

---

# Learning Multiple Classifiers with Dirichlet Process Mixture Priors

---

**Ya Xue\*, Xuejun Liao, Lawrence Carin**  
Dept. of Electrical and Computer Engineering  
Duke University  
Durham, NC 27708  
{yx10,xjliao,lcarin}@ee.duke.edu

**Balaji Krishnapuram**  
Siemens Medical Solutions USA, Inc.  
Malvern, PA 19355  
balaji.krishnapuram@siemens.com

**Introduction** A real world classification task can often be viewed as consisting of multiple subtasks. In remote sensing, for example, one may have multiple sets of radar images, each collected at a particular geographical location, with the aim of designing classifiers for detecting objects of interest in images at all locations. In this situation, one can either learn a single classifier from simple pooling of images from different locations; or learn multiple classifiers, each for a particular location and based on using images from that location only. Unfortunately, neither of the two are optimal, because the first ignores the difference between different locations and the second ignores the analogy between them.

The above example represents a typical instance of a general learning scenario called “multitask learning (MTL)” [4]. The MTL is distinct from standard learning in two major aspects: the tasks are not identical, thus simply pooling them and treating them as a single task is not proper; the tasks are dependent on each other, thus isolating them and treating them as independent tasks is not appropriate.

The fact that the tasks are dependent implies that what is learned from one task is transferable to another correlated task. By learning the tasks in parallel under a unified representation, the transferability of expertise between tasks is exploited to the benefits of all tasks. This expertise transfer is particularly important in the situation for which the training data of each task are scarce. By using the data of related tasks, the training set of each task is strengthened and the generalization of the resulting classifier is improved.

A major challenge in multitask learning is to find a representation of the tasks that simultaneously characterizes the differences and similarities between them. In the existing multitask representations, each task typically has its own parameters to capture its characteristics. What is different is the approach to modelling the between-task similarities, i.e., the way in which the tasks are related to each other.

In hierarchical Bayesian models, the between-task similarities are reflected in a common prior distribution placed on the model parameters of individual tasks [2, 8, 12]. This ap-

---

\*The first author was at Siemens Medical Solutions USA for part of the time when this algorithm was developed.

proach is based on the assumption that all tasks are equally related to each other. In many applications, this assumption is too simplistic to be valid, because it does not account for heterogeneity among tasks and the corresponding data generating distributions.

Bakker and Heskes introduced an approach of exploiting relationships between tasks by task clustering and gating [1]. The common prior was modelled as a mixture of distributions. There are two issues in this method: first, the number of mixture components has to be known *a priori*; second, extra “high-level” features, other than those used for learning individual task parameters, are needed to decide the relative weights of mixture components.

In this paper, we propose a nonparametric Bayesian model for jointly learning multiple classifiers, each corresponding to a task and an associated dataset. In particular, we employ the Dirichlet Process Mixture (DPM) as the common prior on the model parameters of the tasks. The model automatically identifies task clusters via Bayesian inference. The main advantage of a nonparametric model is that it makes no assumptions regarding the underlying distributions, and therefore it provides a richer and more flexible representation than its parametric counterparts.

Exact Bayesian inference of DPM using Gibbs sampling can be extremely expensive in computation. In this work, we employ the variational Bayesian (VB) method as an efficient approximate method to learn the posterior distributions of the multitask model parameters. Compared to Gibbs sampling, VB is computationally more efficient and allows one to analyze large datasets with tens of thousands of samples.

It is noted that our model is different from that in [10], although the two models both take a nonparametric approach. The model in [10] groups *identical* tasks together, while our model investigates the *similarity* between tasks. In [10], the relationship between tasks is represented in a binary form (identical or different), with this too restrictive in practice. Our model is more general and includes the model in [10] as a special case.

**Proposed Hierarchical Bayesian Model of Multitask Classifiers** Consider  $M$  tasks indexed as  $1, \dots, M$ . Let the dataset of task  $m$  be  $\mathcal{D}_m = \{(\mathbf{x}_{m,n}, y_{m,n}) : n = 1, \dots, N_m\}$ , where  $\mathbf{x}_{m,n} \in \mathbb{R}^d$ ,  $y_{m,n} \in \{0, 1\}$ , and  $(\mathbf{x}_{m,n}, y_{m,n})$  are drawn i.i.d. from the underlying distribution of task  $m$ . For task  $m$ ,  $m = 1, \dots, M$ , the conditional distribution of  $y_{m,n}$  given  $\mathbf{x}_{m,n}$  is modelled via logistic regression,

$$p(y_{m,n} | \mathbf{w}_m, \mathbf{x}_{m,n}) = \sigma(\mathbf{w}_m^T \mathbf{x}_{m,n})^{y_{m,n}} (1 - \sigma(\mathbf{w}_m^T \mathbf{x}_{m,n}))^{1-y_{m,n}} \quad (1)$$

where  $\mathbf{w}_m$  parameterizes the classifier of task  $m$ . The goal is to learn  $\{\mathbf{w}_m\}_{m=1}^M$  jointly so that the resulting classifiers can accurately predict class labels for new test samples  $\mathbf{x}_{m,n}$  for tasks  $m = 1, \dots, M$ . The hierarchical model of  $\{\mathbf{w}_m\}_{m=1}^M$  is specified as

$$\mathbf{w}_m | \boldsymbol{\theta}_m \sim F(\mathbf{w}_m | \boldsymbol{\theta}_m), \quad \boldsymbol{\theta}_m | G \sim G, \quad G \sim DP(\alpha, G_0) \quad (2)$$

where  $DP(\alpha, G_0)$  is a Dirichlet process with precision parameter  $\alpha$  and base distribution  $G_0$ . The Dirichlet process is used to account for the uncertainty of  $G$ . Using Sethuraman’s stick-breaking representation [9, 5], we can write

$$G = \sum_{k=1}^{\infty} \pi_k(\mathbf{v}) \delta_{\boldsymbol{\theta}_k^*}, \quad \text{where } \pi_k(\mathbf{v}) = v_k \prod_{i=1}^{k-1} (1 - v_i), \quad v_k \sim \text{Beta}(1, \alpha), \quad \boldsymbol{\theta}_k^* \sim G_0 \quad (3)$$

It is clear from (2) and (3) that

$$\mathbf{w}_m | \{\mathbf{c}_m, \boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_\infty^*\} \sim \prod_{k=1}^{\infty} (F(\mathbf{w}_m | \boldsymbol{\theta}_k^*))^{c_{m,k}} \quad (4)$$

where  $\mathbf{c}_m = [c_{m,1}, \dots, c_{m,\infty}]^T$  is a cluster membership indicator defined as:  $c_{m,k} = 1$  if  $\mathbf{w}_m$  belongs to cluster  $k$ ;  $c_{m,k} = 0$  otherwise. The clustering structure of  $\mathbf{w}_m$  represents relationships among tasks.

Given the cluster membership indicators  $\mathbf{c}_m$  and the cluster parameters  $\theta_k^* = \{\boldsymbol{\mu}_k, \boldsymbol{\lambda}_k\}$ ,  $k = 1, \dots, \infty$ , we let  $\mathbf{w}_m$  be drawn from an infinite Gaussian mixture,

$$p(\mathbf{w}_m | \mathbf{c}_m, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_\infty, \boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_\infty) = \prod_{k=1}^{\infty} ((2\pi)^{-\frac{d}{2}} |\boldsymbol{\Lambda}_k|^{\frac{1}{2}} \exp(-\frac{1}{2}(\mathbf{w}_m - \boldsymbol{\mu}_k)^T \boldsymbol{\Lambda}_k (\mathbf{w}_m - \boldsymbol{\mu}_k)))^{c_{m,k}} \quad (5)$$

where  $\boldsymbol{\mu}_k$  is mean of the  $k$ th cluster and the precision matrix  $\boldsymbol{\Lambda}_k = \text{diag}[\boldsymbol{\lambda}_k]$  with  $\boldsymbol{\lambda}_k \in \mathbb{R}^d$ . If the diagonal elements of  $\boldsymbol{\Lambda}_k$  all go to infinity, the Gaussian mixture becomes a sum of point mass functions, which is the model in [10], therefore the model in [10] can be seen as a special case of our model.

The cluster membership indicators  $\mathbf{c}_m$  are multinomial distributed with parameters  $\{\pi_1, \dots, \pi_\infty\}$ ,

$$p(\mathbf{c}_m | v_1, \dots, v_\infty) = \sum_{m=1}^M (v_1^{c_{m,1}} \prod_{k=2}^{\infty} (v_k \prod_{i=1}^{k-1} (1 - v_i))^{c_{m,k}}) \quad (6)$$

where  $v_k \sim \text{Beta}(1, \alpha)$ . Here  $G_0$  is characterized with a Normal-Gamma distribution

$$p(\boldsymbol{\mu}_k, \boldsymbol{\lambda}_k | \boldsymbol{\eta}_0, \beta_0, \gamma_{10}, \gamma_{20}) = (2\pi)^{-\frac{d}{2}} |\beta_0 \boldsymbol{\Lambda}_k|^{\frac{1}{2}} \exp(-\frac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{\eta}_0)^T \beta_0 \boldsymbol{\Lambda}_k (\boldsymbol{\mu}_k - \boldsymbol{\eta}_0)) \times \prod_{j=1}^d (\frac{\gamma_{20}^{\gamma_{10}}}{\Gamma(\gamma_{10})} \lambda_{k,j}^{\gamma_{10}-1} \exp(-\gamma_{20} \lambda_{k,j})) \quad (7)$$

where  $\boldsymbol{\eta}_0$  and  $\beta_0$  is mean and scaling parameter of the Normal part;  $\gamma_{10}$  is the shape parameter and  $\gamma_{20}$  is the inverse scale parameter in the Gamma part.

A Gamma distribution prior is placed on the DP scaling parameter  $\alpha$

$$p(\alpha | \tau_{10}, \tau_{20}) = \frac{\tau_{20}^{\tau_{10}}}{\Gamma(\tau_{10})} \alpha^{\tau_{10}-1} \exp(-\tau_{20} \alpha), \quad (8)$$

where  $\tau_{10}$  is the shape parameter and  $\tau_{20}$  is the inverse scale parameter.

For simplicity, let  $Z$  collect all latent variables, i.e.,  $Z = \{(\mathbf{w}_m)_{m=1}^M, (\mathbf{c}_m)_{m=1}^M, (\boldsymbol{\mu}_k)_{k=1}^{\infty}, (\boldsymbol{\lambda}_k)_{k=1}^{\infty}, (v_k)_{k=1}^{\infty}, \alpha\}$ ; similarly the hyperparameters are collected in  $\Phi = \{\boldsymbol{\eta}_0, \beta_0, \gamma_{10}, \gamma_{20}, \tau_{10}, \tau_{20}\}$ .

**Variational Bayesian (VB) Inference** We are interested in estimating  $p(Z | \mathcal{D}_1, \dots, \mathcal{D}_M, \Phi)$ , the posterior distribution of the latent variables given the observed data and the hyperparameters. By Bayes rule,

$$p(Z | \mathcal{D}_1, \dots, \mathcal{D}_M, \Phi) = \frac{p(\mathcal{D}_1, \dots, \mathcal{D}_M | Z, \Phi) p(Z | \Phi)}{p(\mathcal{D}_1, \dots, \mathcal{D}_M | \Phi)}, \quad (9)$$

where  $p(\mathcal{D}_1, \dots, \mathcal{D}_M | \Phi) = \int p(\mathcal{D}_1, \dots, \mathcal{D}_M | Z, \Phi) p(Z | \Phi) dZ$  is the marginal distribution. We use the variational Bayesian (VB) method [7] to approximate the true posterior  $p(Z | \mathcal{D}_1, \dots, \mathcal{D}_M, \Phi)$  by a variational distribution  $q(Z)$ . To make the VB tractable, we assume that  $q(Z)$  is fully factorized and belongs in the conjugate-exponential family, i.e.,

$$q(Z) = (\prod_{m=1}^M q_{\mathbf{w}_m}(\mathbf{w}_m)) (\prod_{m=1}^M q_{\mathbf{c}_m}(\mathbf{c}_m)) (\prod_{k=1}^K q_{\boldsymbol{\mu}_k, \boldsymbol{\lambda}_k}(\boldsymbol{\mu}_k, \boldsymbol{\lambda}_k)) (\prod_{k=1}^K q_{v_k}(v_k)) q(\alpha). \quad (10)$$

where, for  $m = 1, \dots, M$  and  $k = 1, \dots, K$ ,  $q_{\mathbf{w}_m}(\mathbf{w}_m) = N(\boldsymbol{\nu}_m, \boldsymbol{\Gamma}_m)$ ,  $q_{\mathbf{c}_m}(\mathbf{c}_m) = \text{Mult}(\phi_{m,1}, \dots, \phi_{m,K})$ ,  $q_{\boldsymbol{\mu}_k, \boldsymbol{\lambda}_k}(\boldsymbol{\mu}_k, \boldsymbol{\lambda}_k) = N(\boldsymbol{\eta}_k, (\beta_k \boldsymbol{\Lambda}_k)^{-1}) \prod_{j=1}^d \text{Ga}(\gamma_{1,k,j}, \gamma_{2,k,j})$  with  $\boldsymbol{\Lambda}_k = \text{diag}[\boldsymbol{\lambda}_k]$ ,  $q_{v_k}(v_k) = \text{Beta}(\varphi_{1,k}, \varphi_{2,k})$ , and  $q(\alpha) = \text{Ga}(\tau_1, \tau_2)$ .

One difficulty in applying VB inference to our multitask learning model is that the sigmoid function in (1) does not belong to the conjugate-exponential family. To overcome this, we use a variational method based on bounding log convex functions [6]

$$\sigma(x) \geq \sigma(\xi) \exp(\frac{x - \xi}{2} - \lambda(\xi)(x^2 - \xi^2)), \quad (11)$$

where  $\lambda(\xi) = \frac{\frac{1}{2} - \sigma(\xi)}{2\xi}$  and  $\xi$  is a variational parameter. The equality holds at  $\xi = \pm x$ .

The VB algorithm iteratively re-estimates  $q_{\mathbf{w}_m}(\mathbf{w}_m)$ ,  $q_{\mathbf{c}_m}(\mathbf{c}_m)$ ,  $q_{\boldsymbol{\mu}_k, \boldsymbol{\lambda}_k}(\boldsymbol{\mu}_k, \boldsymbol{\lambda}_k)$ ,  $q_{v_k}(v_k)$ ,  $q(\alpha)$  and  $\xi$ . The estimation works by alternating between the parameters: each time it estimates a single set of parameters, keeping the others fixed at their current estimates. The estimation stops upon convergence of the lower bound of the log-likelihood, which is guaranteed to increase monotonically [7]. The re-estimates can be derived by using the mean-field variational approach [3].

Due to the page limit, we cannot present these re-estimate formulae here. We are currently conducting more experiments on realistic data sets to study the performance of the proposed method.

**Discussion** We set the truncation level equal to the number of tasks in the experiments presented in the technical report. It is an optimal solution but computationally expensive if there are a large number of tasks. Ishwaran and James established two theorems for selecting an appropriate truncation level which leads to a model virtually indistinguishable from the infinite DP model [5]. In the VB framework, the model is a non-truncated full Dirichlet process while the variational distribution is truncated. Empirical results on truncation level selection were given in [3], but no theoretical criterion has been developed in the VB framework up to now.

In the above discussions, we assume the tasks are defined by the application, so the problem is naturally a MTL. In certain scenarios, the task appears as a single task, but by artificially decomposing the task into a number of simpler subtasks, one artificially makes up a MTL problem. The artificial MTL problem is useful when each subtask is simpler and easier to solve than the original problem. Consider regression as an example. If each sample in the dataset is treated as a task and the precision of Gaussian mixture components goes to infinity, i.e., the prior of  $\mathbf{w}_m$  in (5) becomes a sum of infinite number of point mass functions, then the MTL model is exactly the model for Bayesian curve fitting with Dirichlet process mixtures in Section 4 of [11]. In classification problems, it may be preferable to have each task consist of samples from multiple classes in the initial setting, and then let the algorithm group these tasks and infer their cluster structure.

## References

- [1] B. Bakker and T. Heskes. Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, 4:83–99, 2003.
- [2] J. Baxter. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, 28:7–39, 1997.
- [3] D.M. Blei and M.I. Jordan. Variational methods for the dirichlet process. In *ICML*, 2004.
- [4] R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.
- [5] H. Ishwaran and L.F. James. Gibbs sampling methods for stick breaking priors. *Journal of the American Statistical Association*, 96:161–173, 2001.
- [6] T. Jaakkola and M. Jordan. A variational approach to bayesian logistic regression models and their extensions. In *Sixth International Workshop on Artificial Intelligence and Statistics*, 1997.
- [7] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. In M.I. Jordan, editor, *Learning in Graphical Models*. MIT Press, Cambridge, 1999.
- [8] N.D. Lawrence and J.C. Platt. Learning to learn with the informative vector machine. In *ICML*, 2004.
- [9] J. Sethuraman. A constructive definition of dirichlet priors. *Statistica Sinica*, 4, 1994.
- [10] V. Tresp and K. Yu. An introduction to nonparametric hierarchical bayesian modelling with a focus on multi-agent learning. In *the Hamilton Summer School on Switching and Learning in Feedback Systems*, 2004.
- [11] M. West, P. Miller, and M.D. Escobar. Hierarchical priors and mixture models, with application in regression and density estimation. *Aspects of Uncertainty: A Tribute to D.V. Lindley*, pages 363–386, 1994.
- [12] K. Yu, A. Schwaighofer, V. Tresp, W.-Y. Ma, and H. Zhang. Collaborative ensemble learning: Combining collaborative and content-based information filtering via hierarchical bayes. In *UAI*, 2003.