf{1}_{x_i^m \geq 0})$ for any $i$ and $m$, with the following ground-truths for the classifier coefficients $\{\xi_i^m\}_{m=1}^6$. The classifier coefficients in tasks $m = 1, 2, 3$ have a common nonzero support $S_1 = \{15, 17, 23, 27, 35, 36, 39, 42, 48, 50\}$ but task-specific nonzero values. The coefficients in tasks $m = 4, 5, 6$ have a common nonzero support $S_2 = \{2, 4, 11, 18, 23, 28, 35, 43, 46, 47\}$, which partially overlap with the support in tasks 1, 2, 3, and the nonzero values are similarly task-specific. The nonzero support in each task defines the features actually used in the associated classifier. Therefore tasks 1-3 share the subset of features defined by $S_1$, and tasks 4-6 shares another subset of features defined by $S_2$. This creates two clusters.

In our model, the nonzero support in task $m$ is specified by the binary vector $\beta_m$, and the nonzero classifier coefficients are specified by the corresponding elements in $w_m$ (the other elements of $w_m$ are not used and their values are irrelevant).

The ground-truth classifier coefficients are plotted in Figure 1, along with their estimates obtained by MTL-BDP, as one sample after the Gibb sampler has converged, based on using 140 randomly determined labeled examples. From Figure 1, we observe that MTL-BDP not only correctly selects the relevant features for each task, but also obtains reasonably good estimates of the parameters associated with the relevant features. The performance comparison on the synthetic data are reported in Figure 2(a), in the format as described at the beginning of Section 4. It is clear that the proposed method (MTL-BDP) performs the best. The methods of MTL-$L_{2,1}$ and MTL-DP performs competitively, because the six tasks form two clear clusters. Recall that MTL-$L_{2,1}$ lacks the mechanism of generating more than one cluster but it allows tasks-specific probit-parameters associated with the selected features, whereas MTL-DP has the mechanism of generating two clusters but it forces identical parameters upon the tasks in each cluster. The fact that MTL-DP is outperformed by MTL-$L_{2,1}$ indicates that feature selection can have a heavier influence on the performance than the detailed parameter values. The method of MTL-BBP performs much worse, because it lacks a clustering mechanism and prevents the tasks form selecting exactly the same features; STL performs poorly and pooling achieves AUCs below 0.65 (not shown here to improve the figure’s readability).

Figure 2(b) plots the task-clusters (groups of similar tasks) inferred by MTL-BDP. The plot is presented in the form of a Hinton diagram [11], where the $(i, j)$-th element indicates the frequency that task $i$ and $j$ are assigned to the same cluster, calculated from the collection iterations of Gibbs samplers across multiple Monte Carlo trials. The Hinton diagram in Figure 2(b) indicates that tasks 1-3 are in a cluster and tasks 4-5 are in another cluster, which agrees with the ground-truth that the tasks in each cluster share the same relevant features (see Figure 1). The performance of MTL-BDP is attributed to two facts: it successfully identifies the two groups of similar tasks, and it does not enforce identical probit-parameters upon the tasks within a group; these two facts ensure that information sharing is appropriate.

4.2. Shallow water target detection

The second example is a real remote-sensing problem, where we consider shallow-water target detection, with the goal of detecting oil rigs in the presence of oil tankers (the data
are available upon request). For each unknown item, a 16-dimensional feature vector is extracted from acoustic data collected with active sonar operating in different frequency bands. There are a total of 9 data sets, constituting 9 tasks, of which the first four are collected in a high-frequency band, and the last five are collected in a low-frequency band.

The results are presented in Figure 3(a). Of the methods compared, the proposed MTL-BDP yields best performance, with the gains more prominent when the labeled data are scarce. The task-clusters inferred by MTL-BDP are plotted in Figure 3(b), which indicates a weak cluster formed by tasks 1-4; this is in good agreement with the ground-truth that tasks 1-4 are collected with sonar operating in one frequency-band, while tasks 5-9 are collected in another frequency-band. However, the clusters are weak, showing that the tasks are not so closely related. This explains the performances of MTL-BBP and MTL-DP. The performance of MTL-BBP is more competitive here than in the synthetic example, because the absence of a clustering-mechanism in MTL-BBP makes it more appropriate for the present example. The performance of MTL-DP is better in the synthetic example because the tasks form two clear clusters there.

5. CONCLUSIONS

We have developed a new method for multi-task feature selection, in the context of binary classification. By utilizing the clustering property of the Dirichlet process (DP) and feature selection property of the beta-Bernoulli process, our method discovers groups of similar tasks, while simultaneously inferring the subset of features in each group. Three prominent features of our method are: (a) it promotes clustering of the tasks; (b) it allows tasks to share relevant features within each cluster; (c) it does not enforce identical classifier parameters upon the tasks within a group. We used a synthetic example to demonstrate the working mechanism of the method. Experiments on real data show that such a sharing structure is appropriate for data in real applications.

6. REFERENCES