MULTI-TASK CLASSIFICATION WITH INFINITE LOCAL EXPERTS

Chunping Wang, Qi An, Lawrence Carin

Department of Electrical and Computer Engineering
Duke University
Durham, NC 27708

David B. Dunson

Department of Statistical Science
Duke University
Durham, NC 27708

ABSTRACT

We propose a multi-task learning (MTL) framework for nonlinear classification, based on an infinite set of local experts in feature space. The usage of local experts enables sharing at the expert-level, encouraging the borrowing of information even if tasks are similar only in subregions of feature space. A kernel stick-breaking process (KSBP) prior is imposed on the underlying distribution of class labels, so that the number of experts is inferred in the posterior and thus model selection issues are avoided. The MTL is implemented by imposing a Dirichlet process (DP) prior on a layer above the task-dependent KSBPs.

Index Terms— Multi-task learning, Classification, Expert, Dirichlet process, Kernel stick-breaking process

1. INTRODUCTION

Multi-task learning (MTL) [1], which allows the learning of multiple tasks simultaneously to improve generalization performance, has become an important topic in the machine learning community. In recent research, a hierarchical structure has been favored, where information is transferred via a common prior, within a hierarchical Bayesian model [2]. In [3], where a Dirichlet process (DP) [4] prior was introduced as the common prior in hierarchical Bayesian models, information is transferred only between related tasks.

To the best of our knowledge, most existing MTL classification algorithms allow for sharing only at the task-level (two tasks share all of their data or none of their data, but do not share partial data). In the work presented here we develop a novel classification model with local (in feature space) experts, allowing for sharing subsets of local experts (and associated data) across tasks. Although related to previous work with local experts [5], the proposed model may have a theoretically infinite number of experts and thus model selection issues are avoided. With experts as the basic components, when sharing occurs between tasks it is not assumed that all parameters must be shared (as required in [3]).

The task-dependent classifiers are based on a novel application of the newly developed kernel stick-breaking process (KSBP) [6], which is an augmentation of the stick-breaking representation of the DP [4]. To learn multiple tasks simultaneously, we impose a DP prior on the upper layer of the hierarchical Bayesian model. Essentially, a "firm" of experts with corresponding locations are provided, and each task "selects" appropriate local experts automatically from the firm.

2. DIRICHLET PROCESS

2.1. Stick-Breaking Cunstruction

The stick-breaking construction [7] provides an explicit form of a draw from a DP [4] prior. Specifically, assume a DP prior with base measure G_0 and precision parameter $\alpha \geq 0$ is assigned on a measure G. It has been proven [7] that the draw G may be constructed as

$$G = \sum_{h=1}^{\infty} \pi_h \delta_{\boldsymbol{\theta}_h^*}, \tag{1}$$

with $0 \leq \pi_h \leq 1$ and $\sum_{h=1}^{\infty} \pi_h = 1$ a.s., where δ_m is a Kronecker's delta function and

$$\pi_h = V_h \prod_{l=1}^{h-1}(1 - V_l), \quad V_h \overset{iid}{\sim} Beta(1, \alpha), \quad \boldsymbol{\theta}_h^* \overset{iid}{\sim} G_0.$$

Since the weights $\{\pi_h\}_{h=1}^{\infty}$ decrease stochastically with h, the summation may be truncated with N terms, yielding an N-level truncated approximation [8].

Assuming that the underlying variables $\{\boldsymbol{\theta}_i\}_{i=1}^n$ are drawn from G, the associated data $y_i \sim F(\boldsymbol{\theta}_i)$ will naturally cluster about distinct values $\boldsymbol{\theta}_h^*$ taken by $\boldsymbol{\theta}_i$. Therefore, the number of clusters is automatically determined. In this work y_i correspond to labels, and we wish to make the model dependent on the feature vector $\boldsymbol{x}_i$.

2.2. Kernel Stick-Breaking Process

Based on the stick-breaking construction, Dunson and Park [6] proposed a class of kernel-based priors called the kernel stick-breaking process (KSBP). The collection $G_{\mathcal{X}} = \{G_{\boldsymbol{x}} : \boldsymbol{x} \in \mathcal{X}\}$ is assigned a KSBP prior, denoted $G_{\mathcal{X}} \sim$

$\mathcal{KSBP}(\boldsymbol{a}, \boldsymbol{b}, \mathcal{Q}, \mathcal{H})$ if for all $\boldsymbol{x} \in \mathcal{X}$

$$G_{\boldsymbol{x}} = \sum_{h=1}^{\infty} \pi_h(\boldsymbol{x}; V_h, \boldsymbol{\Gamma}_h) G_h^*, \tag{2}$$

$$\pi_h(\boldsymbol{x}; V_h, \boldsymbol{\Gamma}_h) = W(\boldsymbol{x}; V_h, \boldsymbol{\Gamma}_h) \prod_{l<h} \{1 - W(\boldsymbol{x}; V_l, \boldsymbol{\Gamma}_l)\},$$

$$W(\boldsymbol{x}; V_h, \boldsymbol{\Gamma}_h) = V_h K(\boldsymbol{x}, \boldsymbol{\Gamma}_h),$$

where $\{V_h, \boldsymbol{\Gamma}_h, G_h^*\}_{h=1}^{\infty}$ is a countable sequence with probability weights $V_h \sim Beta(a_h, b_h)$, locations $\boldsymbol{\Gamma}_h \overset{iid}{\sim} \mathcal{H}$ and probability measures $G_h^* \overset{iid}{\sim} \mathcal{Q}$; $K(\boldsymbol{x}, \boldsymbol{\Gamma}_h)$ defines a kernel function. Comparing (2) with (1), we notice that a kernel function at sample-independent locations $\boldsymbol{\Gamma}_h$ is introduced as a discount on V_h, and the further away $\boldsymbol{x}$ is from $\boldsymbol{\Gamma}_h$, the larger the penalty is. As a result, The KSBP prior reflects our belief that nearby points in feature space tend to cluster together and share a common G_h^*.

3. CLASSIFIER WITH INFINITE LOCAL EXPERTS

3.1. Mathematical Model

Consider a c-class classification problem with a training size of n, i.e., $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$, where $\boldsymbol{x}_i$ are feature vectors and $y_i \in \{1, \ldots, c\}$ are labels. We assume that y_i have multinomial distributions with probabilities $\boldsymbol{\theta}_i = [\theta_{i1}, \ldots, \theta_{ic}]^T$ on labels $\{1, \ldots, c\}$ and impose a KSBP prior measure, i.e.,

$$y_i \sim \sum_{k=1}^{c} \theta_{ik} \delta_k, \boldsymbol{\theta}_i | G_{\boldsymbol{x}_i} \sim G_{\boldsymbol{x}_i}, G_{\mathcal{X}} \sim \mathcal{KSBP}(1, \alpha, Q_0, H_0),$$

$$\text{with} \quad G_{\boldsymbol{x}_i} = \sum_{h=1}^{\infty} \pi_h(\boldsymbol{x}_i; V_h, \boldsymbol{\Gamma}_h, \psi_h) \boldsymbol{g}_h, \tag{3}$$

where $\boldsymbol{\Gamma}_h \overset{iid}{\sim} H_0$, $V_h \overset{iid}{\sim} Beta(1, \alpha)$ and $\boldsymbol{g}_h \overset{iid}{\sim} Q_0$. According to (3), the prior measure of probabilities $\boldsymbol{\theta}_i$ is an infinite mixture of experts $\boldsymbol{g}_h$, which are simply probability mass functions over class labels, with weights $\boldsymbol{\pi}_h$ data-dependent. As in mixture models, we decouple the mixture components by introducing an indicator z_i for each data point such that $z_i = h$ when the ith subject is assigned to the hth expert.

For the sake of conjugacy, we choose Q_0 to be a Dirichlet measure with parameter $\gamma \boldsymbol{u}_0$, and H_0 to be a discrete measure $H_0(\cdot) = \sum_{j=1}^{L_{\boldsymbol{\Gamma}}} e_j \delta_{\tilde{\boldsymbol{\Gamma}}_j}(\cdot)$ with a total of $L_{\boldsymbol{\Gamma}}$ discrete location candidates $\tilde{\boldsymbol{\Gamma}}_j$. As discussed further when presenting results, if $\boldsymbol{x}_i$ are of low dimension, it is possible to randomly constitute $\tilde{\boldsymbol{\Gamma}}_j$ as draws from a continuous distribution over the support of interest; for high-dimensional data we simply use all the available data to define the set $\{\tilde{\boldsymbol{\Gamma}}_j\}_{j=1}^{L_{\boldsymbol{\Gamma}}}$. This latter approach is related to the SVM [9] and RVM [10], in the sense of selecting basis locations from available data samples. Weights $\pi_h(\boldsymbol{x}_i; V_h, \boldsymbol{\Gamma}_h, \psi_h)$ are defined as in Section 2.2 except that the scale parameter of the kernel function is expressed explicitly, with $K(\boldsymbol{x}_i, \boldsymbol{\Gamma}_h, \psi_h) = \exp(-\psi_h \parallel \boldsymbol{x}_i - \boldsymbol{\Gamma}_h \parallel^2 /2)$ defined as a Gaussian kernel function. By assigning a discrete prior on ψ_h, we infer ψ_h in principle and allow that the variance of each cluster is distinct.

3.2. Posterior Inference

A Gibbs sampler with a data augmentation strategy is performed to infer the posterior of the hidden variables for the model of truncation level N. Due to space limitations, we only provide details on the updating of V_h, which involves the data augmentation.

We introduce auxiliary variables $A_{ih} \sim Bern(V_h)$, and $B_{ih} \sim Bern(K(\boldsymbol{x}_i, \boldsymbol{\Gamma}_h, \psi_h))$ for $h = 1, \ldots, N-1$ and let $A_{iN} = B_{iN} = 1$. Consequently, $z_i = \min\{h : A_{ih} = B_{ih} = 1\}$ is equivalent to $p(z_i = h) = \pi_h(\boldsymbol{x}_i; V_h, \boldsymbol{\Gamma}_h, \psi_h)$. Thus, the conditional posteriors for V_h are

$$\begin{aligned}&(V_h | \boldsymbol{z}, \boldsymbol{\Gamma}_h, \psi_h, \alpha)\\ \sim &\text{Beta}(1 + \sum_{i: z_i \geq h} A_{ih}, \alpha + \sum_{i: z_i \geq h} (1 - A_{ih}))\end{aligned}$$

where for $h = 1, \ldots, z_i - 1$,

$$p(A_{ih} = B_{ih} = 0 | z_i) = \frac{(1 - V_h)(1 - K(\boldsymbol{x}_i, \boldsymbol{\Gamma}_h, \psi_h))}{1 - V_h K(\boldsymbol{x}_i, \boldsymbol{\Gamma}_h, \psi_h)},$$

$$p(A_{ih} = 0, B_{ih} = 1 | z_i) = \frac{(1 - V_h) K(\boldsymbol{x}_i, \boldsymbol{\Gamma}_h, \psi_h)}{1 - V_h K(\boldsymbol{x}_i, \boldsymbol{\Gamma}_h, \psi_h)},$$

$$p(A_{ih} = 1, B_{ih} = 0 | z_i) = \frac{V_h (1 - K(\boldsymbol{x}_i, \boldsymbol{\Gamma}_h, \psi_h))}{1 - V_h K(\boldsymbol{x}_i, \boldsymbol{\Gamma}_h, \psi_h)},$$

for $h = z_i$, $A_{ih} = B_{ih} = 1$.

4. MULTI-TASK CLASSIFICATION WITH INFINITE LOCAL EXPERTS

Assume we have M sets of data $\mathcal{D}_m = \{\boldsymbol{x}_{mi}, y_{mi}\}_{i=1}^{n_m}$, and our goal is to design a classifier for each data set and infer classifier parameters for all the tasks simultaneously. In the hierarchical Bayesian framework, we may borrow information across those M tasks in several different ways since our classification model for a single task is fairly flexible. For example, we may impose that the whole model should be shared if two tasks are similar, with no sharing otherwise, as in [3]. Instead, we prefer a partial sharing strategy so that data from two tasks may be shared even when they are only partially related. Specifically, we encourage the sharing of the local experts $\boldsymbol{g}_{mh}$ and the associated locations $\boldsymbol{\Gamma}_{mh}$ by imposing an upper layer DP on their priors Q_m and H_m with a common precision τ, and base measures Q_0 and H_0, respectively. The model could be denoted as

$$\begin{aligned}&y_{mi} \sim \sum_{k=1}^{c} \theta_{mi,k} \delta_k, \quad \boldsymbol{\theta}_{mi} | G_{\boldsymbol{x}_{mi}} \sim G_{\boldsymbol{x}_{mi}},\\ &G_{\mathcal{X}_m} \sim \mathcal{KSBP}(1, \alpha_m, Q_m, H_m),\\ &Q_m \otimes H_m \sim \mathcal{DP}(\tau, Q_0 \otimes H_0)\end{aligned}$$

The hierarchical KSBP implemented here is related to the hierarchical DP (HDP) [11]; however, clear differences exist. In [11], only the supports of the lower layer DP draws are shared; however, we share the locations appearing in a kernel function as well. For the sake of inference, instead of using one combined indicator as in [11], we decouple the upper (inter-task) and bottom (intra-task) layers by explicitly introducing indicators for both.

5. EXPERIMENTS

In the experiments below, the distribution H_0 was in general constituted by $L_{\mathbf{\Gamma}} = 1000$ random draws from broad multivariate Gaussian distributions, and exceptions will be indicated otherwise.

5.1. Single-Task Learning on Benchmark Data Sets

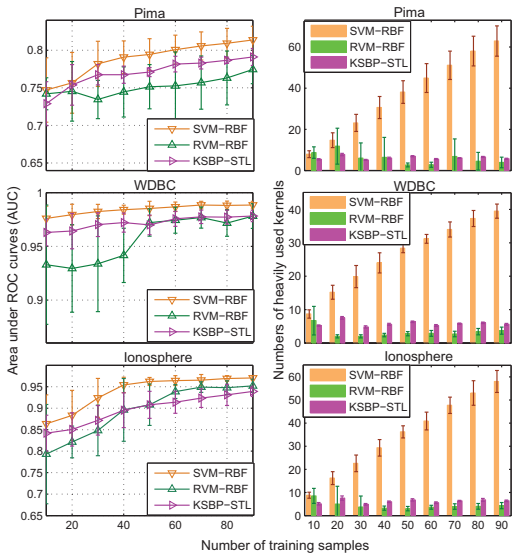

Fig. 1. STL results on benchmark data sets. The area under the ROC curves (AUC) (left) and the number of kernels used (right). Error bars reflect the standard deviation across ten random partitions of training and testing subsets.

In order to evaluate the proposed KSBP-STL classifier, we consider three benchmark data sets available from the UCI machine learning repository [12]: the Pima Indians Diabetes database (Pima), Wisconsin Diagnostic Breast Cancer (WDBC) and the Johns Hopkins University Ionosphere database (Ionosphere). For high-dimensional WDBC (30-d) and Ionosphere (34-d) sets, we use all available data points as location candidates $\tilde{\mathbf{\Gamma}}_j$ for H_0. Since the proposed model is capable of handling nonlinear problem, we compare it with both the SVM [9] and the RVM [10] with RBF kernels. As discussed, the kernel parameters for the proposed model are automatically inferred; however, those for the SVM and RVM are preset to be five and one, respectively, which appear to be the most appropriate values among several trials.

As in Figure 1, all classifiers are evaluated from two aspects: the number of kernels used (the right column), and the area under the receive operating characteristics (ROC) curves for the test subset (the left column). As indicated in Figure 1, the performance of the proposed KSBP classifier is consistently comparable to the state-of-art classifiers with nonlinear kernels and appropriate kernel parameters. We also observed that the KSBP classifier is as sparse as the RVM classifier on average with a tighter variance, and generally markedly sparser than the SVM.

5.2. Multi-Task Learning

5.2.1. Synthetic Data

To illustrate the local sharing mechanism of the KSBP-based MTL model, we simulate six classification tasks in a 2-dimensional space. As suggested by Figure 2, the data are designed such that each row of tasks are generated from the same underlying distributions and there is local similarity between tasks in different rows.

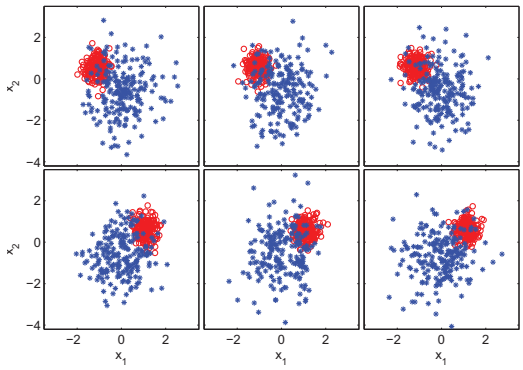

Fig. 2. Six simulated tasks with the true labels dictated.

Figure 3 shows the estimated posterior probability of $y = 1$ as heat maps for a random partition with ten training data each task. With such insufficient training data, the algorithm automatically reveals the sophisticated sharing structure and produces reasonable predictions. At the task-level, each row of tasks share a common model, which can also be achieved by a DP-based classifier as in [3]. However, only the expert at location $\mathbf{\Gamma}_3$ is shared by all the tasks, while the other experts are selectively shared, which is beyond the scope of such a DP-based classifier without local components. Consequently, on such a data set with clear local similarity, the KSBP-based MTL algorithm generally outperforms the DP-based MTL classifier proposed in [3]. (The plot is omitted due to space limitations.)

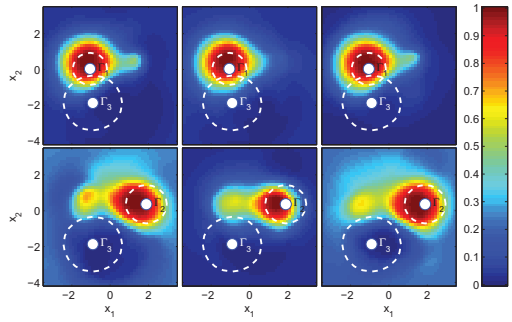

Fig. 3. Color heat map of the probability of $y = 1$ computed by integrating across the full posterior for the parameters (10 training samples each task). Depicted are locations (white big dots) and kernel widths (radius of white dashed circles) of dominant sticks from the 1000th MCMC sample.

5.2.2. *Landmine Detection*

In an application of landmine detection, data collected from 19 landmine fields [3] are treated as 19 subtasks. In all tasks, each target is characterized by a 9-dimensional feature vector.

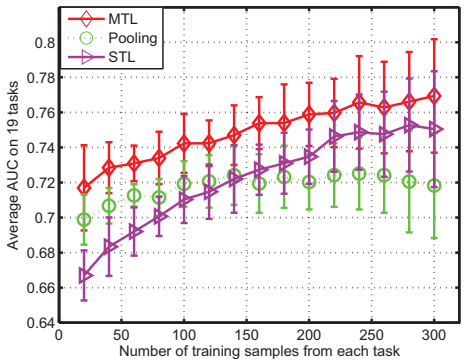

Fig. 4. Average AUC on 19 tasks of landmine detection, for the KSBP-based classifiers. Error bars reflect the standard deviation across ten random partitions of training and testing subsets.

The average AUC over all the 19 tasks as given by the STL, pooling and the MTL KSBP-based models are presented in Figure 4, where we observe the impact of sharing. For the STL, no information sharing among tasks is assumed; for the pooling, data from all the tasks are imposed to share a common underlying model; while for the MTL, sharing is encouraged only when tasks are at least partially related. As a result, the STL suffers from insufficient training data, and the assumption of a universal model impairs the performance when training data are abundant, while the MTL always performs best. This behavior coincides with what was presented in [3] where logistic regression (LR) models were employed. We also notice that (by comparing to [3]) the KSBP-based STL classifier outperforms the linear LR STL classifier overall, while the KSBP-based MTL classifier performs comparably to the LR MTL classifier [3].

6. CONCLUSIONS

We have proposed a supervised classification model capable of handling problems with nonlinear decision boundaries, without model selection issues. Experiments show that the proposed STL model is comparable to state-of-art classifiers with preset appropriate kernel parameters. With task-dependent models consisting of local experts, we have proposed a hierarchical Bayesian framework for multi-task learning, which allows for partial sharing of information across tasks. Encouraging results have been demonstrated on simulated data, and on a multi-task data set corresponding to a real sensing example.

7. REFERENCES

[1] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, pp. 41–75, 1997.

[2] J. Zhang, Z. Ghahramani, and Y. Yang, "Learning multiple related tasks using latent independent component analysis," in *Advances in Neural Information Processing Systems*, 2006.

[3] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram, "Multi-task learning for classification with Dirichlet process priors," *Journal of Machine Learning Research*, vol. 8, pp. 35–63, 2007.

[4] T. Ferguson, "A Bayesian analysis of some nonparametric problems," *The Annals of Statistics*, vol. 1, pp. 209–230, 1973.

[5] S. R. Waterhouse and A. J. Robinson, "Classification using hierarchical mixtures of experts," in *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing IV*, 1994, pp. 177–186.

[6] D. B. Dunson and J.-H. Park, "Kernel stick-breaking processes," *Biometrika*, vol. 95(2), pp. 307–323, 2008.

[7] J. Sethuraman, "A constructive definition of Dirichlet priors," *Statistica Sinica*, vol. 1, pp. 639–650, 1994.

[8] H. Ishwaran and L. F. James, "Gibbs sampling methods for stick-breaking priors," *Journal of the American Statistical Association*, vol. 96, pp. 161–173, 2001.

[9] M. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge Univ. Press, Cambridge, U.K., 2000.

[10] M. E. Tipping, "The relevance vector machine," in *Advances in Neural Information Processing Systems (NIPS)*, T. K. Leen S. A. Solla and K. R. Müller, Eds. 2000, vol. 12, pp. 652–658, MIT Press.

[11] Y. W. Teh, M. J. Beal M. I. Jordan, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the American Statistical Association*, vol. 101, pp. 1566–1581, 2006.

[12] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz, "UCI repository of machine learning databases," *http://www.ics.uci.edu/∼mlearn/MLRepository.html*, 1998.