# Factorial Multi-Task Learning : A Bayesian Nonparametric Approach

**Sunil Kumar Gupta** SUNIL.GUPTA@DEAKIN.EDU.AU
**Dinh Phung** DINH.PHUNG@DEAKIN.EDU.AU
**Svetha Venkatesh** SVETHA.VENKATESH@DEAKIN.EDU.AU
Center for Pattern Recognition and Data Analytics (PRaDA), Deakin University, VIC 3216, Australia

## Abstract

Multi-task learning is a paradigm shown to improve the performance of related tasks through their joint learning. However, for real-world data, it is usually difficult to assess the task relatedness and joint learning with unrelated tasks may lead to serious performance degradations. To this end, we propose a framework that groups the tasks based on their relatedness in a subspace and allows a varying degree of relatedness among tasks by sharing the subspace bases across the groups. This provides the flexibility of no sharing when two sets of tasks are unrelated and partial/total sharing when the tasks are related. Importantly, the number of task-groups and the subspace dimensionality are automatically inferred from the data. To realize our framework, we introduce a *novel* Bayesian nonparametric prior that extends the traditional hierarchical beta process prior using a Dirichlet process to permit potentially *infinite* number of child beta processes. We apply our model for multi-task regression and classification applications. Experimental results using several synthetic and real datasets show the superiority of our model to other recent multi-task learning methods.

## 1. Introduction

Multi-task learning aims to improve generalization performance of related tasks by joint learning (Caruana, 1997). Empirical and theoretical evidence support this claim (Ando & Zhang, 2005; Baxter, 2000; Argyriou et al., 2008; Xue et al., 2007), with applications in medical diagnosis (Bi et al., 2008), handwritten digit recognition (Kang et al., 2011) and image/video search (Wang et al., 2009). In real-world situations, where there is limited data, it is useful to combine related tasks and exploit common statistical strengths for improved prediction. A common approach is to assume similarity across tasks - the task predictors may belong to a low dimensional subspace or manifold (Argyriou et al., 2008; Rai & Daumé III, 2010; Agarwal et al., 2010), form clusters (Xue et al., 2007; Kang et al., 2011; Passos et al., 2012) or share a generative process (Daumé III, 2009). However, given a pool of so called "related" tasks, it is not easy to assess the degree of similarity/relatedness. If tasks are not related or related minimally, joint learning may degrade performance – a phenomenon widely known as negative transfer learning (Rosenstein et al., 2005). Therefore, automatically inferring task relatedness is crucial to the success of multi-task learning.

Although a critical problem, there had been few works on separating the unrelated tasks from the set of related tasks. The problem is hard since the tasks usually have varying degree of relatedness. One approach represents task predictors (or parameters) in a low dimensional subspace (Argyriou et al., 2008; Rai & Daumé III, 2010; Agarwal et al., 2010). However, a single subspace assumes all the tasks are related, and this may cause performance degradation when tasks are unrelated. Dealing with this problem, some works divide tasks into groups and learn one subspace for each group (Kang et al., 2011; Passos et al., 2012). Modeling tasks in this way is extreme in that the tasks in different groups can not influence each other. This is because real world tasks can rarely be categorized as totally "related" or "unrelated". Instead, there exists a varying degree of relatedness. This problem was partially tackled by (Kumar & Daumé III, 2012), who propose a model that learns a subspace whose bases are shared across tasks and the relatedness between tasks is determined by the number of shared bases. However, being a parametric model, this model needs *a priori* specification of parameters: the num-

*Proceedings of the 30$^{th}$ International Conference on Machine Learning*, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

ber of task groups and the dimensionality of the subspace. The performance of this method crucially depends on these parameters and it is hard to estimate them for the real data. Therefore, *a nonparametric extension of this model* is required. The work in (Passos et al., 2012) is a related Bayesian nonparametric MTL model - however, it does not allow sharing between tasks across groups, and thus is not flexible to capture varying degree of relatedness. (Gupta et al., 2012a;b) model data from multiple groups through a subspace allowing sharing of bases. However, their model is suitable only for *unsupervised* learning and requires the number of groups and group membership of data points to be specified *a priori*.

Addressing this gap, we propose a Bayesian nonparametric MTL framework that groups tasks based on their relatedness in a low dimensional subspace. The assumption is that the when tasks are related, task predictors lie close in subspace. To model varying degrees of sharing across tasks, we use a joint factor modeling approach that allows task predictors to have both shared and individual subspace bases. We refer to the approach as *factorial multi-task learning* (F-MTL). Since our goal is to model a set of $T$ task predictors jointly, we employ hierarchical factor analysis – a modeling paradigm that can jointly model the data from multiple groups through a subspace such that some of the subspace bases are shared across groups while the others are individual to a group. However, using hierarchical factor analysis in its standard form requires the tasks to be grouped in advance – something that is *unknown* for multi-task learning problem. To address this, we extend the hierarchical factor analysis for modeling tasks whose grouping is unknown. The task membership to the groups and the number of groups are inferred by clustering the tasks in the subspace through a Dirichlet process (DP), while the subspace dimensionality and the sharing configurations are inferred using a hierarchical beta process (HBP). For an optimal solution, the two processes are unified by coupling the DP prior with the HBP prior, leading to a *novel* Bayesian nonparametric prior termed as *generalized hierarchical beta process* (G-HBP).

The proposed model is applied in *two* settings : multi-task regression and classification. The model inference is done using Gibbs sampling. Experimental results using several synthetic and real-world datasets show the superiority of the proposed model to recent state-of-the-art multi-task learning methods. Our main contributions are

- A *novel* Bayesian nonparametric prior extending the HBP using a Dirichlet process to permit potentially *infinite* number of child beta processes.
- A Bayesian nonparametric, multi-task learning framework that allows joint learning of multiple tasks with varying degrees of relatedness and demonstrations on real data.
- Inference using a novel combination of Gibbs sampler and Laplace approximation.

The significance of our approach is that the number of task-groups, the subspace dimensionality and the usage of bases by different groups are automatically inferred from data. In addition, the Bayesian nonparametric priors keep the model flexible to allow each task group to also have its *own* set of factors and therefore every group *need not* necessarily share factors. This feature is the key to overcome the problem of any negative inductive bias due to unrelated tasks. This leads to a flexible model that can be applied freely on a pool of related/unrelated tasks without any performance degradation, exploiting statistical strengths from even marginally related tasks.

## 2. Background

### 2.1. Dirichlet Process Mixture (DPM) Model

Dirichlet process (DP) (Ferguson, 1973) has been widely used in Bayesian mixture models for clustering applications. It provides a Bayesian nonparametric prior over clustering partitions enabling a mixture model to accommodate infinitely many components. For a given set of observations, the active set of components are finite and can be inferred from the posterior distribution. Assuming that we have a set of observations $\{c_t\}_{t=1}^{T}$ with the corresponding mixture component parameters as $\{\psi_t\}_{t=1}^{T}$ where $\psi_t$'s are realizations from a Dirichlet process $G$ with concentration parameter $\xi_0$ and base measure $G_0$. Using a parametric distribution $F_c(\psi)$ for data, we can write

$$c_t \mid \psi_t \sim F_c(\psi_t),\ \psi_t \sim G,\ G \sim \text{DP}(\xi_0, G_0) \qquad (1)$$

Using a constructive definition of DP (Sethuraman, 1994), the measure $G$ can also be written as $G = \sum_{j=1}^{\infty} \mu_j \delta_{\pi_j}$. To relate $\{\psi_1, \ldots, \psi_T\}$ with $\{\pi_1, \ldots, \pi_J\}$, we can use an indicator $u_t$ for each $\psi_t$ such that $u_t = j$ if $\psi_t = \pi_j$. In applications of DPM to clustering, $u_t$ represents the cluster indicator for data $c_t$ and we have

$$c_t \mid u_t, \{\pi_1, \ldots, \pi_J\} \sim F_c(\pi_{u_t}) \qquad (2)$$

### 2.2. Beta Process

In factor analysis, inferring the number of factors requires us defining a prior on the usage probabilities of factors. Under Bayesian framework, this can be realized using a *Bernoulli process* prior that is parametrized by a *beta process* (Thibaux & Jordan, 2007). Formally, a beta process $B$ is a completely random measure implying that for any $r$ disjoint sets $S_1, \ldots, S_r \in \Omega$ (where $\Omega$ is a measurable space with sigma field $\mathcal{F}$), $B(S_1), \ldots, B(S_r)$ are independent and the draws from the beta process $B$ are discrete with probability one (Kingman, 1967). We denote the beta process as $B \sim \text{BP}(\gamma_0, B_0)$, where $\gamma_0 > 0$ is referred to as a *concentration parameter* and $B_0$ is a *base measure* with total mass $B_0(\Omega) = \tau_0$. In set function notation, we can write $B = \sum_k \beta_k \delta_{\phi_k}$ where $\{\phi_k, \beta_k\}$ are drawn from a non-homogeneous Poisson process defined on the product space $\Omega \times [0,1]$. The distribution over weight $\beta_k$ follows

$$p(\beta_k) = \text{beta}(\gamma_0 B_0(\phi_k), \gamma_0(1 - B_0(\phi_k))) \qquad (3)$$

For a discrete $B_0$, i.e. $B_0 = \sum_k p_k \delta_{\phi_k}$, we have $B_0(\phi_k) = p_k$. For a continuous $B_0$, $\beta_k$ are drawn from a Poisson process with the following rate measure

$$\nu(d\beta, d\phi) = \gamma_0 \beta^{-1} (1-\beta)^{\gamma_0 - 1} d\beta B_0(d\phi) \qquad (4)$$

Fixing $\gamma_0$ to one, $\beta_k$ can be sampled using the stick-breaking construction (Teh et al., 2007), i.e.

$$\beta_k \sim \text{StickIBP}(\tau_0) \qquad (5)$$

The beta process defined above can be used to parametrize a Bernoulli process, which then can be used to infer the number of factors in the factor analysis. Formally, let $\mathbf{z}_t$ be a draw from a Bernoulli process, i.e. $\mathbf{z}_t \sim \text{BeP}(B)$, then we have $\mathbf{z}_{t,k} \sim \text{Bernoulli}(\beta_k)$. If $\mathbf{Z}$ is defined to be a collection of $\{\mathbf{z}_t\}_{t=1}^{T}$, the posterior samples of $\mathbf{Z}$ given data gives us an estimate of which factors out of an infinite set are required to explain the data – indirectly inferring the number of factors automatically from the data.

### 2.3. Hierarchical Beta Process and Factor Analysis

A major strength of probabilistic modeling is to be able to express the dependencies through hierarchies. Building a hierarchy over Dirichlet processes, (Teh et al., 2006) proposed hierarchical Dirichlet process (HDP) that allows data form multiple groups to share a common set of parameters. Motivated by the construction of HDP, (Thibaux & Jordan, 2007) developed a similar hierarchy over beta processes called as hierarchical beta process (HBP). Formally,

$$B \sim \text{BP}(\gamma_0, B_0), A_j \sim \text{BP}(\alpha_j, B), \mathbf{Z}_j^{t,:} \sim \text{BeP}(A_j) \qquad (6)$$

where $\mathbf{Z}_j^{t,:}$ denotes $t$-th data point from $j$-th group. Similar to the use of beta process in nonparametric factor analysis, hierarchical beta process is used for a nonparametric hierarchical factor analysis (NHFA) (Gupta et al., 2012a) – a model that can jointly model data from multiple groups (or sources). Given data $\{\mathbf{X}_j\}_{j=1}^{J}$ from $J$-groups, hierarchical factor analysis (also encountered in shared subspace learning (Gupta et al., 2011; 2013)) is carried out as

$$\mathbf{X}_j = \Phi \mathbf{H}_j^{\mathsf{T}} + \mathbf{E}_j,\ j = 1, \ldots, J \quad \textit{(known grouping)} \qquad (7)$$

where $\Phi = [\phi_1, \ldots, \phi_K]$ contains the subspace bases, and $\mathbf{H}_j$ denotes the subspace representations for $\mathbf{X}_j$. Carrying out a joint factor analysis in this manner allows some of the bases to be shared across the groups while keeping others individual to a group. Being a Bayesian nonparametric model, NHFA can infer the number of shared and individual bases using a hierarchical beta-Bernoulli process prior (Thibaux & Jordan, 2007). For this, $\mathbf{H}_j$ is decomposed as $\mathbf{H}_j = \mathbf{Z}_j \odot \mathbf{W}_j$ where $\mathbf{W}_j$ contains the actual subspace representations and $\mathbf{Z}_j$ is a binary matrix with its $(t,k)$-th element indicating the presence or absence of the basis $\phi_k$ for the $t$-th data point of $j$-th group. We note that (6) can be used as a prior over $\mathbf{Z}_1, \ldots, \mathbf{Z}_J$ and given the data, posterior distribution can be used to get an estimate of the number of shared/individual bases.

## 3. Multi-Task Learning Framework

In this section, we describe a our framework for multi-task learning. Our goal is to jointly model the tasks with varying degree of relatedness in such a way that related tasks strengthen each other while unrelated tasks do not affect themselves. The underlying assumption is that the when tasks are related, the tasks predictors closely lie in a subspace. For this, we use a joint factor modeling approach that allows task predictors to have both shared and individual subspace bases. We refer to our approach as *factorial multi-task learning* (F-MTL). Let us assume we have $T$ tasks, indexed as $t = 1, \ldots, T$ and $t$-th task has labeled training examples denoted as $D_t = \{(\mathbf{x}_{ti}, y_{ti}) \mid i = 1, \ldots, N_t\}$ where $\mathbf{x}_{ti} \in \mathbb{R}^{M \times 1}$ and we define $\mathbf{X}_t = [\mathbf{x}_{t1}, \ldots, \mathbf{x}_{tN_t}]$ and $\mathbf{y}_t = [y_{t1}, \ldots, y_{tN_t}]^{\mathsf{T}}$. Let the task predictor (the regression or classification weights) of $t$-th task be denoted as $\theta_t$ where $\theta_t \in \mathbb{R}^{M \times 1}$. We use $\Theta$ to collectively define all the task predictors, i.e. $\Theta = [\theta_1, \ldots, \theta_T]$.

Since our goal is to model a set of $T$ task predictors jointly, we employ hierarchical factor analysis – a modeling paradigm that can jointly model the data from multiple groups through a subspace such that some of the subspace bases are shared across groups while the others are individual to a group. However, using hierarchical factor analysis in its standard form requires the tasks to be grouped *in advance* – something that is *unknown* for multi-task learning problem. To address

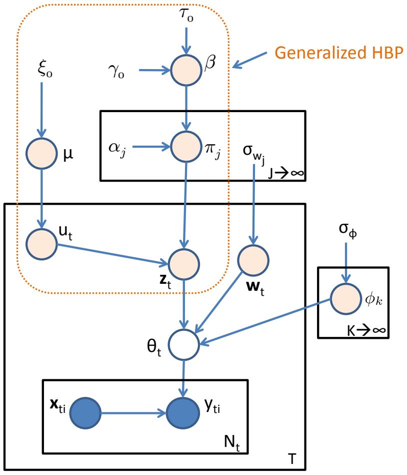


Figure 1: Directed graphical representation for the proposed factorial multi-task learning (F-MTL). We have $T$ tasks with $t$-th task having $N_t$ examples. The task predictors $\{\theta_t\}_{t=1}^T$ are clustered in $J$ groups via a Dirichlet process (DP) and represented in a subspace having the bases $\{\phi_k\}_{k=1}^K$ that are shared across groups.

this problem, we shall extend the hierarchical factor analysis for modeling a set of tasks whose grouping is unknown. If the task groupings are *known* (i.e. a partition of $\Theta$ having $J$ groups as $\{\Theta_1, \ldots, \Theta_J\}$), following (7) we can model the task predictors as

$$\Theta_j = \Phi \mathbf{H}_j^\mathsf{T} + \mathbf{E}_j, \ j = 1, \ldots, J \tag{8}$$

where $\Phi = [\phi_1, \ldots, \phi_K]$ contains basis vectors of a subspace (spanned by the column of $\Phi$) and $\mathbf{H}_j$ is the representation of the task predictors of $j$-th group (i.e. $\Theta_j$) in the subspace. The matrix $\mathbf{E}_j$ represent modeling errors. We note that $K$ is overall subspace dimensionality. For $j$-th task group, some of the basis vectors may not be used and thus, its effective dimensionality $K_j \le K$. For real world applications, the values of $K$ and $K_j$ are hard to guess and require *model selection.* Conventional model selection approaches are computationally expensive and wasteful of data (Corduneanu & Bishop, 2001). The subspace dimensionalities can be automatically inferred using a hierarchical beta process prior over $\Phi$ and $\mathbf{H}_j$ (Gupta et al., 2012a). In particular, $\mathbf{H}_j$ can be written as an element-wise product $\mathbf{H}_j = \mathbf{W}_j \odot \mathbf{Z}_j$ where $\mathbf{W}_j$ denotes the subspace representations and $\mathbf{Z}_j$ is a binary matrix denoting the presence or absence of a basis vector. Using Bernoulli process priors over matrices $\mathbf{Z}_{1:J}$ in combination with the task-predictor likelihood, an estimate of $K$ and $K_j$ is obtained from the posterior distribution. The $g$-th vector of $j$-th group, $\mathbf{z}_{g,j}$ is drawn as

$$B \sim \mathrm{BP}(\gamma_0, B_0), A_j \sim \mathrm{BP}(\alpha_j, B), \mathbf{z}_{g,j} \sim \mathrm{BeP}(A_j) \tag{13}$$

where $B = \sum_k \beta_k \delta_{\phi_k}$, $A_j = \sum_k \pi_{jk} \delta_{\phi_k}$ and $A_j$ are i.i.d. draw from a beta process. When the task groupings are *unknown*, we can use a group indicator variable $u_t$ such that $u_t \in \{1, \ldots, J\}$ *and* $u_t$ *is to be inferred from data.* For this, we use a Dirichlet process (DP) as a prior to induce clusters over $\mathbf{z}_t$, creating an *infinite mixture of Bernoulli processes.* Formally

$$\mu \sim \mathrm{GEM}(\xi_0), \quad u_t \sim \mathrm{Discrete}(\mu) \tag{14}$$

where $\mathrm{GEM}(\xi_0)$ denotes the well-known stick-breaking construction for DP. Since $t$-th task predictor belongs to the group indexed by $u_t$, we further use $g_{u_t}$ to index its position in $u_t$-th group. To keep the notation simple, we write $g_{u_t}$ just as $g_t$. As an example, consider 4 tasks indexed as $t = 1, 2, 3, 4$; assuming a partition $(1, 4)$ and $(2, 3)$, we have $(u_1 = 1, g_1 = 1)$, $(u_2 = 2, g_2 = 1)$, $(u_3 = 2, g_3 = 2)$, $(u_4 = 1, g_4 = 2)$. Using this notation

$$\mathbf{z}_{g_t, u_t} \mid u_t, A_{1:J} \sim \mathrm{BeP}(A_{u_t}) \tag{15}$$

We call the prior described by (13-15) as *generalized hierarchical beta process* (G-HBP) prior.

$$B \sim \mathrm{BP}(\gamma_0, B_0)$$

$$B = \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k} \text{ where } \beta \sim \mathrm{stickIBP}(\tau_0), \ \phi_k \overset{\text{iid}}{\sim} B_0$$

$$G \sim \mathrm{DP}(\xi_0, \mathrm{BP}(\alpha, B)), \mu \sim \mathrm{GEM}(\xi_0), A_j \overset{\text{iid}}{\sim} \mathrm{BP}(\alpha, B)$$

$$G = \sum_{j=1}^{\infty} \mu_j \delta_{A_j} \text{ where } A_j = \sum_{k=1}^{\infty} \pi_{j,k} \delta_{\phi_k},$$

$$\text{and } \pi_{j,k} \sim \text{beta}\left(\alpha_j \beta_k, \alpha_j \bar{\beta}_k\right), \ j = 1, 2, \ldots \tag{9}$$

For $t = 1$ to $T$

$$u_t \sim \mathrm{Discrete}(\mu), \ \boldsymbol{z}_t \sim \mathrm{Ber}(\pi_{u_t}) \tag{10}$$

$$\theta_t \sim \mathcal{N}\Big(\Phi(\mathbf{z}_t \odot \mathbf{w}_t)^\mathsf{T}, \sigma_{n_{u_t}}^2 \mathbf{I}\Big), \mathbf{w}_t \sim \mathcal{N}\left(\mathbf{0}, \sigma_{w_{u_t}}^2 \mathbf{I}\right) \tag{11}$$

$$\begin{cases} y_{ti} \sim \mathcal{N}\left(\theta_t^\mathsf{T} \mathbf{x}_{ti}, \sigma_y^2\right), \ i = 1 \text{ to } N_t & \text{regression} \\ y_{ti} \sim \mathrm{Ber}\left(\sigma\left(\theta_t^\mathsf{T} \mathbf{x}_{ti}\right)\right), \ i = 1 \text{ to } N_t & \text{classification} \end{cases} \tag{12}$$

Figure 2: A summary of the G-HBP prior, and the F-MTL generative process. In above, $\sigma(.)$ denotes the sigmoid function and $\bar{\beta}_k \triangleq 1 - \beta_k$. The notations $\mathcal{N}(.,.)$, Ber(.) and beta(.,.) denote the normal, the Bernoulli and the beta distributions respectively; $\Phi = [\phi_1, \phi_2, \ldots]$ and $z_t$ represents $z_{g_t, u_t}$ described earlier.

## 4. Model Inference

The closed form inference for the proposed model is intractable. Therefore, we use Markov chain Monte Carlo (MCMC) (Gilks et al., 1995), which is widely used for performing inference with such kind of hierarchical Bayesian models. The MCMC state space comprises of the variables $\{\Theta, \Phi, \mathbf{W}, \mathbf{Z}, \pi, \beta, \mathbf{u}, \mu, \alpha\}$ where $\mathbf{u} \triangleq \{u_1, \ldots, u_T\}$. However, some of these variables $\{\pi, \beta, \mu\}$ are nuisance variables. Due to using conjugate priors, we can integrate out $\{\pi, \mu\}$ and sample the remaining variables. For inference, we combine Gibbs sampling with adaptive rejection sampling/Laplace approximation (Bishop et al., 2006).

To sample $\beta$, we use stick-breaking process construction of the Indian buffet process (Teh et al., 2007). This is made possible by fixing the concentration parameter of the parent beta process to one. However, the stick-breaking process construction requires us maintaining an infinite set of atoms. This is dealt with by using the slice sampler (Gupta et al., 2012b), which turns the infinite representation into a finite one. The slice sampler employs an auxiliary variable $\rho$ that can be sampled from a uniform distribution. In particular, $\rho \mid \mathbf{Z}, \beta \sim U(0, \beta^*)$ where $\beta^* = \min_{k \mid \exists i,\ \mathbf{Z}^{ik}=1} \beta_{(k)}$ and $\beta_{(k)}$ is a decreasing order representation of $\beta_k$. Given $\rho$, if $\beta_{(k)} < \rho$, $\mathbf{Z}^{t,k} = 0$ for all $t$ and therefore, we need to update $\mathbf{Z}^{t,k}$ for only those $k$ such that $\beta_{(k)} \geq \rho$. Assuming $K^\dagger$ to be an index so that all active features have index $k < K^\dagger$, if $\beta_{(K^\dagger)} \geq \rho$, we extend our stick-breaking representation until $\beta_{(K^\dagger)} < \rho$ (see (Gupta et al., 2012b) for details).

### 4.1. Sampling $\Theta$

Sampling of $\theta_t$ can be done independently for each $t$. Under the Gibbs sampling framework, we can sample $\theta_t$ from the following conditional posterior distribution

$$p(\theta_t \mid \ldots) \propto p(\mathbf{y}_t \mid \mathbf{X}_t, \theta_t)\, p(\theta_t \mid \Phi, \mathbf{z}_t, \mathbf{w}_t) \qquad (16)$$

Since our data generative process differs for regression and classification, we separately describe them below.

#### 4.1.1. Regression

Using the prior distributions of (11-12) for regression model, the posterior $p(\theta_t \mid \ldots)$ can be derived to be a multi-variate Gaussian with mean and covariance as

$$\Sigma_{\theta_t}^{\text{post}} = \left(\frac{\mathbf{X}_t\mathbf{X}_t^{\mathsf{T}}}{\sigma_y^2} + \frac{\mathbf{I}}{\sigma_t^2}\right)^{-1}, m_{\theta_t}^{\text{post}} = \Sigma_{\theta_t}^{\text{post}} \left(\frac{\mathbf{X}_t\mathbf{y}_t}{\sigma_y^2} + \frac{\Phi\mathbf{h}_t}{\sigma_t^2}\right) \qquad (17)$$

where we have $\mathbf{h}_t \triangleq \mathbf{z}_t \odot \mathbf{w}_t$.

#### 4.1.2. Classification

Using the generative distributions of (11-12), the conditional posterior $p(\theta_t \mid \ldots)$ of (16) can be written as

$$p(\theta_t \mid \ldots) \propto \left[\Pi_{i=1}^{N_t} s_{ti}^{y_{ti}} (1 - s_{ti})^{1-y_{ti}}\right] e^{-R_t/2\sigma_t^2} \qquad (18)$$

where we define $s_{ti} \triangleq \sigma\left(\theta_t^{\mathsf{T}}\mathbf{x}_{ti}\right)$ and $R_t \triangleq ||\theta_t - \Phi\mathbf{h}_t||_2^2$. The above expression can not be simplified to standard parametric distributions. Therefore, exact inference is intractable. From this point, we found two ways to proceed (1) adaptive rejection sampling (ARS) (2) Laplace approximation. The *first* method fits well under the Gibbs framework while ensuring sampling from the exact distribution (Gilks & Wild, 1992). To motivate the application of adaptive rejection sampling, we note that the second derivative of *log* of the above posterior is given as

$$\bigtriangledown_{\theta_t}^2 \ln p(\theta_t \mid \ldots) = -\mathbf{X}_t\mathbf{D}_s(\theta_t)\mathbf{X}_t^{\mathsf{T}} - \frac{\mathbf{I}}{\sigma_t^2} \qquad (19)$$

where $\mathbf{D}_s(\theta_t) \triangleq \operatorname{diag}\left([s_{t1}(1 - s_{t1}), \ldots, s_{tN_t}(1 - s_{tN_t})]\right)$ is a diagonal matrix with entries between 0 and 1. We note that the Hessian given by (19) is a negative semi-definite implying that $p(\theta_t \mid \ldots)$ is *log-concave.* This allows efficient sampling of $\theta_t$ through ARS.

The *second* method to obtain the samples of $\theta_t$ is to use Laplace approximation, which is obtained by finding the mode of the posterior distribution and then fitting a Gaussian having mean at the computed mode. We can find the mode of the posterior by maximizing the *log* of the posterior, which can be done by finding a solution $\theta_t^{\text{Laplace}}$ to $\bigtriangledown_{\theta_t} \ln p(\theta_t \mid \ldots) = 0$. This can be done using either gradient-descent or Newton's method. The *co-variance* of the Gaussian is given by the negative of the inverse of Hessian of log posterior, i.e. $\Sigma_{\theta_t}^{\text{Laplace}}$. Given $\theta_t^{\text{Laplace}}$ and $\Sigma_{\theta_t}^{\text{Laplace}}$, the posterior in (18) takes the form

$$q(\theta_t) = \mathcal{N}\left(\theta_t^{\text{Laplace}}, \Sigma_{\theta_t}^{\text{Laplace}}\right) \qquad (20)$$

In our implementation, we found that both gradient-descent and Newton's method methods worked fairly well and the later was clearly faster.

### 4.2. Sampling $u_t$

Gibbs conditional posterior of $u_t$ can be written as

$$\begin{aligned} &p(u_t \mid \ldots) \propto p\left(u_t \mid \mathbf{u}^{-t}, \mathbf{z}_{g_t,u_t}, \mathbf{Z}_{u_t}^{-g_t}, \beta, \mathbf{w}_{g_t,u_t}\right) \\ &\propto \underbrace{p\left(u_t \mid \mathbf{u}^{-t}\right)}_{\text{CRP}} \underbrace{p\left(\mathbf{z}_{g_t,u_t} \mid \mathbf{Z}_{u_t}^{-g_t}, \beta\right) p\left(\mathbf{w}_{g_t,u_t} \mid \sigma_{w_{u_t}}\right)}_{\text{joint likelihood in latent space}} \end{aligned} \qquad (21)$$

where we define $\mathbf{u}^{-t} = \{u_{t'} \mid t' \neq t\}$ and $\mathbf{Z}_{u_t}^{-g_t} = \{\mathbf{z}_{g',u_t} \mid g' \neq g_t\}$. In the above expression, the first term is the predictive prior distribution of $u_t$ and widely known as Chinese Restaurant process (CRP). The second term is the likelihood, which measures how well $t$-th task predictor goes along with the group indexed by $u_t$ in latent space. The predictive distribution of $\mathbf{z}_{g_t,u_t}$ given $\mathbf{Z}_{u_t}^{-g_t}$ can be simplified to

$$\begin{aligned} &p\left(\mathbf{z}_{g_t,u_t} \mid \mathbf{Z}_{u_t}^{-g_t}, \beta\right) \\ &= \frac{1}{\beta^*} \prod_{k=1}^{K^\dagger} \frac{\Gamma(\alpha_{u_t})\,\Gamma\left(\alpha_{u_t}\beta_{(k)} + f_{u_t}^{-t,k}\right)\Gamma\left(\alpha_{u_t}\bar{\beta}_{(k)} + \bar{f}_{u_t}^{-t,k}\right)}{\Gamma\left(\alpha_{u_t+T_{u_t}}\right)\Gamma\left(\alpha_{u_t}\beta_{(k)}\right)\Gamma\left(\alpha_{u_t}\bar{\beta}_{(k)}\right)} \end{aligned} \qquad (22)$$

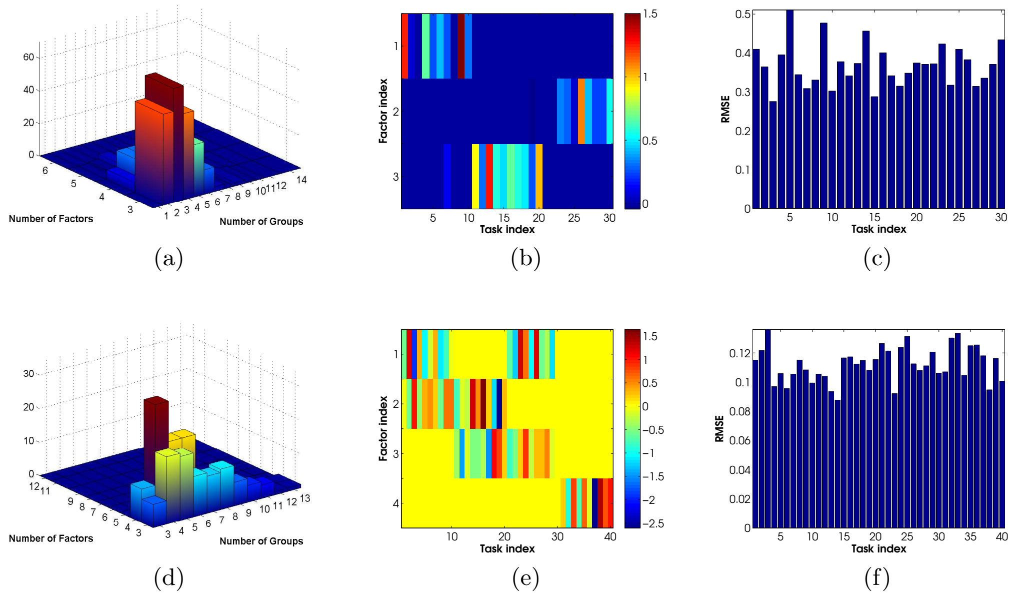


Figure 3: Synthetic data results (a) The joint posterior over the number of groups ($J$) and the number of subspace bases ($K$) for the first dataset (b) Inferred basis usages for the first dataset (c) The RMSE for various tasks in the first dataset (d) The joint posterior over the number of groups ($J$) and the number of subspace bases ($K$) for the second dataset (e) Inferred basis usages for the second dataset. (f) The RMSE for various tasks in the second dataset.

where $f_{u_t}^{-t,k} \triangleq \sum_{t' \neq t} \mathbf{z}_{u_t}^{t',k}$, $\bar{f}_{u_t}^{-t,k} = T_{u_t} - f_{u_t}^{-t,k}$ and $T_{u_t}$ is the total number of tasks in the group indexed by $u_t$. Plugging (22), the CRP prior over $u_t$ and the normal prior over $\mathbf{w}_{g_t,u_t}$ into (21), we can sample $u_t$ from a *categorical* distribution.

### 4.3. Sampling Z

Gibbs conditional posterior for $\mathbf{Z}$ can be written as

$$p\left(z_{tk} = 1 \mid ...\right) \propto \underbrace{p\left(z_{tk} = 1 | \mathbf{Z}_{u_t}^{-g_t,k}, \beta_k\right)}_{\text{Predictive HBP prior}} \underbrace{p\left(\theta_t | \Phi, z_{tk} = 1, \mathbf{z}_t^{-k}, \mathbf{w}_t\right)}_{\text{task parameter likelihood}} \quad (23)$$

where $\mathbf{z}_{u_t}^{-g_t,k} \triangleq \{z_{t'k} \mid u_{t'} = u_t, t' \neq g_t\}$ and $\mathbf{z}_t^{-k} \triangleq \{z_{tk'} \mid k' \neq k\}$. The predictive prior simplifies to

$$p\left(z_{tk} = 1 \mid \mathbf{z}_{u_t}^{-g_t,k}, \beta_k\right) = \frac{\alpha_{u_t}\beta_{(k)} + f_{u_t}^{-t,k}}{\beta^* \left(\alpha_{u_t} + T_{u_t}\right)} \quad (24)$$

Thus the variable $z_{tk}$ can be drawn from the posterior distribution of (23), which is a *Bernoulli* distribution.

### 4.4. Sampling W, $\Phi$ and $\beta$

Once we sample the group indicator of the $t$-th task (i.e. the variable $u_t$), we have a partition of the task predictors $\Theta$ into $J$ groups. In other words, we have a partition of the tasks as $\Theta = [\Theta_1, \ldots, \Theta_J]$ where $\Theta_j$ is a matrix formed by stacking (column-wise) each task predictor $t$ such that $u_t = j$. Using $u_t$'s, we can partition the matrix $\mathbf{W}$ similarly. Given this partition, we can utilize the sampling steps in (Gupta et al., 2012b) to sample $\Phi$, $\mathbf{W}$, $\beta$ and the hyperparameters.

## 5. Experiments

We evaluate our proposed F-MTL on multi-task regression and classification applications using both synthetic and real datasets. Experiments with synthetic data do a sanity check of the model while also illustrating the model behavior. Experiments with real datasets demonstrate the true effectiveness of our model for multi-task learning.

### 5.1. Synthetic Data Experiments

For synthetic data experiments, we use *two* different datasets. The *first* dataset is identical to the synthetic data used in (Kang et al., 2011; Kumar & Daumé III, 2012). This dataset has 3 groups of tasks. Within each task-group, there are 10 tasks whose predictors (i.e. $\theta_t$) are identical up to a scaling factor. Each task has 15 examples lying in a 20-dim space. The task predictors are used in a linear regression model to generate target values of the training data. We note that only tasks within a group are related.

The *second* dataset is generated to simulate the varying degree of relatedness among task predictors. The degree of relatedness is controlled by varying the number of subspace bases shared by the tasks. For this dataset, we have 4 groups of tasks, 10 tasks per group and 4 bases. The task predictors of the first group are generated as random (Gaussian with zero mean

Table 1: Regression/classification results on real datasets: the performance is reported in terms of RMSE for regression and classification error (in %) for classification tasks. The results are averaged over 20 trials.

| | | *Regression* | | *Classification* | |
|---|---|---|---|---|---|
| Method | | Computer survey | School | MNIST | USPS |
| STL | | $2.70 \pm 0.10$ | $10.67 \pm 0.20$ | $14.8 \pm 0.34$ | $9.0 \pm 0.4$ |
| NG-MTL | | $2.06 \pm 0.07$ | $10.18 \pm 0.15$ | $14.4 \pm 0.28$ | $7.8 \pm 0.2$ |
| DG-MTL | | $2.01 \pm 0.10$ | $10.18 \pm 0.20$ | $14.0 \pm 0.30$ | $7.8 \pm 0.2$ |
| GO-MTL | | $1.76 \pm 0.09$ | $10.04 \pm 0.24$ | $13.4 \pm 0.30$ | $7.2 \pm 0.2$ |
| **F-MTL** | | $\mathbf{1.65 \pm 0.02}$ | $\mathbf{9.73 \pm 0.03}$ | $\mathbf{6.99 \pm 0.34}$ | $\mathbf{3.58 \pm 0.07}$ |

Table 2: Comparison with a recent Bayesian nonparametric multi-task learning model (referred to as MFA-MTL) (Passos et al., 2012). The number of examples in the training set for these comparison are quite small (*refer* text for details).

| | | *Regression* | | *Classification* | |
|---|---|---|---|---|---|
| | | Computer survey | School | Landmine | 20-Newsgroup |
| MFA-MTL | | **5.46** | 19.35 | 37.6 | 23.1 |
| **F-MTL** | | $8.33 \pm 0.75$ | $\mathbf{12.62 \pm 0.29}$ | $\mathbf{7.37 \pm 1.09}$ | $\mathbf{17.39 \pm 1.15}$ |

and one standard deviation) linear combination of the basis-1 and basis-2. The task predictors of the second group are generated similarly using basis-2 and basis-3 causing basis-2 to be shared between groups 1 and 2. Tasks of the third group are generated using basis-1 and basis-3 to simulate relatedness with the tasks in the first group. Finally, a fourth group is created with task predictors using only basis-4 and thus not share anything from the tasks of other groups. For each task, we randomly generate 15 examples in a 20-dim space. These task predictors are used in a linear regression model to generate the target values where we introduce an additive Gaussian noise with zero mean and standard deviation 0.1.

For our model, we initialized all the hyperparameters ( i.e. $\sigma_y, \sigma_n, \sigma_w, \tau, \gamma, \alpha_j$) to 1 while both the number of bases ($K$) and the number of groups ($J$) to 10. We run the Gibbs sampler for 500 iterations and the results (after rejecting first 200 samples as burn-in) are shown in Figure 3. The first row of figures depict the results for the *first* synthetic dataset. The joint posterior over $(K, J)$ in Figure 3(a) clearly shows the mode of the distribution at $(3, 3)$–accurately recovering the subspace dimension and the number of task groups. Figure 3(b) depicts the basis usages of different tasks clearly showing that the three groups are nearly disjoint in using the bases. The 3(c) shows the root-mean-square-error (RMSE) for all the tasks, which is roughly at the true noise level. The second row of figures depict the similar results for the *second* synthetic dataset. We can see that model infers the true number of bases and groups (see the mode at $(4, 4)$ in Figure 3(d)) along with the correct basis usages including the sharing/non-sharing patterns. The model captures the fact that the first three task groups are related with one another while the fourth is disjoint. This ensures that there is no inductive bias transferred from the first three groups to the fourth group and *vice versa.*

### 5.2. Real Data Experiments

We perform experiments on *four* real datasets: two of them have regression tasks and the other two have classification tasks.

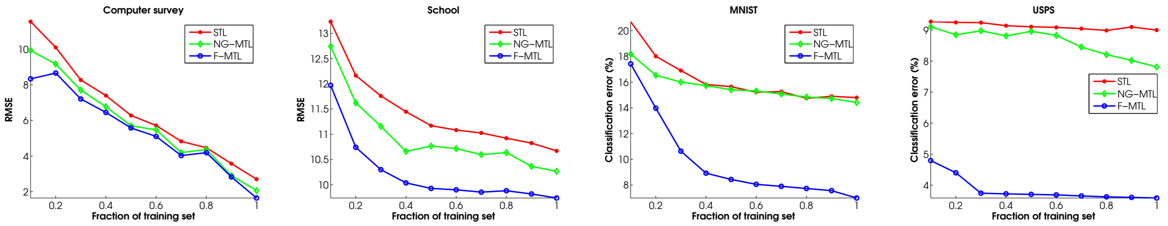
#### 5.2.1. Datasets

- **Computer Survey**: This dataset contains a survey where 190 students rated 20 personal computers (PC). Students rated each PC over 13 features (see (Argyriou et al., 2008)) and assigned a score (scale: 0-10) indicating their likelihood of buying. We treat students as tasks while PCs as examples.

- **School**: This dataset consists of examination scores of 15,362 students from 139 schools. For each student, there are 26 features (see (Argyriou et al., 2008)) covering year of examination, school and student-specific attributes. Each school is treated as a task and its students as examples.

- **MNIST and USPS**: Both of these datasets consist of handwritten digits having 10 classes and a total of 2000 examples where 1000, 500 and 500 samples are used for training, test and validation respectively. The problem is posed as MTL where each task is classification of one class *vs.* others.

To have a fair comparison, we keep the training and test sets for all the above datasets *identical* to the ones used in (Kumar & Daumé III, 2012). Further, all our comparisons are based on the RMSE for regression and



(a) (b) (c) (d)

Figure 4: Multi-task regression/classification performances for various datasets w.r.t. varying amount of training data.

*multi-class classification error* for classification.

### 5.2.2. Baseline Methods

- **STL**: This baseline learns each task separately and does not exploit any relatedness of the tasks.
- **NG-MTL** (Argyriou et al., 2008): This model represents the task predictors in a subspace without any grouping.
- **DG-MTL** (Kang et al., 2011): This model partitions the tasks into groups and tasks in a group are represented in a subspace. There is *no* sharing of subspace bases across groups.
- **GO-MTL** (Kumar & Daumé III, 2012): This model groups the task predictors in a subspace and allows the bases to be shared across groups.
- **MFA-MTL** (Passos et al., 2012): This is a *Bayesian nonparametric* model dividing tasks into groups, each group modeled using a subspace and *no* sharing of the bases across groups.

### 5.2.3. Experimental Results

Table 1 presents a comparison of our proposed F-MTL with the first *four* baselines for both regression and classification tasks. The F-MTL clearly outperforms all the baselines for both tasks irrespective of the datasets. The closest contender is the GO-MTL, whose performance on regression tasks is somewhat close, however, the difference is still statistically significant (details omitted). On classification tasks, F-MTL gets significantly better results (6.4% improvement on MNIST and 3.6% improvement on USPS) than GO-MTL. This is important especially under the view that both methods use logistic regression for mapping inputs to target outputs. We attribute this improvement in performance to the following (1) F-MTL being a Bayesian nonparametric model automatically infers the number of groups and subspace bases from data and thus uses optimal sharing (2) F-MTL shares the latent bases across different task groups (enables overlapping/grouping) while allowing a separate distribution for each group respecting the idiosyncrasies.

We also compare our method with MFA-MTL (Passos et al., 2012). This comparison is presented separately as this baseline uses different datasets (Landmine and 20 Newsgroup) for classification and the computer and school datasets for regression *but* with different training/test settings. We use the same training and test sets as in (Passos et al., 2012). Table 2 shows the comparison between the two models where we can see that F-MTL clearly outperforms MFA-MTL on school dataset for regression and both Landmine and 20-Newsgroup datasets for classification.

Finally we show the performance of our method for varying fractions of training data and compare it with those of STL and NG-MTL. For this, we used the same training and test sets as used for generating Table 1. For all the datasets, we increased the training examples from 10% to 100% with a step of 10% and measured the performances. The results are shown in Figure 4. We can see from the figure that the performance of the proposed method is consistently better irrespective of the size of the training set.

## 6. Conclusion

We present a *novel* Bayesian nonparametric, factorial multi-task learning (F-MTL) framework that groups the similar task predictors by representing them in a low dimensional subspace. A key feature of the proposed F-MTL is that it automatically infers the number of task groups and allows the groups to share the subspace bases. This feature enables the framework to jointly model the tasks with varying degree of relatedness. For this, we propose a generalized hierarchical beta process (G-HBP) prior that permits a hierarchy of potentially infinite number of child beta processes controlled via a Dirichlet process. Another key feature of the proposed model is that it uses a different distribution of basis usages for each group allowing each group to vary when necessary. This has implications in overcoming the negative inductive biases from unrelated tasks. Using synthetic and real datasets, we demonstrate the utility of the model for regression and classification tasks while outperforming many recent state-of-the-art multi-task learning techniques.

## References

Agarwal, A., Daumé III, H., and Gerber, S. Learning multiple tasks using manifold regularization. *Advances in neural information processing systems*, 23:46–54, 2010.

Ando, R.K. and Zhang, T. A framework for learning predictive structures from multiple tasks and unlabeled data. *The Journal of Machine Learning Research*, 6: 1817–1853, 2005.

Argyriou, A., Evgeniou, T., and Pontil, M. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

Baxter, J. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

Bi, J., Xiong, T., Yu, S., Dundar, M., and Rao, R. An improved multi-task learning approach with applications in medical diagnosis. *Machine Learning and Knowledge Discovery in Databases*, pp. 117–132, 2008.

Bishop, Christopher M et al. *Pattern recognition and machine learning*, volume 4. springer New York, 2006.

Caruana, R. Multitask learning. *Machine Learning*, 28(1): 41–75, 1997.

Corduneanu, A. and Bishop, C.M. Variational bayesian model selection for mixture distributions. In *Artificial intelligence and Statistics*, volume 2001, pp. 27–34. Morgan Kaufmann Waltham, MA, 2001.

Daumé III, H. Bayesian multitask learning with latent hierarchies. In *Proc. of the 25th Conference on Uncertainty in Artificial Intelligence*, pp. 135–142, 2009.

Ferguson, T.S. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.

Gilks, Walter R and Wild, Pascal. Adaptive rejection sampling for gibbs sampling. *Applied Statistics*, pp. 337–348, 1992.

Gilks, W.R., Richardson, S., and Spiegelhalter, D. *Markov Chain Monte Carlo in practice: interdisciplinary statistics*, volume 2. Chapman & Hall/CRC, 1995.

Gupta, S.K., Phung, D., Adams, B., and Venkatesh, S. A Bayesian framework for learning shared and individual subspaces from multiple data sources. In *Advances in Knowledge Discovery and Data Mining, 15th Pacific-Asia Conference*, pp. 136–147, 2011.

Gupta, S.K., Phung, D., and Venkatesh, S. A Bayesian nonparametric joint factor model for learning shared and individual subspaces from multiple data sources. In *Proceedings of 12th SIAM International Conference on Data Mining*, pp. 200–211, 2012a.

Gupta, S.K., Phung, D., and Venkatesh, S. A slice sampler for restricted hierarchical beta process with applications to shared subspace learning. In *Proc. of 28th Uncertainty in Artificial Intelligence (UAI)*, pp. 316–325, 2012b.

Gupta, S.K., Phung, D., Adams, B., and Venkatesh, S. Regularized nonnegative shared subspace learning. *Data Mining and Knowledge Discovery*, 26(1):57–97, 2013.

Kang, Z., Grauman, K., and Sha, F. Learning with whom to share in multi-task feature learning. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 521–528, 2011.

Kingman, J.F.C. Completely random measures. *Pacific Journal of Mathematics*, 21(1):59–78, 1967.

Kumar, A. and Daumé III, H. Learning task grouping and overlap in multi-task learning. In *International Conference on Machine Learning (ICML)*, 2012.

Passos, A., Rai, P., Wainer, J., and Daumé III, H. Flexible modeling of latent task structures in multitask learning. In *International Conference on Machine Learning (ICML)*, 2012.

Rai, P. and Daumé III, H. Infinite predictor subspace models for multitask learning. *Journal of Machine Learning Research - Proceedings Track*, 9:613–620, 2010.

Rosenstein, M.T., Marx, Z., Kaelbling, L.P., and Dietterich, T.G. To transfer or not to transfer. In *NIPS Workshop on Inductive Transfer*, volume 10, 2005.

Sethuraman, J. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4(2):639–650, 1994.

Teh, Y.W., Jordan, M.I., Beal, M.J., and Blei, D.M. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

Teh, Y.W., Görür, D., and Ghahramani, Z. Stick-breaking construction for the Indian buffet process. *Journal of Machine Learning Research - Proceedings Track*, 2:556–563, 2007.

Thibaux, R. and Jordan, M.I. Hierarchical beta processes and the Indian buffet process. *Journal of Machine Learning Research - Proceedings Track*, 2:564–571, 2007.

Wang, X., Zhang, C., and Zhang, Z. Boosted multi-task learning for face verification with applications to web image and video search. In *CVPR 2009. IEEE Conference on*, pp. 142–149. IEEE, 2009.

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