Journal of Machine Learning Research 8 (2007) 35-63 Submitted 4/06; Revised 9/06; Published 1/07

Multi-Task Learning for Classification with Dirichlet Process Priors

Ya Xue YX10@EE.DUKE.EDU
Xuejun Liao XJLIAO@EE.DUKE.EDU
Lawrence Carin LCARIN@EE.DUKE.EDU
Department of Electrical and Computer Engineering
Duke University
Durham, NC 27708, USA

Balaji Krishnapuram BALAJI.KRISHNAPURAM@SIEMENS.COM
Siemens Medical Solutions USA, Inc.
Malvern, PA 19355, USA

Editor: Peter Bartlett

Abstract

Consider the problem of learning logistic-regression models for multiple classification tasks, where the training data set for each task is not drawn from the same statistical distribution. In such a multi-task learning (MTL) scenario, it is necessary to identify groups of similar tasks that should be learned jointly. Relying on a Dirichlet process (DP) based statistical model to learn the extent of similarity between classification tasks, we develop computationally efficient algorithms for two different forms of the MTL problem. First, we consider a *symmetric* multi-task learning (SMTL) situation in which classifiers for multiple tasks are learned jointly using a variational Bayesian (VB) algorithm. Second, we consider an *asymmetric* multi-task learning (AMTL) formulation in which the posterior density function from the SMTL model parameters (from previous tasks) is used as a prior for a new task: this approach has the significant advantage of not requiring storage and use of all previous data from prior tasks. The AMTL formulation is solved with a simple Markov Chain Monte Carlo (MCMC) construction. Experimental results on two real life MTL problems indicate that the proposed algorithms: (a) automatically identify subgroups of related tasks whose training data appear to be drawn from similar distributions; and (b) are more accurate than simpler approaches such as single-task learning, pooling of data across all tasks, and simplified approximations to DP.

Keywords: classification, hierarchical Bayesian models, Dirichlet process

1. Introduction

A real world classification task can often be viewed as consisting of multiple correlated subtasks. In remote sensing, for example, one may have multiple sets of data, each collected at a particular geographical location; rather than designing individual classifiers for each of these sensing tasks, it is desirable to share data across tasks to enhance overall sensing performance. This represents a typical example of a general learning scenario called multi-task learning (MTL) (Caruana, 1997), or learn to learn (Thrun and Pratt, 1998). In contrast to MTL, single-task learning (STL) refers to the approach of learning one classification task at a time, only using the corresponding data set; often STL assumes that the training samples are drawn independently from an identical distribution. MTL is distinct from standard STL in two principal respects: (i) the tasks are not identical, thus simply pooling them and treating them as a single task is not appropriate; and (ii) some of the classification tasks may be highly correlated (dependent on each other), but the strategy of isolating each task and learning the corresponding classifier independently does not exploit the potential information one may acquire from other classification tasks.

The fact that some of the classification tasks are correlated (dependent) implies that what is learned from one task is transferable to another. By learning the classifiers in parallel under a unified representation, the transferability of expertise between tasks is exploited to the benefit of all. This expertise transfer is particularly important when we are provided with only a limited amount of training data for learning each classifier. By exploiting data from related tasks, the training data for each task is strengthened and the generalization of the resulting classifier-estimation algorithm is improved.

1.1 Previous Work on MTL

Multi-task learning has been the focus of much interest in the machine learning community over the last decade. Typical approaches to information transfer among tasks include: sharing hidden nodes in neural networks (Baxter, 1995, 2000; Caruana, 1997); placing a common prior in hierarchical Bayesian models (Yu et al., 2003, 2004, 2005; Zhang et al., 2006); sharing parameters of Gaussian processes (Lawrence and Platt, 2004); learning the optimal distance metric for K-Nearest Neighbors (Thrun and O'Sullivan, 1996); sharing a common structure on the predictor space (Ando and Zhang, 2005); and structured regularization in kernel methods (Evgeniou et al., 2005), among others.

In statistics, the problem of combining information from similar but independent experiments has been studied under the category of meta-analysis (Glass, 1976) for a variety of applications in medicine, psychology and education. Researchers collect data from experiments performed at different sites or times—including information from related experiments published in literature—to obtain an overall evaluation on the significance of an experimental effect. Therefore, meta-analysis is also referred to as quantitative synthesis, or overview. The objective of multi-task learning is different from that of meta analysis. Instead of giving an overall evaluation, our objective is to learn multiple tasks jointly, either to improve the learning performance (i.e., classification accuracy) of each individual task, or to boost the performance of a new task by transferring domain knowledge learned from previously observed tasks. Despite the difference in objectives, many of the techniques employed in the statistical literature on meta-analysis can be applied to multi-task learning as well.

1.1.1 DIRICHLET PROCESSES FOR NONPARAMETRIC HIERARCHICAL BAYESIAN MODELING

Hierarchical Bayesian modeling is one of the most important methods for meta analysis (Burr and Doss., 2005; Dominici et al., 1997; Hoff, 2003; Müller et al., 2004; Mallick and Walker, 1997). Hierarchical Bayesian models provide the flexibility to model both the individuality of tasks (experiments), and the correlations between tasks. Statisticians refer to this approach as "borrowing strength" across tasks. Usually the bottom layer of the hierarchy is individual models with task-specific parameters. On the layer above, tasks are connected together via a common prior placed on those parameters. The hierarchical model can achieve efficient information-sharing between tasks for the following reason. Learning of the common prior is also a part of the training process, and data from all tasks contribute to learning the common prior, thus making it possible to transfer information between tasks (via sufficient statistics). Given the prior, individual models are learnt independently. As a result, the estimation of a classifier (task) is affected by both its own training data and by data from the other tasks related through the common prior.

Often, the common prior in a hierarchical Bayesian model is specified in a parametric form with unknown hyper-parameters, for example, a Gaussian distribution with unknown mean and variance. Information is transferred between tasks by learning those hyper-parameters using data from all tasks. However, it is preferable to also learn the functional form of the common prior from the data, instead of being pre-defined. In this paper, we provide such a nonparametric hierarchical Bayesian model for jointly learning multiple logistic regression classifiers. Such nonparametric approaches are desirable because it is often difficult to know what the true distribution should be like, and an inappropriate prior could be misleading. Further, the model parameters of individual tasks may have high complexity, and therefore no appropriate parametric form can be found easily.

In the proposed nonparametric hierarchical Bayesian model, the common prior is drawn from the Dirichlet process (DP). The advantage of applying the DP prior to hierarchical models has been addressed in the statistics literature, see for example Mukhopadhyay and Gelfand (1997), Mallick and Walker (1997) and Müller et al. (2004). Research on the Dirichlet process model goes back to Ferguson (1973), who proved that there is positive (non-zero) probability that some sample function of the DP will be as close as desired to any probability function defined on the same support set. Therefore, the DP is rich enough to model the parameters of individual tasks with arbitrarily high complexity, and flexible enough to fit them well without any assumption about the functional form of the prior distribution.

1.1.2 IDENTIFYING THE EXTENT OF SIMILARITIES BETWEEN TASKS IN MTL

A common assumption in the previous literature on MTL work is that all tasks are (equally) related to each other, but recently there have been a few investigations concerning the extent of relatedness between tasks. An ideal MTL algorithm should be able to automatically identify similarities between tasks and only allow similar tasks to share data or information. Thrun and O'Sullivan (1996) first presented a task-clustering algorithm with K-Nearest Neighbors. Bakker and Heskes (2003) model the common prior in the hierarchical model as a mixture distribution, but two issues exist in that work: (i) Extra "high-level" task characteristics, other than the features used for learning the model parameters of individual tasks, are needed to decide the relative weights of mixture components; and (ii) the number of mixtures is a pre-defined parameter. Both these issues are avoided in the models presented here. Based only on the features and class labels, the proposed statistical models automatically identify the similarities between the various tasks and adjust the complexity of the model, that is, the number of task clusters, relying on the implicit nonparametric clustering mechanism of the DP (see Sec. 2).

Before we proceed with the technical details, we clarify two issues about task-similarity. First, we define two classification tasks as similar when the two classification boundaries are close, that is, when the weight vectors of two classifiers are similar. Note that this is different from some previous work such as Caruana (1997) where two tasks are defined to be similar if they use the same features to make their decision.

Secondly, the property that the distributions drawn from a Dirichlet process are discrete with probability one introduces questions, because it implies that we cluster *identical* tasks instead of *similar* tasks. This may appear restricting, but for the following reasons this is not the case. We are interested in the posterior distribution of the model parameters when we learn a model with Bayesian methods. The posterior is decided by both the prior of the parameters and the data likelihood given the parameters. If the DP prior is employed, the prior promotes clustering while the likelihood encourages the fitting of the parameters of individual classifiers to their own data. Therefore, the model parameters learned for any task are the result of the tradeoff between sharing with other tasks and retaining the individuality of the current task. This gives similar tasks the chance to share the same model parameters.

As discussed above, the direct use of DP yields identical parameter estimates to similar tasks. One may more generally wish to give similar (but not identical) parameters to similar tasks. This may be accomplished by adding an additional layer of randomness, that is, modeling the prior of the model parameters as a DP mixture (DPM) (Antoniak, 1974), which is a usual solution to the discrete character of DP. We have compared the models with the DPM prior and the DP prior and found that the former underperforms the latter in all our empirical experiments. We hypothesize that this result is attributable to the increase in model complexity, since the DPM prior introduces one more layer in the hierarchical model than the DP prior. Therefore, in this paper we only present the approach of placing a DP prior on the the model parameters of individual tasks.

1.2 Novel Contributions of this Paper

We develop novel solutions to the following problems: (i) joint learning of multiple classification tasks, which may differ in data statistics due to temporal, geographical or other variations; (ii) efficient transfer of the information learned from (i) to a new observed task for the purpose of improving the new task's learning performance. For notational simplification, problem (i) is referred as SMTL (symmetric multi-task learning) and problem (ii) as AMTL (asymmetric multi-task learning). Our discussion is focused on classification tasks, although the proposed methods can be extended directly to regression problems.

The setting of the SMTL is similar to that used in meta analysis. Most meta-analysis work considers only simple linear models for regression problems. Mukhopadhyay and Gelfand (1997) and Ishwaran (2000) discuss the application of DP priors in the general framework of generalized linear models (GLMs). They use the mixed random effects model to capture heterogeneity in the population studied in an experiment. The fixed effects are modeled with a parametric prior while the random effects are mixed over DP. Our work is closely related to their research, in that the logistic regression model, which we use for classification, is a special case of GLMs. Yet, we modify the model to suit the multi-task setting. First, we group the population by task and add an additional constraint that each group share the same model. Second, we place a DP prior on the whole vector of linear coefficients, including both fixed effects and random effects. The linear coefficients of covariates in the logistic regression model correspond to the classifiers in linear classification problems. For our problem, it is too limiting to consider only the variability in the random effect, that is, the intercept of the classification boundary; the variability in the orientation of the classification boundary should be considered as well.

The inference techniques employed here constitute a major difference between our SMTL part and the work of Mukhopadhyay and Gelfand (1997) and Ishwaran (2000). Mukhopadhyay and Gelfand (1997) use Gibbs sampling based on the Polya Urn representation of DP, while Ishwaran (2000) develops a Gibbs sampler based on the truncated stick-breaking representation of DP. In the work reported here, the proposed SMTL models are implemented with a deterministic inference method, specifically variational Bayesian (VB) inference, avoiding the overfitting associated with maximum-likelihood (ML) approximations while preserving computational efficiency.

The AMTL problem addressed here is a natural extension of SMTL. In AMTL the computational burden is relatively small and thus a means of Monte Carlo integration with acceptance-rejection is suggested to make predictions for the new task. We note that Yu et al. (2004) propose a hierarchical Bayesian framework for information filtering, which similarly applies a DP prior on individual user profiles. However, due to limitations of the approximate DP prior employed in Yu et al. (2004), their approach differs from ours in two respects: (i) it cannot be used to improve the classification performance of multiple tasks by learning them in parallel, that is, it is not a solution to the SMTL problem; and (ii) in the AMTL case, their approach conducts an effective information transfer to the new observed task only when the size of the training set is large in previous tasks. These points are clarified further in Section 4.

1.3 Organization of the Paper

The remainder of this paper is organized as follows. Section 2 provides an introduction to the Dirichlet process and its application to data clustering. The SMTL problem is addressed in Section 3, which describes the proposed Bayesian hierarchical framework and presents a variational inference algorithm. Section 4 develops an efficient method of information transfer for the AMTL case and compares it with the method presented by Yu et al. (2004). Experimental results are reported in Section 5, demonstrating the application of the proposed models to a landmine detection and an art image retrieval problem. Section 6 concludes the work and outlines future directions.

2. Dirichlet Process

Assume the model parameters of individual tasks, denoted by w, are drawn from a common prior distribution G. The distribution G itself is sampled from a Dirichlet process $DP(\alpha, G_0)$, where α is a positive scaling (innovation) parameter and G_0 is a base distribution. The mathematical representation of the DP model is

$$\begin{aligned} w_m|G &\sim G, \\ G &\sim DP(\alpha, G_0). \end{aligned}$$

where $m = 1, \ldots, M$ for M tasks.

Integrating out G, the conditional distribution of w_m, given observations of the other $M-1$ w values $w_{-m} = \{w_1, \cdots, w_{m-1}, w_{m+1}, \cdots, w_M\}$, is

$$p(w_m|w_{-m}, \alpha, G_0) = \frac{\alpha}{M-1+\alpha} G_0 + \frac{1}{M-1+\alpha} \sum_{j=1, j\neq m}^{M} \delta_{w_j}, \tag{1}$$

where δ_{w_j} is the distribution concentrated at the single point w_j.

Let w_k^*, $k = 1, \ldots, K$, denote the K distinct values among $w_1, \ldots, w_M$ and $n_{-m,k}$ denote the number of w's equal to w_k^*, excluding w_m. Equation (1) can be rewritten as

$$p(w_m|w_{-m}, \alpha, G_0) = \frac{\alpha}{M-1+\alpha} G_0 + \frac{1}{M-1+\alpha} \sum_{k=1}^{K} n_{-m,k} \delta_{w_k^*}. \tag{2}$$

Relevant observations concerning (2) are: (i) the Dirichlet process model has an implicit mechanism of clustering samples into groups, since a new sample prefers to join a group with a large population; (ii) the model is flexible, creating new clusters or merging existing clusters to fit the observed data better; (iii) parameter α controls the probability of creating a new cluster, with larger α yielding more clusters; (iv) in the limit as $\alpha \to \infty$ there is a cluster for each w, and since w are drawn from G_0, $\lim_{\alpha\to\infty} G = G_0$.

The Dirichlet Process has the following important properties:

1. $E[G] = G_0$; the base distribution represents our prior knowledge or expectation concerning G.

2. $p(G|w_1, \ldots, w_M, \alpha, G_0) = DP(\alpha + M, \frac{\alpha}{M+\alpha} G_0 + \frac{1}{M+\alpha} \sum_{j=1}^{M} \delta_{w_j})$; the posterior of G is still a Dirichlet process with the updated base distribution $\frac{\alpha}{M+\alpha} G_0 + \frac{1}{M+\alpha} \sum_{j=1}^{M} \delta_{w_j}$, and a confidence in this base distribution that is enhanced relative to the confidence in the original base distribution G_0, reflected in the increased parameter $\alpha \to \alpha + M$.

The lack of an explicit form of G is addressed with the stick-breaking view of the Dirichlet process. Sethuraman (1994) introduces a constructive definition of the Dirichlet process, based upon which Ishwaran and James (2001) characterize the DP priors with a stick-breaking representation

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{w_k^*}, \tag{3}$$

where

$$\pi_k = v_k \prod_{i=1}^{k-1} (1 - v_i).$$

For each k, v_k is drawn from a Beta distribution $Be(1, \alpha)$;[1] simultaneously another random variable w_k^* is drawn independently from the base distribution G_0; w_k^* and π_k represents the location and weight of the kth stick.

If v_K is set to one instead of being drawn from the beta distribution, it yields a truncated approximation to the Dirichlet process

$$G = \sum_{k=1}^{K} \pi_k \delta_{w_k^*}.$$

Ishwaran and James (2001) establish two theorems for selecting an appropriate truncation level, leading to a model virtually indistinguishable from the infinite DP model; the truncated DP model is computationally more efficient in practice.

3. Learning Multiple Classification Tasks Jointly (SMTL)

In this section we propose a Bayesian multi-task learning model for jointly estimating classifiers for several data sets. The model automatically identifies relatedness by task clustering with nonparametric methods. A variational Bayesian (VB) approximation is used to learn the posterior distributions of the model parameters.

[1] Notation for distributions follows Robert and Casella (2004); this same notation is used in the rest of the paper.

3.1 Mathematical Model

Consider M tasks, indexed as $1,\cdots,M$. Let the data set of task m be $\mathcal{D}_m = \{(x_{m,n}, y_{m,n}) : n = 1,\cdots,N_m\}$, where $x_{m,n} \in \mathbb{R}^d$, $y_{m,n} \in \{0,1\}$, and $(x_{m,n}, y_{m,n})$ are drawn i.i.d. from the underlying distribution of task m. For task m the conditional distribution of $y_{m,n}$ given $x_{m,n}$ is modeled via logistic regression as,

$$p(y_{m,n}|w_m, x_{m,n}) = \sigma(w_m^T x_{m,n})^{y_{m,n}}[1-\sigma(w_m^T x_{m,n})]^{1-y_{m,n}} \tag{4}$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ and w_m parameterizes the classifier for task m. The goal is to learn $\{w_m\}_{m=1}^M$ jointly, sharing information between tasks as appropriate, so that the resulting classifiers can accurately predict class labels for new test samples for tasks $m = 1,\cdots,M$.

To complete the hierarchical model, we place a Dirichlet process prior on the parameters w_m, which based on the discussion in Section 2 implies clustering of the tasks. The base distribution G_0 is specified as a d-dimensional multivariate normal distribution $N_d(\mu,\Sigma)$. We define an indicator variable $c_m = [c_{m,1},\ldots,c_{m,\infty}]^T$, which is an all-zero vector except that the kth entry is equal to one if task m belongs to cluster k, using the infinite set of mixture components (clusters) reflected in (3). The data can be seen as drawn from the following generative model, obtained by employing the stick-breaking view of DP:

SMTL Model. Given the parameters α, μ and Σ,

1. Draw v_k from the Beta distribution $Be(1,\alpha)$ and independently draw w_k^* from the base distribution $N_d(\mu,\Sigma)$, $k = 1,\ldots,\infty$.

2. $\pi_k = v_k \prod_{i=1}^{k-1}(1-v_i)$, $k = 1,\ldots,\infty$.

3. Draw the indicators $(c_{m,1},\ldots,c_{m,\infty})$ from a multinomial distribution $M_\infty(1;\pi_1,\ldots,\pi_\infty)$, $m = 1,\ldots,M$

4. $w_m = \prod_{k=1}^{\infty}(w_k^*)^{c_{m,k}}$, or in an equivalent form $w_m = \sum_{k=1}^{\infty} c_{m,k} w_k^*$, $m = 1,\ldots,M$.

5. Draw $y_{m,n}$ from a Binomial distribution $B(1,\sigma(w_m^T x_{m,n}))$, $m = 1,\ldots,M$, $n = 1,\cdots,N_m$.

We refer to this as symmetric multi-task learning (SMTL) because all tasks are treated symmetrically; asymmetric multi-task learning (AMTL) is addressed in Section 4.

3.1.1 HYPER-PRIORS

In the SMTL model, α, μ and Σ are given parameters. The scaling parameter α often has a strong impact on the number of clusters, as analyzed in Section 2. To make the algorithm more robust, it is suggested by West et al. (1994) that α be integrated over a diffuse hyper-prior. This leads to a modified SMTL model:

SMTL-1 Model. Given the parameters μ, Σ and hyper-parameters τ_{10}, τ_{20},

- Draw α from a Gamma distribution $Ga(\tau_{10},\tau_{20})$.

- Follow Step 1-5 in the SMTL model.

Similarly, we are also interested in the effects of integrating μ and Σ, the parameters of the base distribution, over a diffuse prior. For notational simplification, we use Λ, the precision matrix of the base distribution, instead of the covariance matrix Σ, where $\Lambda = \Sigma^{-1}$. We assume Λ is a diagonal matrix with diagonal elements $\lambda_1, \cdots, \lambda_d$. The conjugate prior for the mean and precision of a normal distribution is a Normal-Gamma distribution. Hence, the SMTL-1 model can be further modified as

SMTL-2 Model. Given the hyper-parameters τ_{10}, τ_{20}, γ_{10}, γ_{20} and β_0,

- Draw λ_j from a Gamma distribution $Ga(\gamma_{10}, \gamma_{20})$, $j = 1, ..., d$.
- Draw μ from a Normal distribution $N_d(\mathbf{0}, (\beta_0 \Lambda)^{-1})$; draw α from a Gamma distribution $Ga(\tau_{10}, \tau_{20})$.
- Follow Step 1-5 in the SMTL model.

The performance of the SMTL-1 and SMTL-2 models is analyzed in Section 5.1.1.

3.1.2 Graphical Representation

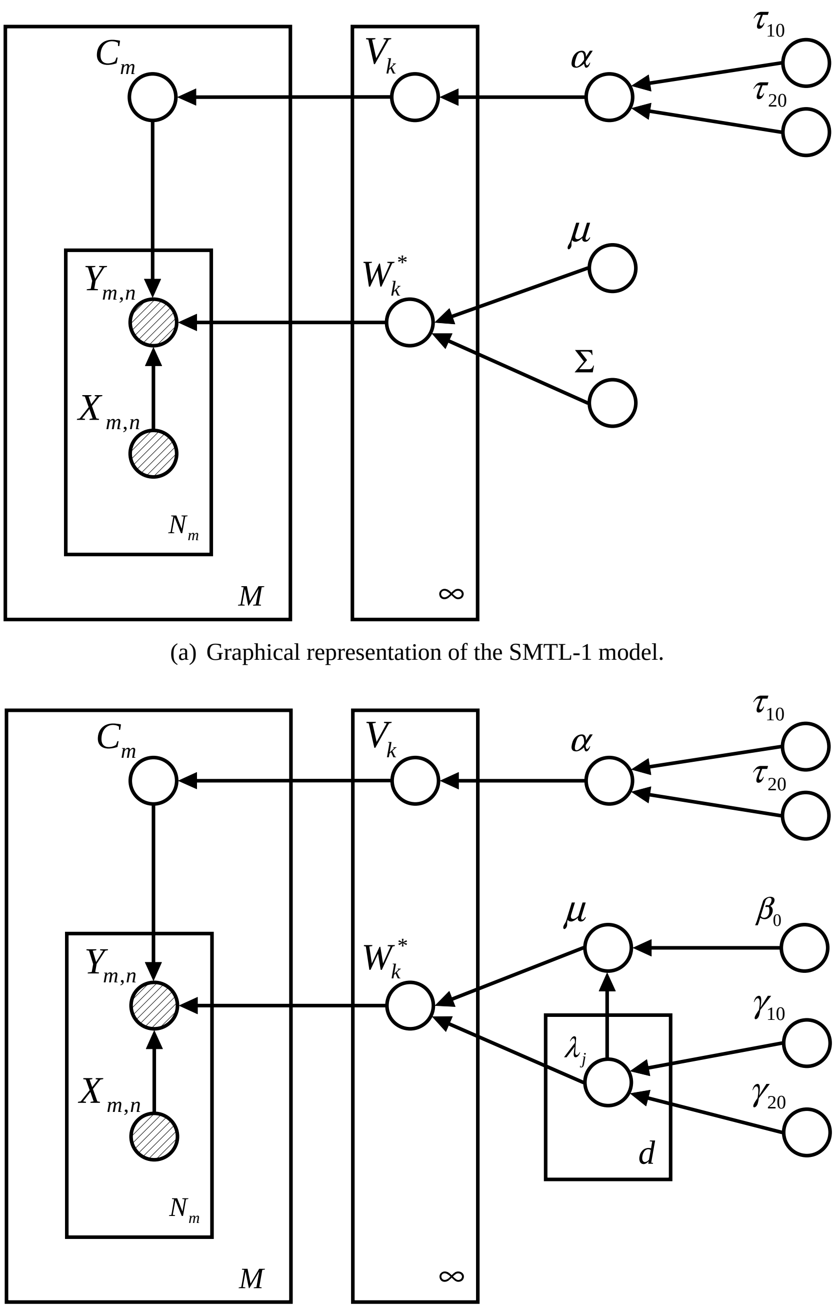
Figure 1 shows a graphical representation of the SMTL models with the Dirichlet process prior; the two SMTL models differ in the ancestor nodes of w_k^*.

In the graph, each node denotes a random variable in the model. The nodes for $y_{m,n}$ and $x_{m,n}$ are shaded because they are observed data. An arrow indicates dependence between variables, that is, the conditional distribution of a variable given its parents. The number, for example, M, at the lower right corner of a box indicates the nodes in that box have M *iid* copies.

The condition distributions between the variables are specified as follows

- $y_{m,n}$
  $p(y_{m,n}|c_m, \{w_k^*\}_{k=1}^{\infty}, x_{m,n}) = \prod_{k=1}^{\infty} \{\sigma(w_k^{*T} x_{m,n})^{y_{m,n}} [1 - \sigma(w_k^{*T} x_{m,n})]^{1-y_{m,n}}\}^{c_{m,k}}$,
  $m = 1, \ldots, M$, $n = 1, \cdots, N_m$.
- c_m
  $p(c_m|\{v_k\}_{k=1}^{\infty}) = v_1^{c_{m,1}} \prod_{k=2}^{\infty} [v_k \prod_{i=1}^{k-1} (1 - v_i)]^{c_{m,k}}$, $m = 1, \ldots, M$.
- v_k
  $p(v_k|\alpha) = \alpha(1 - v_k)^{\alpha-1}$, $k = 1, \ldots, \infty$,
- α
  $p(\alpha|\tau_{10}, \tau_{20}) = \frac{\tau_{20}^{\tau_{10}}}{\Gamma(\tau_{10})} \alpha^{\tau_{10}-1} \exp(-\tau_{20}\alpha)$, where $\Gamma(\cdot)$ is the Gamma function.
- w_k^*
  · SMTL-1 Model:
  $p(w_k^*|\mu, \Sigma) = (2\pi)^{-\frac{d}{2}} |\Sigma|^{-\frac{1}{2}} \exp(-\frac{1}{2}(w_k^* - \mu)^T \Sigma^{-1} (w_k^* - \mu))$, $k = 1, \ldots, \infty$.
  · SMTL-2 Model:
  $p(w_k^*|\mu, \{\lambda_j\}_{j=1}^{d}) = (2\pi)^{-\frac{d}{2}} |\Lambda|^{\frac{1}{2}} \exp(-\frac{1}{2}(w_k^* - \mu)^T \Lambda (w_k^* - \mu))$, $k = 1, \ldots, \infty$,
  where Λ is a diagonal matrix with diagonal elements $\lambda_1, \cdots, \lambda_d$.
- $\mu, \lambda_1, \cdots, \lambda_d$ (if the SMTL-2 Model is used)

$$\begin{aligned} & p(\mu, \{\lambda_j\}_{j=1}^{d}|\gamma_{10}, \gamma_{20}, \beta_0) \\ = \; & (2\pi)^{-\frac{d}{2}} |\beta_0 \Lambda|^{\frac{1}{2}} \exp(-\frac{\beta_0}{2} \mu^T \Lambda \mu) \cdot \prod_{j=1}^{d} [\frac{\gamma_{20}^{\gamma_{10}}}{\Gamma(\gamma_{10})} \lambda_j^{\gamma_{10}-1} \exp(-\gamma_{20}\lambda_j)]. \end{aligned}$$



(a) Graphical representation of the SMTL-1 model.

(b) Graphical representation of the SMTL-2 model.

Figure 1: Graphical representation of the SMTL models with the Dirichlet process prior.

For simplicity, let **Z** denote the collection of latent variables and Φ denote the collection of given parameters and hyper-parameters. For the SMTL-1 model, $\mathbf{Z} = \{\{c_m\}_{m=1}^{M}, \{v_k\}_{k=1}^{\infty}, \alpha, \{w_k^*\}_{k=1}^{\infty}\}$

and $\Phi = \{\tau_{10}, \tau_{20}, \mu, \Sigma\}$; for the SMTL-2 model, $\mathbf{Z} = \{\{c_m\}_{m=1}^M, \{v_k\}_{k=1}^\infty, \alpha, \{w_k^*\}_{k=1}^\infty, \mu, \{\lambda_j\}_{j=1}^d\}$ and $\Phi = \{\tau_{10}, \tau_{20}, \gamma_{10}, \gamma_{20}, \beta_0\}$.

3.2 Variational Bayesian Inference

In the Bayesian approach, we are interested in $p(\mathbf{Z}|\{\mathcal{D}_m\}_{m=1}^M, \Phi)$, the posterior distribution of the latent variables given the observed data and hyper-parameters,

$$p(\mathbf{Z}|\{\mathcal{D}_m\}_{m=1}^M, \Phi) = \frac{p(\{\mathcal{D}_m\}_{m=1}^M|\mathbf{Z}, \Phi)p(\mathbf{Z}|\Phi)}{p(\{\mathcal{D}_m\}_{m=1}^M|\Phi)},$$

where $p(\{\mathcal{D}_m\}_{m=1}^M|\Phi) = \int p(\{\mathcal{D}_m\}_{m=1}^M|\mathbf{Z}, \Phi)p(\mathbf{Z}|\Phi)dZ$ is the marginal distribution, and evaluation of this integration is the principal computational challenge; the integral does not have an analytic form in most cases. The Markov Chain Monte Carlo (MCMC) method is a powerful and popular simulation tool for Bayesian inference. To date most research on applications of the Dirichlet process has been implemented with Gibbs sampling, an MCMC method (Escobar and West, 1995; Ishwaran and James, 2001). However, slow speed and difficult-to-evaluate convergence of the DP Gibbs sampler impede its application in many practical situations.

In this work, we employ a computationally efficient approach, mean-field variational Bayesian (VB) inference (Ghahramani and Beal, 2001). The VB method approximates the true posterior $p(\mathbf{Z}|\{\mathcal{D}_m\}_{m=1}^M, \Phi)$ by a variational distribution $q(\mathbf{Z})$. It converts computation of posteriors into an optimization problem of minimizing the Kullback-Leibler (KL) distance between $q(\mathbf{Z})$ and $p(\mathbf{Z}|\{\mathcal{D}_m\}_{m=1}^M, \Phi)$, which is equivalent to maximizing a lower bound of $\log p(\{\mathcal{D}_m\}_{m=1}^M|\Phi)$, the log likelihood. To make the optimization problem tractable, it is assumed that the variational distribution $q(\mathbf{Z})$ is sufficiently simple - fully factorized with each factorized component in the exponential family. Under this assumption, an analytic solution of the optimal $q(\mathbf{Z})$ can be obtained by taking functional derivatives; refer to Ghahramani and Beal (2001) and Jordan et al. (1999) for more details about VB.

3.2.1 LOCAL CONVEX BOUND

One difficulty of applying VB inference to the SMTL models is that the sigmoid function in (4) does not lie within the conjugate-exponential family. We use a variational method based on bounding log convex functions (Jaakkola and Jordan, 1997).

Consider the logistic regression model $p(y|w,x) = \sigma(w^T x)^y [1 - \sigma(w^T x)]^{1-y}$. The prior distribution of w is assumed to be normal with mean $\tilde{\mu}_0$ and variance $\tilde{\Sigma}_0$. We want to estimate the posterior distribution of w given the data (x, y). This does not have an analytic solution due to the non-exponential property of the logistic regression function. Jaakkola and Jordan (1997) present a method that uses an accurate variational transformation of $p(y|w,x)$ as follows

$$p(y|w,x) \geq \sigma(\xi)\exp\left(\frac{(2y-1)w^T x - \xi}{2} + \rho(\xi)(x^T w w^T x - \xi^2)\right),$$

where $\rho(\xi) = \frac{\frac{1}{2} - \sigma(\xi)}{2\xi}$ and ξ is a variational parameter. The equality holds when $\xi = \pm w^T x$.

The posterior $p(w|x, y, \tilde{\mu}_0, \tilde{\Sigma}_0)$ remains the normal form with this variational approximation. Given the variational parameter ξ, the mean $\tilde{\mu}$ and variance $\tilde{\Sigma}$ of the normal distribution can be computed as

$$\tilde{\Sigma} = (\tilde{\Sigma}_0^{-1} + 2|\rho(\xi)|xx^T)^{-1}, \quad \tilde{\mu} = \tilde{\Sigma}[\tilde{\Sigma}_0^{-1}\tilde{\mu}_0 + (y - \frac{1}{2})x]. \tag{5}$$

Note that we use the top script $\tilde{}$ to avoid confusion with the usage of μ and Σ as the mean and variance of the DP base distribution G_0 in other sections of this paper.

Since the optimal value of ξ depends on w, an EM algorithm is devised in Jaakkola and Jordan (1997) to optimize ξ. The E step updates $\tilde{\mu}$ and $\tilde{\Sigma}$ following (5), given the estimate of ξ in the last iteration; the M step computes the optimal value of ξ as

$$\xi^2 = x^T(\tilde{\Sigma}+\tilde{\mu}\tilde{\mu}^T)x.$$

This EM algorithm is assured to converge and has been verified to be fast and stable in Jaakkola and Jordan (1997).

The variational method can give us a lower bound of the predictive likelihood as

$$p(y|x,\tilde{\mu}_0,\tilde{\Sigma}_0) \geq \exp(\log\sigma(\xi) - \frac{\xi}{2} - \rho(\xi)\xi^2 - \frac{1}{2}\tilde{\mu}_0^T\tilde{\Sigma}_0^{-1}\tilde{\mu}_0 + \frac{1}{2}\tilde{\mu}^T\tilde{\Sigma}^{-1}\tilde{\mu} + \frac{1}{2}\log\frac{|\tilde{\Sigma}|}{|\tilde{\Sigma}_0|}). \tag{6}$$

3.2.2 TRUNCATED VARIATIONAL DISTRIBUTION

Another difficulty that must be addressed is the computational complexity of the infinite stick-breaking model. Following Blei and Jordan (2005), we employ a truncated stick-breaking representation for the variational distribution. It is worth noting that 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 are given by Blei and Jordan (2005), but no theoretical criterion has been developed in the VB framework up to now. In all the experiments presented here we set the truncation level equal to the number of tasks. There is no loss of generality with this approach but it is computationally expensive if there are a large number of tasks.

Let K denote the truncation level. In the SMTL-1 model, the factorized variational distribution is specified as

$$q(\mathbf{Z}) = [\prod_{m=1}^{M} q_{c_m}(c_m)]\cdot[\prod_{k=1}^{K} q_{v_k}(v_k)]\cdot q_\alpha(\alpha)\cdot[(\prod_{k=1}^{K} q_{w_k^*}(w_k^*)],$$

where

- $q_{c_m}(c_m)$ is a multinomial distribution,

$$c_m \sim M_K(1;\phi_{m,1},\ldots,\phi_{m,K}), m = 1,\ldots,M.$$

- $q_{v_k}(v_k)$ is a Beta distribution

$$v_k \sim Be(\varphi_{1,k},\varphi_{2,k}), k = 1,\ldots,K-1.$$

  Note $q_{v_K}(v_K) = \delta_1(v_K)$.

- $q_\alpha(\alpha)$ is a Gamma distribution

$$\alpha \sim Ga(\tau_1,\tau_2).$$

- $q_{w_k^*}(w_k^*)$ is a normal distribution,

$$w_k^* \sim N_d(\theta_k,\Gamma_k), k = 1,\ldots,K.$$

Similarly, in the SMTL-2 model, the factorized variational distribution is specified as

$$q(\mathbf{Z}) = [\prod_{m=1}^{M} q_{c_m}(c_m)] \cdot [\prod_{k=1}^{K} q_{v_k}(v_k)] \cdot q_\alpha(\alpha) \cdot [(\prod_{k=1}^{K} q_{w_k^*}(w_k^*)] \cdot q_{\mu,\{\lambda_j\}_{j=1}^d}(\mu, \{\lambda_j\}_{j=1}^d).$$

where $q_{c_m}(c_m)$, $q_{v_k}(v_k)$, $q_\alpha(\alpha)$ and $q_{w_k^*}(w_k^*)$ are the same as the specifications above.

The distribution $q_{\mu,\{\lambda_j\}_{j=1}^d}(\mu, \{\lambda_j\}_{j=1}^d)$ is Normal-Gamma,

$$(\mu, \{\lambda_j\}_{j=1}^d) \sim N_d(\eta, (\beta\Lambda)^{-1}) \prod_{j=1}^{d} Ga(\gamma_{1j}, \gamma_{2j}). \tag{7}$$

A coordinate ascent algorithm is developed for the SMTL model by applying the mean-field method (Ghahramani and Beal, 2001). Each factor in the factorized variational distribution and ξ, the variational parameter of the sigmoid function, are re-estimated iteratively conditioning on the current estimate of all the others, assuring the lower bound of the log likelihood increases monotonically until it converges. The re-estimation equations can be found in the Appendix.

3.3 Prediction for Test Samples

The prediction function for a new test sample $x_{m,\star}$ is

$$p(y_{m,\star} = 1 | c_m, \{w_k^*\}_{k=1}^K, x_{m,\star}) = \sum_{k=1}^{K} c_{m,k} \sigma({w_k^*}^T x_{m,\star}). \tag{8}$$

Integrating (8) over the variational distributions $q_{w_k^*}(w_k^*)$ and $q_{c_m}(c_m)$ yields

$$\begin{aligned} & p(y_{m,\star} = 1 | \{\{\phi_{m,k}\}_{k=1}^K\}_{m=1}^M, \{\theta_k\}_{k=1}^K, \{\Gamma_k\}_{k=1}^K, x_{m,\star}) \\ = & \sum_{k=1}^{K} \phi_{m,k} \int \sigma({w_k^*}^T x_{m,\star}) N_d(\theta_k, \Gamma_k) dw_k^*. \end{aligned} \tag{9}$$

The integral in (9) does not have an analytic form. The variational method described in Section 3.2.1 could give us a lower bound of the integral in the form of (6), if we apply the EM algorithm by taking $N_d(\theta_k, \Gamma_k)$ as the prior of w_k^* and $(x_{m,\star}, y_{m,\star} = 1)$ as the observation. However, we prefer an accurate estimate of the integral instead of the lower bound of it. In addition, the iterative EM algorithm might be inefficient in some applications that have a strict requirement to the testing speed. Therefore, we use the approximate form of the integral in MacKay (1992)

$$\int \sigma({w_k^*}^T x_{m,\star}) N_d(\theta_k, \Gamma_k) dw_k^* \approx \sigma(\frac{\theta_k^T x_{m,\star}}{\sqrt{1 + \frac{PI}{8} x_{m,\star}^T \Gamma_k x_{m,\star}}}), \tag{10}$$

where PI is the constant approximately 3.1416 (here PI is used instead of π to avoid confusion with π used in the stick-breaking representation of the DP).

We design a simple experiment to empirically evaluate accuracy of the approximation. Assume $x_{m,\star}$ is a 1-dimensional vector. Let the value of θ_k be $-10, -9.5, \cdots, 10$ and the value of Γ_k be $10^{-1}, 10^0, 10^1$ and 10^2. For any combination of the values of θ_k and Γ_k, we compare the average error between the approximation and the true value of the integration over the range of $x_{m,\star}$ from -10 to 10 with an interval of 0.5. The "true" value of the integral is approximated by the MCMC

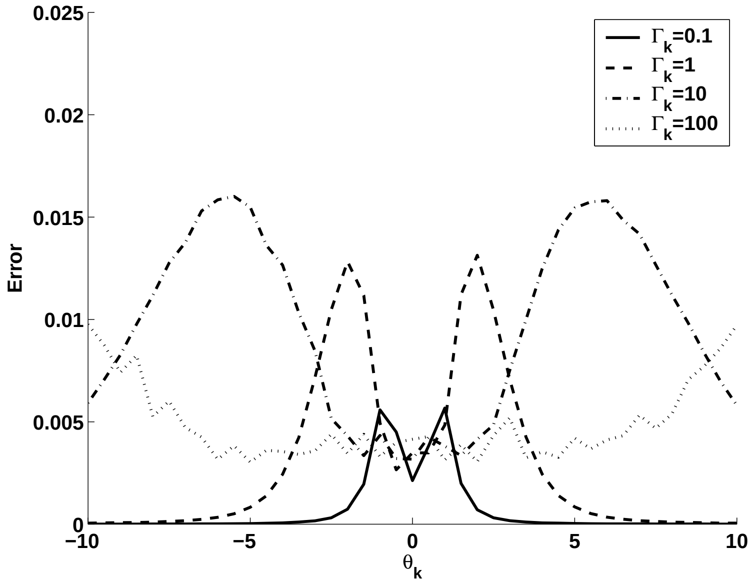


Figure 2: The error between the approximation and the true value of the integral in (10), averaged over the range of $x_{m,\star}$ from -10 to 10 with an interval of 0.5.

method, that is, we randomly draw 10^4 samples of w_k^* from the normal distribution $N_d(\theta_k, \Gamma_k)$, substitute the samples into the logistic function $\sigma(w_k^{*T} x_{m,\star})$ and take the average of the function values. The results are plotted in Fig. 2, which shows the approximation is rather accurate.

Substituting (10) into (9) yields the prediction function as

$$p(y_{m,\star} = 1|\{\{\phi_{m,k}\}_{k=1}^{K}\}_{m=1}^{M}, \{\theta_k\}_{k=1}^{K}, \{\Gamma_k\}_{k=1}^{K}, x_{m,\star}) \approx \sum_{k=1}^{K} \phi_{m,k}\sigma(\frac{\theta_k^T x_{m,\star}}{\sqrt{1+\frac{PI}{8}x_{m,\star}^T \Gamma_k x_{m,\star}}}).$$

4. Information Transfer to a New Task (AMTL)

Assume the SMTL model has been applied to M tasks. Now assume a new task $M+1$ is considered. When learning task $M+1$ we wish to benefit from the information learned from the previous M tasks. One option is to re-run the SMTL algorithm on all $M+1$ tasks, but this requires storage of the data from all previous tasks and may be computationally prohibitive for real-time applications. In fact, re-running SMTL on all tasks is in many situations not necessary, as the previous M tasks may simply represent the history and re-estimating them is uninteresting. In these cases, we need only concentrate on learning the new task, treating the previous M tasks as the background tasks, from which relevant information is transferred to the new task. Such an approach provides us the advantage of algorithmic efficiency, since we do not have to manipulate a bulk of data from the past tasks. In this section, we develop an efficient method for fast learning of the new observed task. Since task $M+1$ is treated differently from tasks 1 though M, this is referred to as asymmetric MTL, or AMTL.

4.1 Prior Learned from Previous Tasks

From (2), we know that the conditional distribution of the classifier w_{M+1} given α, G_0 and the other M classifiers is

$$p(w_{M+1}|w_1,\cdots,w_M,\alpha,G_0) = \frac{\alpha}{M+\alpha}G_0 + \frac{1}{M+\alpha}\sum_{k=1}^{K} n_k \delta_{w_k^*}, \tag{11}$$

where $n_k = \sum_{m=1}^{M} c_{m,k}$ is the number of w_m which are equal to w_k^*.

Assume the SMTL-1 model has been applied to M previous tasks and thus the variational distributions $q_{c_m}(c_m)$, $q_\alpha(\alpha)$ and $q_{w_k^*}(w_k^*)$ have been optimized. We have $E_q[c_{m,k}] = \phi_{m,k}$, $E_q[\alpha] = \frac{\tau_1}{\tau_2}$ and $E_q[w_k^*] = \theta_k$, where E_q is the expectation with respect to the variational distributions. Substituting the expectations into (11) yields

$$p(w_{M+1}|\{\{\phi_{m,k}\}_{k=1}^{K}\}_{m=1}^{M},\{\theta_k\}_{k=1}^{K},\tau_1,\tau_2,G_0) \approx \frac{\frac{\tau_1}{\tau_2}}{M+\frac{\tau_1}{\tau_2}}G_0 + \frac{1}{M+\frac{\tau_1}{\tau_2}}\sum_{k=1}^{K}\bar{n}_k\delta_{\theta_k}, \tag{12}$$

where $\bar{n}_k = \sum_{m=1}^{M}\phi_{m,k}$. For future convenience, we define $\Omega = \{\{\{\phi_{m,k}\}_{k=1}^{K}\}_{m=1}^{M},\{\theta_k\}_{k=1}^{K},\tau_1,\tau_2\}$.

Equation (12) represents our belief about the classifier w_{M+1} before we actually see the data $\mathcal{D}_{M+1}$. Therefore, by taking it as a prior for w_{M+1}, information learned from previous tasks can be transferred to the new task. The posterior of w_{M+1}, after observing the data $\mathcal{D}_{M+1}$, is computed according to the Bayes Theorem

$$p(w_{M+1}|\mathcal{D}_{M+1},\Omega,G_0) = \frac{p(\mathcal{D}_{M+1}|w_{M+1})p(w_{M+1}|\Omega,G_0)}{p(\mathcal{D}_{M+1}|\Omega,G_0)}, \tag{13}$$

where

$$\begin{aligned} & p(\mathcal{D}_{M+1}|w_{M+1}) \\ = & \prod_{n=1}^{N_{M+1}} p(y_{M+1,n}|w_{M+1},x_{M+1,n}), \\ = & \prod_{n=1}^{N_{M+1}} \sigma(w_{M+1}^T x_{M+1,n})^{y_{M+1,n}}[1-\sigma(w_{M+1}^T x_{M+1,n})]^{1-y_{M+1,n}}. \end{aligned} \tag{14}$$

Note in Section 4 we limit the discussion on learning from previous tasks to the SMTL-1 model, for which the parameters of G_0 are given; the approach developed in this section can be extended to the SMTL-2 model by substituting the expectations on the parameters of G_0 into (12).

4.2 Sampling Posterior Using Metropolis-Hastings Algorithm

The posterior (13) does not have an analytic form because the prior (12) is a mixture of the base distribution G_0 with several point mass distributions, and the sigmoid function in the likelihood function (14) is not conjugate to the prior. Considering that the computational burden is small in the AMTL case (we only deal with data from task $M+1$), we appeal to MCMC methods and develop a Metropolis-Hastings algorithm to draw samples from the posterior (Robert and Casella, 2004; Neal, 1998). This algorithm is feasible in practice since it is a simple MCMC solution and the computational cost is low.

Metropolis-Hastings Algorithm

1. Draw a sample $\dot{w}$ from (12).
2. Draw a candidate $\hat{w}$ also from (12).
3. Compute the acceptance probability
   $a(\hat{w}, \dot{w}) = \min[1, \frac{p(\mathcal{D}_{M+1}|\hat{w})}{p(\mathcal{D}_{M+1}|\dot{w})}]$.
4. Set the new value of $\dot{w}$ to $\hat{w}$ with this probability; otherwise let the new value of $\dot{w}$ be the same as the old value.
5. Repeat Step 2-4 until the required number of samples, denoted by N_{SAM}, are taken.

This yields an approximation to the posterior in (13)

$$p(w_{M+1}|\mathcal{D}_{M+1}, \Omega, G_0) \approx \frac{1}{N_{SAM}} \sum_{i=1}^{N_{SAM}} \delta_{\dot{w}_i}.$$

4.3 Prediction Algorithm for Test Samples in New Task

Our goal is to learn w_{M+1} so that the resulting classifier can accurately predict the class label for a new test sample $x_{M+1,\star}$. The prediction function is

$$\begin{aligned} & p(y_{M+1,\star} = 1|x_{M+1,\star}, \Omega, G_0) \\ = \; & \int p(y_{M+1,\star} = 1|x_{M+1,\star}, w_{M+1}) p(w_{M+1}|\mathcal{D}_{M+1}, \Omega, G_0) dw_{M+1}, \\ \approx \; & \tfrac{1}{N_{SAM}} \sum_{i=1}^{N_{SAM}} p(y_{M+1,\star} = 1|x_{M+1,\star}, \dot{w}_i), \\ = \; & \tfrac{1}{N_{SAM}} \sum_{i=1}^{N_{SAM}} \sigma\left(\dot{w}_i^T x_{M+1,\star}\right). \end{aligned} \tag{15}$$

Hence, the entire learning procedure for a new task is
AMTL-1 Algorithm Given Ω and G_0,

1. Compute the parameters in (12) - the prior learned from previous M tasks.
2. Draw samples $\dot{w}_1, \cdots, \dot{w}_{N_{SAM}}$ with the Metropolis-Hastings algorithm.
3. Predict for test samples using (15).

4.4 Comparison with Method of Yu et al. (2004)

Yu et al. (2004) present a hierarchical Bayesian framework for information filtering. Their purpose is to find the right information item for an active user, utilizing both item content information and an accumulated database of item ratings cast by a large set of users. The problem is modeled as a classification problem by labeling the items a user likes “1” and “0” otherwise. The learning situation is similar to our AMTL case, if each user is treated as a task. In Yu’s approach, a Dirichlet process prior is learned from the database

$$p(w_{M+1}|\{D_m\}_{m=1}^{M}, \alpha_0, G_0) = \frac{\alpha_0}{M+\alpha_0} G_0 + \frac{1}{M+\alpha_0} \sum_{m=1}^{M} \zeta_m \delta_{\hat{w}_m}, \tag{16}$$

where α_0 and G_0 are pre-defined. They treat $\hat{w}_m$ as the maximum a posteriori (MAP) estimate of task m. The weights ζ_m are learned with an Expectation Maximization (EM) algorithm presented in Yu et al. (2004, Section 3).

From the stick-breaking view of DP, the prior in (16) is an approximation to the standard DP, in that the locations of sticks are fixed, at the classifiers learned from individual models, while the weights are inferred with the EM algorithm. This approximation is inappropriate if the training samples in each previous task are not sufficient, so that each task cannot learn an accurate classifier by only using the corresponding user's profile. In other words, it is not a good use of the information in previous tasks.

Some other approximations are made in the prediction step of Yu's approach, although they are relatively trivial compared to the approximation to the DP prior mentioned above. For comparison, we develop the second AMTL algorithm with the approximate DP prior.

AMTL-2 Algorithm Given $\hat{w}_1, \cdots, \hat{w}_M$, α_0 and G_0,

1. Optimize the weights ζ_m in (16) with the EM algorithm in Yu et al. (2004, Section 3).

2. Substitute (12) with (16), then draw samples $\dot{w}_1, \cdots, \dot{w}_{N_{SAM}}$ with the Metropolis-Hastings algorithm.

3. Predict for test samples using (15).

Empirical comparisons of the two AMTL algorithms are reported in Section 5.1.2.

5. Experiments and Results Analysis

An empirical study of the proposed methods is conducted on two real applications: (i) a landmine detection problem, and (ii) an art image retrieval problem.

5.1 Landmine Detection

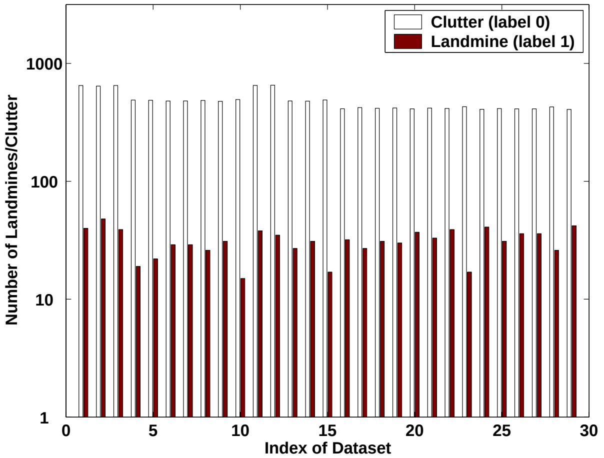


Figure 3: Number of landmines/clutter in each of the 29 data sets.

Data from 29 tasks are collected from various landmine fields.[2] Each object in a given data set is represented by a 9-dimensional feature vector and the corresponding binary label (1 for landmine and 0 for clutter). The feature vectors are extracted from radar images, concatenating four moment-based features, three correlation-based features, one energy ratio feature and one spatial variance feature. Figure 3 shows the number of landmines/clutter in each data set.

The landmine detection problem is modeled as a binary classification problem. The objective is to learn a classifier from the labeled data, with the goal of providing an accurate prediction for an unlabeled feature vector. We treat classification of each data set as a learning task and evaluate the proposed SMTL and AMTL methods on this landmine detection problem.

Among these 29 data 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. We first evaluate the SMTL models using data sets 1-10 and 16-24; next, for the AMTL setting, these data are treated as previous tasks and data sets 11-15 and 25-29 are treated as new observed tasks.

5.1.1 SMTL

Data sets 1-10 and 16-24 are used for the SMTL experiment, so there are a total of 19 tasks. We examine the performance of four methods on accuracy of label prediction: (i) SMTL-1, (ii) SMTL-2, (iii) the STL method—learn each classifier using the corresponding data set only—with the variational approach to logistic regression models in Section 3.2.1, and (iv) simply pooling the data in all tasks and then learning a single classifier with the variational approach as for (iii).

The performance is measured by average AUC on 19 tasks, where AUC denotes area under the Receiver Operation Characteristic (ROC) curve. A larger AUC value indicates a better classification performance. To have a comprehensive evaluation, we test the algorithms with different sizes of training sets. The number of training samples for every task is set as 20, 40, $\cdots$, 300. For each task, the training samples are randomly chosen from the corresponding data set and the remaining samples are used for testing. Since the data have severely unbalanced labels, as shown in Fig. 3, we have a special setting that assures there is at least one “1” and one “0” sample in the training set of each task.

Hyper-parameter settings are as follows: (i) $\tau_{10} = 5e^{-2}$, $\tau_{20} = 5e^{-2}$, $\gamma_{10} = 1e^{-2}$, $\gamma_{20} = 1e^{-3}$ and $\beta_0 = 1e^{-2}$, (ii) $\tau_{10} = 5e^{-2}$, $\tau_{20} = 5e^{-2}$, $\mu = \mathbf{0}$ and $\boldsymbol{\Sigma} = 10\mathbf{I}$, (iii) and (iv) $\tilde{\mu}_0 = \mathbf{0}$ and $\tilde{\boldsymbol{\Sigma}}_0 = 10\mathbf{I}$. We also tested with other choices of hyper-parameters and found that the algorithms are not sensitive to the hyper-parameter settings as long as the hyper-priors are rather diffuse.

We plot the results of 100 random runs in Fig. 4. The two SMTL methods generally outperform the STL method and simple pooling. To gain insight into how the SMTL method identifies the clustering structure of tasks, we calculate the between-task similarity matrix as follows: at a certain setting of the size of training set (e.g., 20 training samples per task), for each random run, the SMTL algorithm outputs the variational distribution $q_{c_m}(c_m)$ with the optimized variational parameters $\{\phi_{m,k}\}_{k=1}^{K}$, where $\phi_{m,k}$ indicates the probability that task m belongs to cluster k, and then we take $k_m^* = \arg\max_k \phi_{m,k}$ as the membership of task m. The element at the ith row and jth column in the between-task similarity matrix records number of occurrences, among 100 random runs, that task i and task j are grouped into the same cluster. Fig. 5 shows the Hinton diagram (Hinton and Sejnowski, 1986) for the between-task similarity matrices corresponding different experiment

[2] The data are available at http://www.ee.duke.edu/∼lcarin/LandmineData.zip.

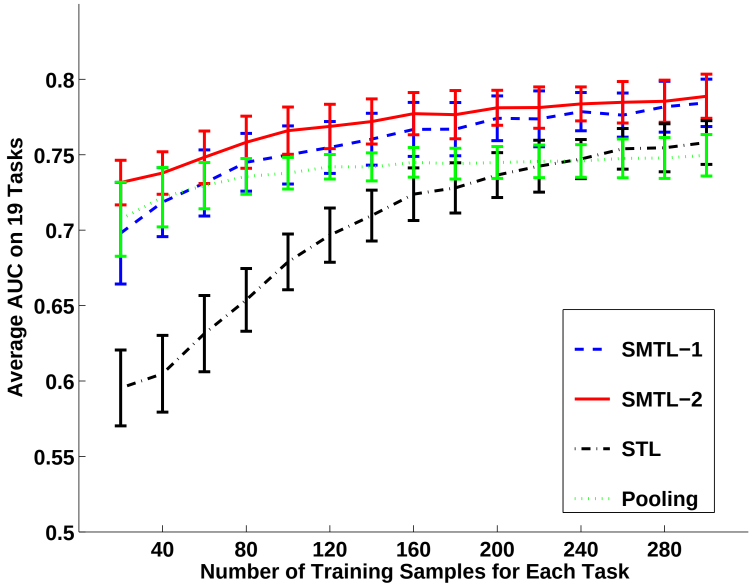


Figure 4: Average AUC on 19 tasks in the landmine detection problem.

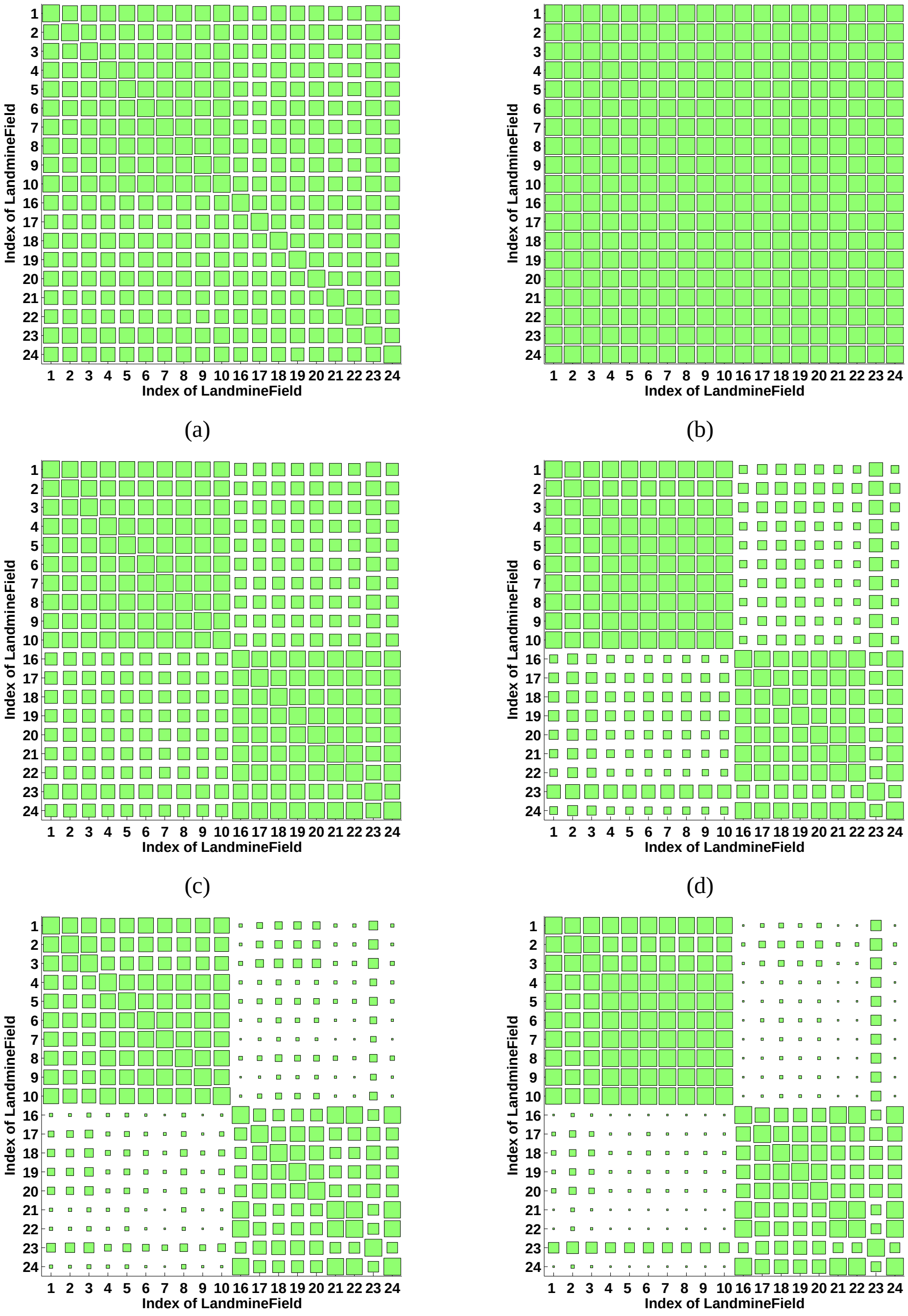
settings: (a)(c)(e) learned by using the SMTL-1 model with 20, 100 and 300 training samples per task; (b)(d)(f) learned by using the SMTL-2 model with 20, 100 and 300 training samples per task. In a Hinton diagram, the size of blocks is proportional to the value of the corresponding matrix elements.

With the help of Fig. 5, we have the following analysis on the behavior of the curves in Fig. 4.

1. When very few training samples are available, for example, 20 per task, the STL method performs poorly. The simple pooling method significantly improves the performance because its effective training size is 18 times larger than that for each individual task. The training samples are so few that although both SMTL methods find that all tasks are similar, they cannot identify the extent of similarity between tasks (see (a) and (b) in Fig. 5). As a result, they perform similar to the simple pooling method. The SMTL-2 performs slightly better due to additional robustness introduced by integrating over the parameters of G_0.

2. When there are a few training samples available, for example, 100 per task, the simple pooling method does not improve further as more training samples are pooled together, because it ignores the statistical differences between tasks. Both SMTL methods begin to learn the clustering structure (see (c) and (d) in Fig. 5) and this leads to better performance than the simple pooling. The clustering structure in (d) is more obvious than that in (c), therefore the SMTL-2 method works slightly better than the SMTL-1 method.

3. When each task has many training samples, for example, 300 per task, both SMTL methods identify the clustering structure (see (e) and (f) in Fig. 5). The number of training samples is large enough for each task to learn well by itself, so the curve for the STL method approaches the curves for the SMTL methods and exceeds the curve for the simple pooling. It is clear



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5: Hinton diagram for the between-task similarity matrix in the landmine detection problem, (a)(c)(e) learned by using the SMTL-1 model with 20, 100 and 300 training samples per task; (b)(d)(f) learned by using the SMTL-2 model with 20, 100 and 300 training samples per task. In a Hinton diagram, the size of blocks is proportional to the value of the corresponding matrix elements.

in (e) and (f) that the 19 tasks are roughly grouped into two clusters, which agree with the ground truth discussed above.

5.1.2 AMTL

In this experiment, the data sets used in Section 5.1.1 are treated as previous tasks and data sets 11-15 and 25-29 as the new observed tasks. We compare three approaches:

1. AMTL-1: First we apply the SMTL-1 model to those previous tasks and learn the variational parameters Ω, and then run the AMTL-1 algorithm on each new task.

2. AMTL-2: First we apply the variational logistic regression approach in Section 3.2.1 to each previous task and learn the individual classifiers $\hat{w}_1,\ldots,\hat{w}_M$, and then run the AMTL-2 algorithm on each new task.

3. STL: Each new task learns by itself.

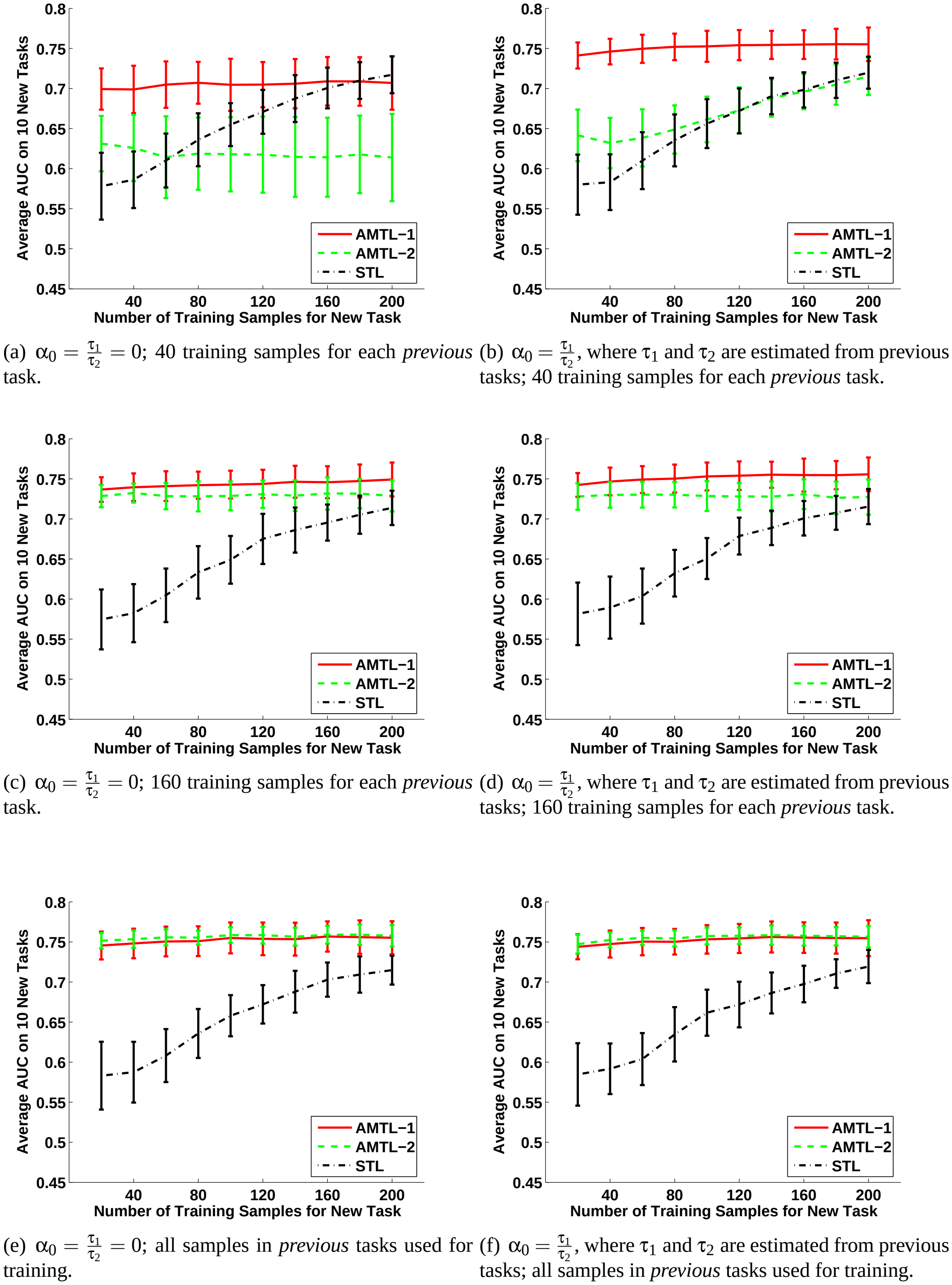
In the two AMTL methods, the mean and variance of the base distribution G_0 are specified as $\mu = \mathbf{0}$ and $\Sigma = 10\mathbf{I}$. The parameters τ_1 and τ_2 in the AMTL-1 model can be estimated from previous tasks, while α_0 in the AMTL-2 model is a predefined parameter, which represents a prior belief about the relatedness between the new task and previous tasks. We have two settings: (i) $\alpha_0 = \frac{\tau_1}{\tau_2} = 0$, which represents a belief that the new task is closely related to previous tasks, and (ii) $\alpha_0 = \frac{\tau_1}{\tau_2}$, where τ_1 and τ_2 are estimated from previous tasks.

The performance of the three approaches is measured by the average AUC over the 10 new tasks. Experimental results of 100 random runs are shown in Fig. 6. Two factors affecting learning performance are considered: (i) number of training samples per *previous* task, used for learning the prior of w_{M+1}, and (ii) number of training samples per *new* task. The first factor is evaluated at 40, 160 and all samples in each *previous* task used for training, corresponding to (a)(b), (c)(d) and (e)(f) in Fig. 6 respectively. The second factor is evaluated at 20, $40,\cdots$, 200 training samples in each *new* task, plotted along the horizonal axis in each subplot of Fig. 6.

We have the following observations from Fig. 6:

1. In the case $\alpha_0 = \frac{\tau_1}{\tau_2} = 0$ (see (a)(c)(e)), the prior belief is that the new task is closely related previous tasks and the purpose of this setting is to focus on the comparison of information transferability of the two AMTL approaches, given the ground truth that the *new* tasks are quite similar to some of *previous* tasks. The AMTL-1 approach efficiently transfers information from *previous* tasks to the *new* task. The SMTL method can learn an informative prior for the *new* task, with only 40 training samples for each *previous* task (see (a) in Fig. 6). The learning performance is slightly improved by using more training samples for each *previous* task. The performance has almost no change as the number of training samples in the *new* task increases, because we use the linear classier and thus the matching classifier can be found with even only a pair of “1” and “0” labeled samples.

   Information transferability of the AMTL-2 approach is weaker than the AMTL-1 approach, due to the approximate DP prior (as analyzed in Section 4.4). As a result, the more training samples in the *new* task, the more confused the algorithm is about which “stick” should be the matching classifier. This explains why the curve for the AMTL-2 approach in (a) even drops a little as the training samples in the *new* task increases. However, with a large set



(a) $\alpha_0 = \frac{\tau_1}{\tau_2} = 0$; 40 training samples for each *previous* task.

(b) $\alpha_0 = \frac{\tau_1}{\tau_2}$, where τ_1 and τ_2 are estimated from previous tasks; 40 training samples for each *previous* task.

(c) $\alpha_0 = \frac{\tau_1}{\tau_2} = 0$; 160 training samples for each *previous* task.

(d) $\alpha_0 = \frac{\tau_1}{\tau_2}$, where τ_1 and τ_2 are estimated from previous tasks; 160 training samples for each *previous* task.

(e) $\alpha_0 = \frac{\tau_1}{\tau_2} = 0$; all samples in *previous* tasks used for training.

(f) $\alpha_0 = \frac{\tau_1}{\tau_2}$, where τ_1 and τ_2 are estimated from previous tasks; all samples in *previous* tasks used for training.

Figure 6: Average AUC on 10 new tasks in the landmine detection problem.

of training samples in *previous* tasks, the AMTL-2 approach works slightly better than the AMTL-1 approach (see (e)), because the former can tell the subtle difference between very similar tasks, while the latter treats them as identical.

2. Next, we recover G_0 by using the parameters τ_1 and τ_2 estimated from *previous* tasks in the AMTL-1 model and setting $\alpha_0 = \frac{\tau_1}{\tau_2}$ in the AMTL-2 model (see (b)(d)(f)). That enables the new task to discover a new classifier by itself as well as use those learned from previous tasks. Information transferability of the AMTL-2 approach is weak when there are only a few training samples for each *previous* task, as disussed above. In such a case, the AMTL-2 approach works just as the STL approach because the *new* task has to learn by itself (see (b)). In contrast, the AMTL-1 approach benefits from *previous* tasks so that incorporation of G_0 has a relatively small effect on its performance.

5.2 Art Image Retrieval

A web survey is built to collect user ratings on 642 paintings from 30 artists.[3] A user chooses his rating of an image from “like”,“dislike” or “not sure”. Every user may give ratings only on a subset of all images. In total 203 user ratings are collected.

Our objective is to estimate a user’s preference on unrated images. We model this as a binary classification problem. Each user corresponds to a classification task. The images he rates as “like” are labeled “1”, “dislike” labeled “0” and the images with the rating “not sure” are not included. The content of an image is described by a 275-dimensional (275-D) feature vector concatenating a 256-D correlagram, a 10-D Pyramid wavelet texture and 9-D first and second color moments.

The painting image data differ with the landmine data in two respects. First, the low-level features of image content, for example, color and texture, are weak indicators of human preferences, therefore the content of an image is less helpful than ratings on that image from other users with similar interests. Second, because user preferences are very diverse, the clustering structure of tasks is expected to be more complex than that of the landmine detection tasks.

We use the 68 users who rate more than 100 images for the SMTL experiment. Then we take these as previous tasks and those 50 users who rate between 50 and 100 images are treated as new tasks in the AMTL experiment.

5.2.1 SMTL

In the SMTL experiment, two methods are compared: (i) SMTL-1, and (ii) the single-task learning (STL) method using the variational logistic regression approach in Section 3.2.1. The simple pooling method is not feasible because different users may look at the same image and give different ratings. The SMTL-2 model is also excluded, because the feature dimension (275) is high relative to the number of training samples for each task, so that it is hard to get an accurate estimation on the variational distribution of λ_j in (7), which is the precision on each feature dimension.

Hyper-parameter settings are as follows: (i) $\tau_{10} = 5e^{-2}$, $\tau_{20} = 5e^{-2}$, $\mu = \mathbf{0}$ and $\Sigma = 10\mathbf{I}$, and (ii) $\tilde{\mu}_0 = \mathbf{0}$ and $\tilde{\Sigma}_0 = 10\mathbf{I}$. Similar to the landmine experiments, the performance is measured by the average AUC on all tasks and evaluated at 10, 20, 30, 40 or 50 randomly selected training samples for each task. Figure 7 plots the results of 10 random runs.

[3] The survey is online at http://honolulu.dbs.informatik.uni-muenchen.de:8080/paintings/index.jsp.

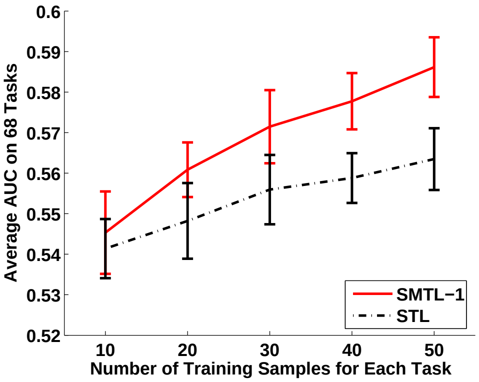


Figure 7: Average AUC on 68 tasks in the art image retrieval problem.

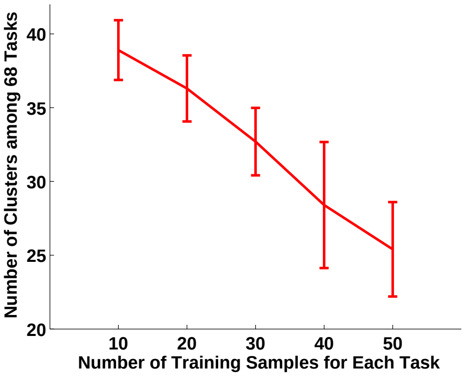


Figure 8: Number of clusters among 68 tasks in the art image retrieval problem.

As mentioned above, the tasks in the art image database are more diverse than those in the landmine data sets. To get a clear view of this, we observe the number of clusters among 68 tasks instead of the between-task similarity matrix. As in Section 5.1.1, we make a hard decision on membership of each task, for each evaluation point and each random run, and then we obtain the statistics of number of clusters among 68 tasks, which is shown in Fig. 8. When the training size is small, the algorithm weakly finds the similarity between tasks and most of the tasks learn by themselves, therefore the SMTL-1 method works similar to the single-task learning. As the number of training samples increases, the clustering structure becomes more clear and information is shared between the users/tasks with similar interests, leading to improvement in learning performance.

5.2.2 AMTL

We first apply the SMTL-1 method on the 68 tasks, using all data as training samples, to learn the prior for a new classifier, then evaluate the performance of the AMTL-1 algorithm on 50 new tasks, measured by the average AUC. The performance is compared to that of the STL approach, which means learning by the new task itself. The number of samples drawn with the Metropolis-Hastings

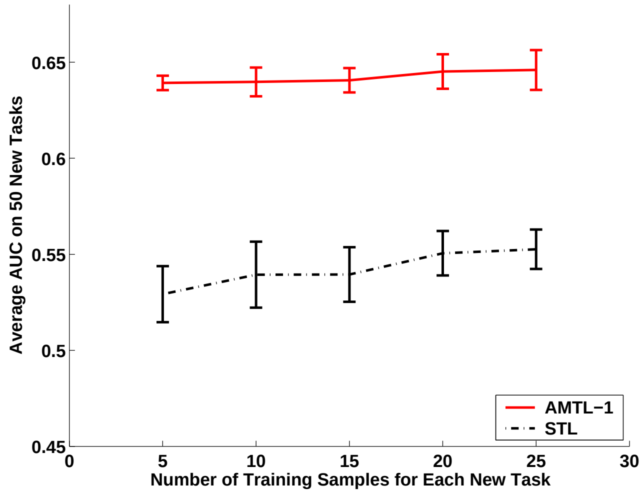


Figure 9: Average AUC on 50 new tasks in the art image retrieval problem.

algorithm in Section 4.2, N_{SAM}, is set to be $1e^3$. The results of 10 random runs are shown in Fig. 9. The curves indicate that the AMTL-1 approach outperforms the STL approach.

6. Conclusions and Future Work

A DP-based multi-task learning algorithm has been applied to the problem of designing logistic-regression classifiers for multiple tasks, for cases in which there is the potential of enhancing individual-task performance via appropriate sharing of inter-task data. Two overarching formulations have been considered. In the symmetric multi-task learning (SMTL) formulation all of the task-dependent classifiers are learned jointly. While this is a useful formulation in many cases, it requires one to store all data across previous tasks. In many cases we may undertake a new task and we would like this task to benefit from experience acquired from previous tasks, without having to return to all data observed previously. This has motivated what we have termed an asymmetric multi-task learning (AMTL) formulation. In addition to the overarching SMTL and AMTL formulations, we have considered different forms of these algorithms based on how the DP priors are handled.

MTL classification performance has been presented on two data sets: (i) a landmine sensing problem based on measured data, and (ii) an art-preference database. Concerning (i), the MTL formulation yielded a clear indication of how the data from the multiple tasks clustered into related physical phenomena. For this data we know the task-dependent environmental conditions under which the sensing was performed, and the task relatedness reflected in Hinton maps demonstrated close agreement with physical expectations. This provides a powerful confirmation of the utility of the DP formulation for a case in which “truth” is known, yielding confidence for new multi-task data sets for which the DP formulation may be used to infer truth.

In the context of the DP formulation, we considered examples for which the innovation parameter and the parameters of the base distribution were fixed, while in a separate formulation prior distributions were placed on these parameters (yielding a further layer in the Bayesian formulation). For the examples presented here we found the performance of the latter formulation to be slightly better than the former. We also compared the DP MTL formulation to several simpler learning approaches: (i) single-task learning in which no data are shared, (ii) pooling in which all data are shared indiscriminately, and (iii) a simplification to DP developed by Yu et al. (2004). For the data considered, the DP-based MTL formulations developed here outperformed these simpler approaches.

In future research we plan to extend the MTL approach to more-general settings. For example, in the MTL classifier formulation it is assumed that labeled data are available for each of the tasks. More realistically, labeled data may be available from previous tasks, but the new task under investigation may only contain unlabeled data. Based on examining the relationship of the data manifold of the new unlabeled data relative to the manifolds of the previous tasks, it may be possible to ascertain which labeled data, from previous tasks, are relevant for the new unlabeled data under test. In such a heterogeneous MTL setting, involving tasks characterized by labeled and unlabeled data, it may be possible to label the new data under test (from the new task) without requiring any associated labeled data.

In this context one may also consider an active-learning setting, in which labels are acquired selectively from the new task of interest, such that after active learning all tasks have labeled data and the MTL formulations presented here may be applied directly. In this context we note that such an approach was examined in the course of the research presented here, with active learning performed using query by committee (QBC) (McCallum and Nigam, 1998). For the data considered in this paper, we found that as long as at least one label was acquired from each of the two labels (we here considered binary labels), the MTL algorithm performed well. Consequently, while the QBC results were good, even a random acquisition of labels yielded good MTL performance, as long as at least one (randomly acquired) label existed from each of the two labels. This suggests a significant robustness of the MTL formulation, in its ability to use a small amount of labeled data for a given task to still yield good task-dependent classification performance, by appropriately sharing labeled data from other tasks. Nevertheless, this phenomenon is worth further examination, with other data sets, to further examine the utility of active learning as applied to unlabeled data from a new task (relative to simply using random sampling to determine which data to acquire labels on).

Acknowledgments

We would like to acknowledge Kai Yu and Volker Tresp for providing the art image data.

Appendix A. Re-Estimation Equations in Coordinate Ascent Algorithm

For the SMTL-1 model, the re-estimation equations are as follows

- $q_{c_m}(c_m)$:

$$\begin{aligned} s_{m,k} \;\; = \;\; & \sum_{n=1}^{N_m}[-\rho(\xi_{m,n})x_{m,n}^T(\theta_k\theta_k^T+\Gamma_k)x_{m,n}+(y_{m,n}-\tfrac{1}{2})\theta_k^T x_{m,n} \\ & +\log(\sigma(\xi_{m,n}))-\tfrac{1}{2}\xi_{m,n}+\rho(\xi_{m,n})\xi_{m,n}^2] \\ & +\mathbf{1}(k<K)[\Psi(\varphi_{1,k})-\Psi(\varphi_{1,k}+\varphi_{2,k})] \\ & +\mathbf{1}(k>1)\{\sum_{i=1}^{k-1}[\Psi(\varphi_{2,i})-\Psi(\varphi_{1,i}+\varphi_{2,i})]\} \end{aligned} ,$$

$$\phi_{m,k}=\frac{\exp(s_{m,k})}{\sum_{k=1}^{K}\exp(s_{m,k})},$$
$$m=1,\dots,M,k=1,\dots,K,$$

where $\Psi(x)=\frac{d\ln\Gamma(x)}{dx}$; $\Gamma(x)$ is the Gamma function; $\mathbf{1}(E)$ is equal to 1 if the logic expression E is true and 0 otherwise.

- $q_{v_k}(v_k)$:

$$\varphi_{1,k}=1+\sum_{m=1}^{M}\phi_{m,k},$$
$$\varphi_{2,k}=\frac{\tau_1}{\tau_2}+\sum_{m=1}^{M}\sum_{i=k+1}^{K}\phi_{m,i},$$
$$k=1,\dots,K-1.$$

- $q(\alpha)$:

$$\tau_1=\tau_{10}+K-1,$$
$$\tau_2=\tau_{20}-\sum_{k=1}^{K-1}[\Psi(\varphi_{2,k})-\Psi(\varphi_{1,k}+\varphi_{2,k})].$$

- $q_{w_k^*}(w_k^*)$:

$$\Gamma_k=[\Sigma^{-1}+2\sum_{m=1}^{M}\phi_{m,k}\sum_{n=1}^{N_m}|\rho(\xi_{m,n})|x_{m,n}x_{m,n}^T]^{-1},$$
$$\theta_k=\Gamma_k[\Sigma^{-1}\mu+\sum_{m=1}^{M}\phi_{m,k}\sum_{n=1}^{N_m}(y_{m,n}-\frac{1}{2})x_{m,n}],$$
$$k=1,\dots,K. \qquad (17)$$

- $\xi_{m,n}$:

$$\xi_{m,n}=\sqrt{\sum_{k=1}^{K}\phi_{m,k}x_{m,n}^T(\theta_k\theta_k^T+\Gamma_k)x_{m,n}},$$
$$m=1,\dots,M,n=1,\dots,N_m.$$

For the SMTL-2 model, we only need to modify (17) and add an update step for μ and $\lambda_1, \cdots, \lambda_d$

- $q_{w_k^*}(w_k^*)$:

$$\begin{aligned}
&\Gamma_k = [\Delta + 2\sum_{m=1}^{M} \phi_{m,k} \sum_{n=1}^{N_m} |\rho(\xi_{m,n})| x_{m,n} x_{m,n}^T]^{-1}, \\
&\theta_k = \Gamma_k[\Delta\eta + \sum_{m=1}^{M} \phi_{m,k} \sum_{n=1}^{N_m} (y_{m,n} - \frac{1}{2}) x_{m,n}], \\
&k = 1, \ldots, K,
\end{aligned}$$

where Δ is a diagonal matrix with diagonal elements $\frac{\gamma_{1,1}}{\gamma_{2,1}}, \ldots, \frac{\gamma_{1,d}}{\gamma_{2,d}}$.

- $q_{\mu, \{\lambda_j\}_{j=1}^d}(\mu, \{\lambda_j\}_{j=1}^d)$:

$$\begin{aligned}
&\beta = \beta_0 + K, \\
&\eta = \frac{\sum_{k=1}^{K} \theta_k}{\beta}, \\
&\gamma_{1,j} = \gamma_{10} + \frac{K}{2}, \\
&\gamma_{2,j} = \gamma_{20} + \frac{1}{2}\sum_{k=1}^{K} (\theta_{k,j}^2 + \Gamma_{k,j}) - \frac{1}{2}\beta\eta_j^2, \\
&j = 1, \ldots, d,
\end{aligned}$$

where $\theta_{k,j}$ denotes the jth element of the vector θ_k (same for η_j), and $\Gamma_{k,j}$ denotes the jth diagonal element of the matrix Γ_k.

References

R.K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817C1853, 2005.

C.E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Annals of Statistics*, 2:1152–1174, 1974.

B. Bakker and T. Heskes. Task clustering and gating for Bayesian multitask learning. *Journal of Machine Learning Research*, 4:83–99, 2003.

J. Baxter. Learning internal representations. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, 1995.

J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 2000.

D. Blei and M.I. Jordan. Variational inference for Dirichlet process mixtures. *Journal of Bayesian Analysis*, 1(1):121–144, 2005.

D. Burr and H. Doss. A bayesian semiparametric model for random-effects meta-analysis. *Journal of the American Statistical Association*, 100(469):242–251, Mar. 2005.

R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

F. Dominici, G. Parmigiani, R. Wolpert, and K. Reckhow. Combining information from related regressions. *Journal of Agricultural, Biological, and Environmental Statistics*, 2(3):294–312, 1997. Good literature review on the application of hierarchical models to meta analysis, Page 4.

M.D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588, 1995.

T. Evgeniou, C.A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

T. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1: 209–230, 1973.

Z. Ghahramani and M. Beal. Propagation algorithms for variational Bayesian learning. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*. MIT Press, Cambridge, MA, 2001.

G. V. Glass. Primary, secondary and meta-analysis of research. *Educatinal Researcher*, 5, 1976.

G.E. Hinton and T.J. Sejnowski. Learning and relearning in Boltzmann machines. In McClelland J.L. Rumelhart D.E. and the PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, chapter 7, pages 282–317. MIT Press, Cambridge, MA, 1986.

P.D. Hoff. Nonparametric modeling of hierarchically exchangeable data. Technical Report 421, University of Washington Statistics Department, 2003.

H. Ishwaran. Inference for the random effects in Bayesian generalized linear mixed models. In *ASA Proceedings of the Bayesian Statistical Science Section*, pages 1–10, 2000.

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

T.S. Jaakkola and M.I. Jordan. A variational approach to Bayesian logistic regression models and their extensions. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, 1997.

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.

N.D. Lawrence and J.C. Platt. Learning to learn with the informative vector machine. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

D.J.C. MacKay. The evidence framework applied to classification networks. *Neural Computation*, 4(5):720–736, 1992.

B.K. Mallick and S.G. Walker. Combining information from several experiments with nonparametric priors. *Biometrika*, 84(3):697–706, 1997.

A. McCallum and K. Nigam. Employing EM in pool-based active learning for text classification. In *Proceedings of the 15th International Conference on Machine Learning*, 1998.

S. Mukhopadhyay and A.E. Gelfand. Dirichlet process mixed generalized linear models. *Journal of the American Statistical Association*, 92(438):633–639, 1997.

P. Müller, F. Quintana, and G. Rosner. A method for combining inference across related nonparametric Bayesian models. *Journal of the Royal Statistical Society Series B*, 66(3):735–749, 2004.

R.M. Neal. Markov chain sampling methods for Dirichlet process mixture models. Technical Report 9815, Dept. of Statistics, University of Toronto, 1998.

C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. New York: Springer-Verlag, second edition, 2004.

J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4, 1994.

S. Thrun and J. O'Sullivan. Discovering structure in multiple learning tasks: The TC algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, 1996.

S. Thrun and L.Y. Pratt, editors. *Learning To Learn*. Kluwer Academic Publishers, Boston, MA, 1998.

M. West, P. Müller, 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.

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 *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, 2003.

K. Yu, V. Tresp, and S. Yu. A nonparametric hierarchical Bayesian framework for information filtering. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004.

K. Yu, A. Schwaighofer, and V. Tresp. Learning Gaussian processes from multiple tasks. In *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

J. Zhang, Z. Ghahramani, and Y. Yang. Learning multiple related tasks using latent independent component analysis. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*. MIT Press, Cambridge, MA, 2006.