LaTeX2e guide for authors using the EngC design

Subtitle, if you have one

ALI WOOLLATT

This guide was compiled using EngC.cls 2011/02/03, v1.10

The latest version can be downloaded from:
https://authornet.cambridge.org/information/productionguide/
LaTeX_files/EngC.zip

Contents

Illustrations

Tables

Boxes

List of contributors

Zhe Gan
Electrical and Computer Engineering, Duke University, NC, USA

Xin Yuan
Electrical and Computer Engineering, Duke University, NC, USA

Ricardo Henao
Electrical and Computer Engineering, Duke University, NC, USA

Ephraim L. Tsalik
Emergency Medicine Service, Durham Veterans Affairs Medical Center, NC, USA
Department of Medicine, Duke University Medical Center, NC, USA

Lawrence Carin
Electrical and Computer Engineering, Duke University, NC, USA

Part I

Getting started

1 Inference of gene networks associated with the host response to infectious disease

Zhe Gan[†], Xin Yuan[†], Ricardo Henao[†], Ephraim L. Tsalik[‡♭] and Lawrence Carin[†]

[†]Electrical and Computer Engineering Department, Duke University

[‡]Emergency Medicine Service, Durham Veterans Affairs Medical Center

[♭]Department of Medicine, Duke University Medical Center

1.1 Background

From a statistical-modeling perspective, gene expression analysis can be roughly divided into two phases: exploration and prediction. In the former, the practitioner attempts to get a general understanding of a dataset by modeling its variability in an interpretable way, such that the inferred model can serve as a feature extractor and hypotheses-generating mechanism of the underlying biological processes. Factor models are among the most widely employed techniques for exploratory gene expression analysis [1,2], with principal component analysis a popular special case [3]. Predictive modeling, on the other hand, is concerned with finding a relationship between gene expression and phenotypes, that can be generalized to unseen samples. Examples of predictive models include classification methods like logistic regression and support vector machines [4,5].

Factor models infer a latent covariance structure among the genes or biomarkers, with data modeled as generated from a noisy low-rank matrix factorization, manifested in terms of a *loadings matrix* and a *factor scores* matrix. Different specifications for these matrices give rise to special cases of factor models, such as principal components analysis [6], non-negative matrix factorization [7], independent component analysis [8], and sparse factor models [1]. Factor models employing a sparse factor loadings matrix are of significant interest in gene-expression analysis, as the non-zero elements in the loadings matrix may be interpreted as correlated gene networks [1,2,9].

Discriminative models, in particular binary linear classification models, aim to find a linear combination of input features or covariates to separate observed data into two groups or phenotypes (using kernel techniques, such approaches are readily extended to nonlinear classifiers [10]). One proceeds by first learning the parameters of the classifier using labeled data, in which one knows the phenotype of every data point (in a semi-supervised learning procedure, some data are labeled and others non-labeled [11]). Once so learned, the model parameters are fixed and used to predict the phenotype of unlabeled data; this is often termed the inference step [12]. In the work reported here we jointly learn the factor model (feature learning) and classifier, and we develop a framework that scales well to high-dimensional data.

1.2 Factor models in gene expression analysis

Gene expression analysis typically involves considering a relatively small number of observations, each of which is composed of expression values from tens of thousands of genes. In this setting, called the “large p, small N” problem [13], the number of biomarkers p is much larger than the number of observations N ($N \ll p$). In this regime direct analysis either using factor models or discriminative models is infeasible, because the problem is ill-posed [14]. For factor models, in order to yield reliable modeling, two key assumptions are widely imposed: (i) the number of factors needed to explain the data is small (low-rank assumption of the dataset), and (ii) each factor is responsible for explaining only a small subset of variables. The latter also applies to discriminative models, in the sense that only a small subset of variables are necessary for classification. From an application point of view, this *sparsity* assumption yields results that can be interpreted, *e.g.* the small subset of correlated genes associated with a factor correspond ideally to biologically meaningful *pathways*, *modules* or *gene networks*. Factor models can be used as a general feature extraction tool in the context of gene expression analysis [1].

Under the Bayesian paradigm, the sparsity assumption is specified via prior distributions, such as spike-and-slab [1,15,16] or shrinkage priors [10,17–20]. The key difference between these two prior distribution families is that the former assumes signal (slab) and no-signal (spike) as coming from a bi-modal distribution, whereas in the latter both signal and no-signal are modeled as a unimodal heavy-tailed distribution, in which values close to zero are designated as no-signal (*i.e.*, as noise). Common choices for continuous shrinkage priors include Student’s-t [10], double exponential (Laplace) [17], the horseshoe [18] and the three parameter beta normal (TPBN) [20]. The TPBN is an example of global-local shrinkage priors as defined in [19], that has demonstrated superior performance in terms of mixing when compared to other shrinkage specifications and spike-and-slab priors.

Factor models and discriminative models have been successfully used to identify host responses in infectious disease studies [21–26]. Such analyses are usually performed by first using a factor model to obtain a low-dimensional representation of the data, encoded by the factor scores. A discriminative model is next used to characterize the phenotype of interest from the factor scores. This two-step procedure can be seen as the exploration and prediction phases described above. Provided that both models are equipped with sparsity priors, we can use the discriminative model to identify a subset of factor scores responsible for the classification rule. Since these should correlate with the phenotype, they are thus proxies for the corresponding host response. Subsequently, we can use the factor loadings to identify the correlated subset of genes responsible for the relevant factor scores. As a result, we can build a classification model and learn about the gene network that contributes to the predictor's outcome, in a systematic manner.

A drawback of the two-step procedure described above is that feature extraction (factor modeling) and classification are performed separately, thus the factor model is not informed of the ultimate use of the factor scores in a classifier (it simply tries to fit the data in a generative sense, and may ignore subtle features of the data that are critical for the subsequent classification task). In a discriminative factor model [27], also known as supervised dictionary learning, the two-steps are performed jointly.

In a Bayesian setting, classification models have always been troubled with complications in terms of learning and inference, due to the lack of conditional conjugacy between the likelihood function implied by the decision function and the prior distributions for the parameters of the model. Probit regression is an interesting case, because it is perhaps the first classification model provided with efficient inference, as a consequence of variable augmentation [28]. Although augmentation schemes for logistic regression [29] and support vector machines [30] have been proposed recently, they have not yet been combined with factor models. Since discriminative factor models using probit regression as the classifier have been investigated [31, 32] previously, we focus this work on effective inference for discriminative factor models using logistic regression and support vector machine classifiers.

The remainder of this chapter is organized as follows. Sections 1.3 and 1.4 describe factor models and discriminative models, respectively. Section 1.5 introduces our discriminative factor model, while Section 1.6 describes the learning and inference procedures employed. Section 1.7 presents extensive results on real gene-expression data, and finally Section 1.8 provides summary comments and observations about the proposed framework.

1.3 Factor models

Assume a matrix of observed data $\mathbf{X} \in \mathbb{R}^{p\times N}$, where each column corresponds to one of $n \in \{1, \ldots, N\}$ samples, $\mathbf{x}_n \in \mathbb{R}^p$, each of which contains expression values for p genes (while we focus the discussion on gene-expression analysis, the basic modeling structure developed here is applicable to other biomarkers, such as proteomics, metabolomics, etc.). We consider a linear factor model of the form

$$\begin{aligned} \mathbf{x}_n &= \mathbf{A}\boldsymbol{\Lambda}\mathbf{s}_n + \boldsymbol{\epsilon}_n \,, \\ \mathbf{s}_n &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \,, \\ \boldsymbol{\epsilon}_n &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}^{-1}) \,, \end{aligned} \tag{1.1}$$

where a_{jk} is an element of loadings matrix $\mathbf{A} \in \mathbb{R}^{p \times K}$, K is the total number of factors, $\boldsymbol{\epsilon}_n$ is additive noise (or model residual) for the n-th sample, ψ_i is an element of the noise precision matrix $\boldsymbol{\Psi} = \mathrm{diag}(\psi_1, \ldots, \psi_p)$, λ_k is one of the factor specific scaling coefficients in matrix $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_K)$, and $\mathbf{s}_n \in \mathbb{R}^K$ is a vector of Gaussian distributed factor scores for the n-th observation. While we truncate the model to a (large) upper bound of K of factors, the shrinkage/sparsity imposed on the factor loadings allows one to infer the subset of factors (typically less than K) actually needed to represent the data.

Provided that factor scores $\mathbf{s}_n$ and noise terms $\boldsymbol{\epsilon}_n$ are independent and Gaussian distributed, from (1.1) it follows that $\mathbf{x}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{A}\boldsymbol{\Lambda}\boldsymbol{\Lambda}\mathbf{A}^\top + \boldsymbol{\Psi}^{-1})$, *i.e.*, the factor model estimates the covariance structure of data $\mathbf{x}_n$ as a low rank representation if $K < p$, once the factor scores $\mathbf{s}_n$ are analytically marginalized out.

The generative model encoded by (1.1) implies that the expression value for a particular gene is a linear combination of columns of $\mathbf{A}$, with the columns weighted by the factor scores $\mathbf{s}_n$. We can thus see the columns of $\mathbf{A}$ as constituting a dictionary with K gene-related dictionary elements (columns); since $\mathbf{A}$ is shared among all data samples $\{\mathbf{x}_n\}$, the factor scores $\mathbf{s}_n$ serve as a low-dimensional representation of each $\mathbf{x}_n$. Prior distributions for $\mathbf{A}$ and $\boldsymbol{\Lambda}$ are specified to encourage (near) sparsity, to ease interpretability of the features encoded by columns of $\mathbf{A}$, while also being able to automatically estimate the usually unknown number of factors needed by the model to effectively explain the data (to infer the subset of K columns actually needed to represent the observed data).

1.3.1 Shrinkage prior

To impose (near) sparseness on the loadings matrix $\mathbf{A}$, we employ the three parameter beta normal (TPBN) prior, a general prior distribution that can be expressed as scale mixtures of normals [20]. Specifically, if $a_{jk} \sim \mathrm{TBPN}(a, b, \phi)$, where $j = 1, \ldots, p, k = 1, \ldots, K$, we can write

$$\begin{aligned} a_{jk} &\sim \mathcal{N}(0, \zeta_{jk})\,, \\ \zeta_{jk} &\sim \mathrm{Gamma}(a, \xi_{jk})\,, \\ \xi_{jk} &\sim \mathrm{Gamma}(b, \phi_k)\,. \end{aligned} \tag{1.2}$$

When $a = b = 1/2$, the TPBN prior reduces to the horseshoe prior [18]. For fixed values of a and b, ϕ_k controls the shrinkage level of the k-th column of $\mathbf{A}$, so that smaller values of ϕ_k yield stronger shrinkage. Furthermore, we can set a prior on ϕ_k to allow the model to learn individual shrinkage levels for each column of $\mathbf{A}$. For instance,

$$\begin{aligned} \phi_k &\sim \mathrm{Gamma}\left(\tfrac{1}{2}, w\right)\,, \\ w &\sim \mathrm{Gamma}\left(\tfrac{1}{2}, 1\right)\,, \end{aligned}$$

where w is a latent variable whose distribution serves as support for the shrinkage levels of $\mathbf{A}$. Alternatively, when prior knowledge about the expected sparsity level of $\mathbf{A}$ is available, ϕ_k can be set accordingly, rather than inferred from data.

The prior in (1.2) also encompasses as special cases well known shrinkage priors, such as double-exponential [17, 33], Strawderman-Berger [34], Normal-Exponential-Gamma [35] and horseshoe [18] priors. Further, it can be seen as an example of a yet larger family of continuous shrinkage hierarchies, known as global-local shrinkage priors [19, 36].

One of the most appealing features of (1.2) is its excellent mixing properties, which stem from the fact of it being marginally a continuous unimodal distribution with closed-form conditional posteriors. Bimodal sparsity, including distributions such as spike-and-slab [1, 15] and the Indian buffet process [37], often face mixing difficulties, whereas commonly used unimodal shrinkage priors (such as double-exponential and Student's-t) are not flexible enough as they tend to over-shrink coefficients with values far away from zero, as previously shown in [18].

1.3.2 Multiplicative gamma process

The multiplicative gamma process (MGP), originally proposed in [38], is imposed on the factor loadings matrix, $\mathbf{A}$, as a global shrinkage prior to estimate the effective number of factors. We choose it over more involved approaches, such as reversible jump Markov chain Monte Carlo [39] or discrete variable selection priors such as Indian buffet processes [37], beta processes [40] or evolutionary stochastic model search [1]. The multiplicative gamma process introduces infinitely many factors to the model in a way that the variance of each column of the loadings $\mathbf{A}$ will stochastically shrink towards zero, as the index of the column increases. Alternatively, we can impose the MGP on the scaling coefficients of the factors denoted by $\boldsymbol{\Lambda}$ in (1.1) similar to [41], in which the factorization $\mathbf{A}\boldsymbol{\Lambda}\mathbf{s}_n$ is seen as a sum of K rank-one matrices weighted by the elements of $\boldsymbol{\Lambda}$, such that large indices k will have negligible impact on the full factorization, which is in essence similar to a singular value decomposition. From the model in (1.1), if $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_K) \sim \mathrm{MGP}(a_1, a_2)$, then

$$\begin{aligned} &\lambda_k \sim \mathcal{N}(0, \tau_k^{-1})\,, && \tau_k = \prod_{l=1}^{k} \delta_l\,, \\ &\delta_1 \sim \mathrm{Gamma}(a_1, 1)\,, && \delta_l \sim \mathrm{Gamma}(a_2, 1)\,, \text{ for } l \geq 2\,, \end{aligned} \tag{1.3}$$

where δ_l for $l = 1, \ldots, \infty$ are independent. Each term of $\prod_{l=1}^{k} \delta_l$ is stochastically increasing provided that $a_2 > 1$, therefore the precision of the Gaussian distribution, τ_k, will shrink λ_k towards zero as k increases. As described in [38], inference for the MGP prior in (1.3) can be done in two ways: (i) Approximate the potentially infinite number of columns of $\boldsymbol{\Lambda}$ by setting K to a reasonably large truncation level. (ii) Selecting the number of factors adaptively. Here we choose the former because, as stated in [38], it produces accurate estimates of the effective number of factors as long as the selected truncation level is large enough; further, it is computationally simpler than the adaptive approach.

One additional benefit of the MGP prior is that it helps alleviate one of the sources of lack of identifiability in factor models. Having stochastically ordered columns for $\mathbf{A}$ helps mitigate factor switching during inference, which happens due to the well known permutation ambiguity of factor models; *i.e.*, the factor models $\mathbf{A}\mathbf{\Lambda}\mathbf{s}_n$ and $\mathbf{A}\mathbf{\Lambda}\mathbf{P}^{-1}\mathbf{P}\mathbf{s}_n$, where $\mathbf{P}$ is an arbitrary permutation matrix, have the same likelihood [14]. In the case of (1.3), $\mathbf{P}$ is no longer arbitrary as $\{\tau_k\}_{k=1}^K$ couples the elements of $\mathbf{\Lambda}$, which as a result locks the ordering of columns of $\mathbf{A}$ and elements of $\mathbf{s}_n$.

1.4 Discriminative models

Consider a set of K covariates $\mathbf{s}_n$ with an associated binary label y_n. The goal of a discriminative model is to predict the label $y^\star$ of unseen covariates $\mathbf{s}^\star$. In a probabilistic model this usually amounts to estimating the joint distribution $p(y, \mathbf{s})$ from a training set $\{y_n, \mathbf{s}_n\}_{n=1}^N$, then to use the predictive distribution $p(y^\star|\mathbf{s}^\star)$ to estimate $y^\star$. In the linear case, we can parameterize the model as

$$y_n = g(\mathbf{h}^\top \mathbf{s}_n)\,, \tag{1.4}$$

where $g(\cdot) : \mathbb{R}^K \to \{-1, 1\}$ or $g(\cdot) : \mathbb{R}^K \to \{0, 1\}$ are mapping functions and $\mathbf{h} \in \mathbb{R}^K$ is a vector of classification coefficients weighting the relative contribution of each of the K covariates to the decision process.

One of the most common choices for function $g(\cdot)$ in (1.4), in Bayesian modeling, is the Heaviside step function that gives rise to probit regression, in which case it can be shown that $p(y_n = 1|\mathbf{s}_n) = \Phi(\mathbf{h}^\top \mathbf{s}_n)$, where $\Phi(\cdot)$ is the cumulative density function of a standard Gaussian distribution [12]. The popularity of the probit function is likely due to the fact that an effective and easy-to-implement inference procedure based on variable augmentation exists [28]. Another two alternatives inspired by the machine learning community are logistic regression and support vector classification, each of which has an associated loss function: log-loss and hinge-loss, respectively [12]. Since Bayesian probit regression is a well understood model, we focus the remainder of this section on recently proposed variable augmentation approaches for logistic regression and support vector classification.

1.4.1 Bayesian log-loss

The log-loss is defined as the logarithm of a Bernoulli likelihood, so for $y_n \in \{0, 1\}$ and the model in (1.4) we can write

$$\ell(y_n, \mathbf{s}_n, \mathbf{h}) = y_n \log(g(\mathbf{h}^\top \mathbf{s}_n)) + (1 - y_n) \log(1 - g(\mathbf{h}^\top \mathbf{s}_n))\,. \tag{1.5}$$

From a Bayesian perspective, instead of optimizing for $\mathbf{h}$ using the loss function in (1.5), we estimate the posterior distribution $p(\mathbf{h}|y, \mathbf{s})$ using the variable-augmentation approach proposed in [29]; to do this, we first introduce the Pólya-Gamma random variables.

A random variable X has a Pólya-Gamma distribution with parameters $b > 0$ and $c \in \mathbb{R}$, denoted $X \sim \mathrm{PG}(b, c)$, if

$$X = \frac{1}{2\pi^2} \sum_{k=1}^{\infty} \frac{g_k}{(k - 1/2)^2 + c^2/(4\pi^2)} ,$$

where each $g_k \sim \mathrm{Gamma}(b, 1)$ is an independent gamma random variable [29].

A key result of [29] is that Bernoulli likelihoods, $y_n \sim \mathrm{Bernoulli}(\sigma(\mathbf{h}^\top \mathbf{s}_n))$, parameterized by the log-odds, $\mathbf{h}^\top \mathbf{s}_n$, and the logistic function, $\sigma(x) = 1/(1 + \exp(-x))$, can be written as scale mixtures of Gaussians w.r.t Pólya-Gamma distributions; this is, if $\gamma \sim \mathrm{PG}(b, 0)$, then

$$\frac{\exp(\psi)^a}{(1 + \exp(\psi))^b} \;=\; 2^{-b} \exp(\kappa\psi) \int_0^{\infty} \exp\left(-\frac{\gamma\psi^2}{2}\right) p(\gamma) d\gamma \,, \tag{1.6}$$

where $\kappa = a - b/2$ and $\gamma|\psi \sim \mathrm{PG}(b, \psi)$. For the logistic regression model, the likelihood function can be written as

$$L(y_n|\mathbf{h}, \mathbf{s}_n) \;=\; \frac{\exp(y_n \mathbf{h}^\top \mathbf{s}_n)}{1 + \exp(\mathbf{h}^\top \mathbf{s}_n)} \,. \tag{1.7}$$

Let $a = y_n$, $b = 1$, $\psi = \mathbf{h}^\top \mathbf{s}_n$, using the Pólya-Gamma data augmentation in (1.6), we can rewrite the likelihood in (1.7) as

$$\begin{aligned} L(y_n|\mathbf{h}, \mathbf{s}_n, \gamma_n) \;&\propto\; \exp\left\{\kappa_n \mathbf{h}^\top \mathbf{s}_n - \frac{1}{2}\gamma_n (\mathbf{h}^\top \mathbf{s}_n)^2\right\} \\ &\propto\; \exp\left\{-\frac{\gamma_n}{2}\left(\mathbf{h}^\top \mathbf{s}_n - \frac{\kappa_n}{\gamma_n}\right)^2\right\} \\ &\propto\; \exp\left\{-\frac{1}{2}(z_n - \mathbf{h}^\top \mathbf{s}_n)^\top \mathbf{\Gamma} (z_n - \mathbf{h}^\top \mathbf{s}_n)\right\} , \end{aligned}$$

where $\kappa_n = y_n - \frac{1}{2}$, $z_n = \kappa_n/\gamma_n$, $\mathbf{\Gamma} = \mathrm{diag}(\boldsymbol{\gamma})$ and $\boldsymbol{\gamma} = [\gamma_1 \; \ldots \; \gamma_N]$. Therefore, the augmented model for $\mathbf{h}$ can be expressed as

$$\begin{aligned} \mathbf{h}|\mathbf{y}, \mathbf{S}, \boldsymbol{\gamma} \;&\propto p(\mathbf{h}) \prod_{n=1}^{N} L(y_n|\mathbf{h}, \mathbf{s}_n, \gamma_n) \,, \\ \gamma_n &\sim \mathrm{PG}(1, 0) \,, \end{aligned} \tag{1.8}$$

where $\mathbf{y} = [y_1 \; \ldots \; y_N]$, $\mathbf{S} = [\mathbf{s}_1 \; \ldots \; \mathbf{s}_N]$ and $p(\mathbf{h})$ is the prior for $\mathbf{h}$. Note that if $p(\mathbf{h})$ is Gaussian, due to conjugacy, the conditional posterior $p(\mathbf{h}|\mathbf{y}, \mathbf{S})$ will have the same distribution because the likelihood $L(\mathbf{y}_n|\mathbf{h}, \mathbf{s}_n, \gamma_n)$ is conditionally Gaussian *w.r.t.* $\mathbf{h}$.

1.4.2 Bayesian hinge-loss

The hinge-loss, most popular for its role in support vector machines (SVMs) [42], can be written for labels $y_n \in \{-1, 1\}$ as

$$\ell(y_n, \mathbf{s}_n, \mathbf{h}) = \max(1 - y_n \mathbf{h}^\top \mathbf{s}_n, 0) \,. \tag{1.9}$$

Minimizing (1.9) amounts to finding a decision boundary $\{\mathbf{s}_n : \mathbf{h}^\top \mathbf{s}_n = 0\}$ with associated decision function, $\mathrm{sign}(\mathbf{h}^\top \mathbf{s}_n)$, such that the distance between the so called *margin* boundaries defined as $\{\mathbf{s}_n : \mathbf{h}^\top \mathbf{s}_n = \pm 1\}$ is as large as possible, hence discriminative models based on the hinge-loss are called max-margin classifiers [42]. Unlike the log-loss, the hinge-loss only penalizes misclassifications and margin violations, which stems from the fact that $\ell(y_n, \mathbf{s}_n, \mathbf{h}) = 0$ whenever $y_n \mathbf{h}^\top \mathbf{s}_n > 1$.

Within the Bayesian framework, we can use (1.9) to form a pseudo-likelihood function that can be written as

$$L(y_n | \mathbf{h}, \mathbf{s}_n) = \exp\left\{-2 \max(1 - y_n \mathbf{h}^\top \mathbf{s}_n, 0)\right\} \,, \tag{1.10}$$

which admits a location-scale mixture of Gaussian representation, by making use of the following integral identity introduced in [43]

$$\exp\{-2 \max(u, 0)\} = \int_0^\infty \mathcal{N}(u | -\gamma, \gamma) d\gamma \,. \tag{1.11}$$

From (1.10) and (1.11) we obtain an augmented likelihood expressed as

$$L(y_n | \mathbf{h}, \mathbf{s}_n, \gamma_n) \propto \gamma_n^{-1/2} \exp\left(-\frac{1}{2} \frac{(1 + \gamma_n - y_n \mathbf{h}^\top \mathbf{s}_n)^2}{\gamma_n}\right) ,$$

which we recognize as the core of a Gaussian density with variance γ_n [44]. Interestingly, we get a similar augmented model to that of (1.8) but with conditional posterior $\gamma_n^{-1} \sim \mathrm{IG}(|1 - y_n \mathbf{h}^\top \mathbf{s}_n|, 1)$ instead of $\gamma_n \sim \mathrm{PG}(1, \mathbf{h}^\top \mathbf{s}_n)$ [29], where $\mathrm{IG}(\cdot, \cdot)$ is the inverse Gaussian distribution [43].

1.5 Discriminative factor model

In the previous two sections, we introduced factor models and discriminative models. We next consider the problem of jointly learning both, by connecting them through the factor scores matrix $\mathbf{S}$. Intuitively, instead of doing dimensionality reduction and classification separately as a two-step process, we use the discriminative model as a way to inform the factor model about the fact that its low-dimensional representation of data, $\mathbf{S}$, should be biased towards discriminative power. As a result, we obtain a factor model with two goals: explain the data via covariance structure estimation and produce a classification rule via dimensionality reduction, using factor scores as proxy for data.

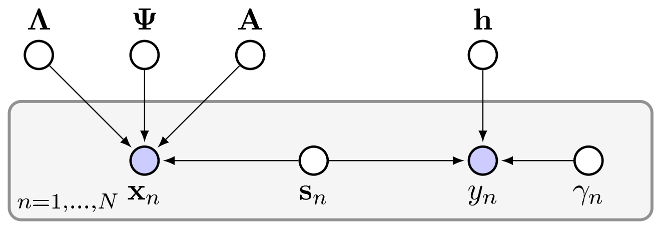

Figure 1.1 Graphical model for the discriminative factor model in (1.12). Note that $\boldsymbol{\epsilon}_n$ has been marginalized out.

Our discriminative factor-model specification with graphical model shown in Figure 1.1 can be written as

$$\begin{aligned}
\mathbf{x}_n &= \mathbf{A}\boldsymbol{\Lambda}\mathbf{s}_n + \boldsymbol{\epsilon}_n\,, \\
y_n &= g(\mathbf{h}^\top \mathbf{s}_n)\,, \\
a_{jk} &\sim \text{TPBN}(a, b, \phi_k)\,, \\
\boldsymbol{\Lambda} &\sim \text{MGP}(a_1, a_2)\,, \\
\mathbf{s}_n &\sim \mathcal{N}(\mathbf{0}, \mathbf{I})\,, \\
\boldsymbol{\epsilon}_n &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}^{-1})\,, \\
h_k &\sim \text{TPBN}(a, b, \phi)\,,
\end{aligned} \tag{1.12}$$

where $g(\mathbf{h}^\top \mathbf{s}_n)$ is the decision function corresponding to either log-loss or hinge-loss, as described in the previous section and summarized in Table 1.1.

Table 1.1 Variable augmentation specifications for classification. $\mathcal{N}(x; \mu, \sigma^2)$ is the density of a Gaussian distribution with parameters μ and σ^2.

Classifier	$L(y_n \mid \mathbf{h}, \mathbf{s}_n, \gamma_n)$	$p(\gamma_n)$	$g(\cdot)$
Probit	$I(\gamma_n > 0)$	$\mathcal{N}(\mathbf{h}^\top \mathbf{s}_n, 1)$	$I(\mathbf{h}^\top \mathbf{s}_n > 0)$
Logistic	$\mathcal{N}\left(\mathbf{h}^\top \mathbf{s}_n; (y_n - \frac{1}{2})\gamma_n^{-1}, \gamma_n^{-1}\right)$	$\text{PG}(1, 0)$	$I(\mathbf{h}^\top \mathbf{s}_n > 0)$
Max-margin	$\mathcal{N}\left(1 - y_n \mathbf{h}^\top \mathbf{s}_n; -\gamma_n, \gamma_n\right)$	$\text{Uniform}(0, \infty)$	$\text{sign}(\mathbf{h}^\top \mathbf{s}_n)$

Note that we have provided the vector of classifier weights, $\mathbf{h}$, with the same shrinkage prior imposed on the elements of the factor loadings matrix, $\mathbf{A}$. This is done to acknowledge that only a subset of the factors learned by the model will likely be geared towards discrimination, while the remaining ones may be entirely (or primarily) focused on explaining the data.

From Figure 1.1 we see that conditioned on the factor scores, $\mathbf{S}$, data and labels are conditionally independent, meaning that inference-wise $\mathbf{S}$ is the only variable in the model whose conditional posterior is both dependent on $\mathbf{X}$ and $\mathbf{y}$, data and labels, respectively. It also implies that inference for all the other variables of the model remains conveniently unchanged, when compared to say just the factor model in (1.1).

1.5.1 Multi-task learning

The proposed discriminative factor model can be readily extended to a multi-task learning setting, in which multiple binary classification problems are performed jointly. This mechanism has the benefit of information sharing across tasks, thus enhancing the representation learned by the model [23, 45, 46]. For instance, in our proposed model, the classifier coefficients may be different in multiple tasks, but the matrix factorization mechanism, *i.e.* the inferred matrices $\mathbf{A}$, $\mathbf{\Lambda}$ and $\mathbf{S}$ are shared, thereby utilizing all label information to make the model discriminative, but assuming that the original high-dimensional gene data lies in a common low-dimensional subspace for all tasks.

Consider the situation in which $c \in \{1, \ldots, C\}$ tasks performed together. The complete model can be expressed as

$$\begin{aligned} \mathbf{x}_n &= \mathbf{A}\mathbf{\Lambda}\mathbf{s}_n + \boldsymbol{\epsilon}_n \,, \\ y_n^{(c)} &= g\left((\mathbf{h}^{(c)})^\top \mathbf{s}_n \right) \,, \\ \gamma_n^{(c)} &\sim p(\gamma_n^{(c)}) \,, \end{aligned} \tag{1.13}$$

where superscript (c) represents a specific task, and $g(\cdot)$ and $p(\gamma^{(c)})$ are one of the choices in Table 1.1, depending on the classification model to be used. By comparing (1.12) and (1.13), we see that the factor model part of the hierarchy remains unchanged, and only the discriminative component reflects that we are feeding more information to the model. As a result of this, we expect the model to learn a richer low-dimensional representation, $\mathbf{S}$, thus a larger number of factors K are likely to be used.

The model in (1.13) can also be used for multi-class tasks, if we encode $\tilde{y}_n \in \{1, \ldots, C\}$, where C is the number of classes, as separate binary "tasks". For this purpose we use a *one-vs-all* encoding, in which we learn a multi-class model by learning C binary classifiers, each of which attempts to differentiate one of the classes from all the others. Once built, the classification rule is

$$\tilde{y}_n = \operatorname{argmax}_c \, (\mathbf{h}^{(c)})^\top \mathbf{s}_n \,.$$

In biologically-motivated applications, learning multiple binary classifiers, as opposed to a single multi-class model, is beneficial in the sense that we can use individual classification coefficient vectors, $\mathbf{h}^{(c)}$, to drive the interpretation of the model one phenotype at the time.

1.6 Inference

Learning the parameters of the model in (1.12) (or (1.13)) with classification specifications in Table 1.1 can be done either via Markov chain Monte Carlo (MCMC) [47, 48] or a mean field Variational Bayes (VB) approximation [49]. The MCMC algorithm involves a sequence of Gibbs updates, where each latent variable is iteratively resampled conditioned on instances of all the others.

Variational Bayes

The VB framework attempts to approximate the full posterior distribution of the model by a simpler distribution, $q(\boldsymbol{\Theta}) \approx p(\boldsymbol{\Theta}|\boldsymbol{y}, \mathbf{X})$, where $\boldsymbol{\Theta} = \{a_{jk}, \lambda_k, s_{kn}, \psi_j, \zeta_{jk}, \xi_{jk}, \phi_k, w, \delta_k, h_k, \gamma_n\}$, for $j = 1, \ldots, p$, $k = 1, \ldots, K$, denotes the set of independent latent variables in the model and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, $\boldsymbol{y} = [y_1, \ldots, y_N]$ represent observed data and labels, respectively. Specifically, VB assumes a complete factorization across latent variables, $q(\boldsymbol{\Theta}) = \prod_i q_i(\boldsymbol{\theta}_i)$, where $\boldsymbol{\theta}_i$ is an element of $\boldsymbol{\Theta}$.

Solving for the optimal distribution $q^\star(\boldsymbol{\Theta})$ that minimizes the distance between p and q effectively estimates the conditional posterior distribution $p(\boldsymbol{\Theta}|\boldsymbol{y}, \mathbf{X})$. A commonly-used distance metric between the two distributions functions is the Kullback-Leibler (KL) divergence [50]. We write the KL-divergence of p from q as follows:

$$\mathrm{KL}(q\|p) = \int_{\boldsymbol{\Theta}} q(\boldsymbol{\Theta}) \log \frac{q(\boldsymbol{\Theta})}{p(\boldsymbol{\Theta}|\boldsymbol{y}, \mathbf{X})} d\boldsymbol{\Theta},$$

through which

$$\begin{aligned} \log p(\mathbf{y}, \mathbf{X}) &= \mathrm{KL}(q\|p) + \mathcal{L}(q), \\ \mathcal{L}(q) &= -\int_{\boldsymbol{\Theta}} q(\boldsymbol{\Theta}) \log \frac{q(\boldsymbol{\Theta})}{p(\boldsymbol{\Theta}, \boldsymbol{y}, \mathbf{X})} d\boldsymbol{\Theta}. \end{aligned}$$

Observe that $\log p(\boldsymbol{y}, \mathbf{X})$ is fixed *w.r.t.* the variations in $q(\boldsymbol{\Theta})$. Therefore, maximizing the Evidence Lower Bound (ELBO), $\mathcal{L}(q)$, is equivalent to minimizing the KL-divergence between the two distributions. This minimal distance occurs when

$$\log q^\star(\boldsymbol{\Theta}) = \mathbb{E}[\log p(\boldsymbol{y}, \mathbf{X}, \boldsymbol{\Theta})] + \text{const.}$$

Assuming a complete factorization across the latent variables $q(\boldsymbol{\Theta}) = \prod_i q_i(\boldsymbol{\theta}_i)$, each parameter in a variational Bayes approximation is independently updated according to

$$q_j^\star(\boldsymbol{\Theta}_j) \propto \exp\{\mathbb{E}_{i\neq j}[\log p(\boldsymbol{y}, \mathbf{X}, \boldsymbol{\Theta})]\}.$$

For the models considered here, both MCMC and VB are straightforward to implement, because local conjugacy of all the parameters of the model allows us to write corresponding conditional posteriors in closed form. A sketch of the inference procedures, and details of conditional posterior distributions of all the parameters of the model, are found in Section 1.9.

Scaling up

Every iteration of the VB inference algorithm requires a full pass through the dataset, which can be time consuming when applied to large datasets. Therefore, an online version of VB inference can be developed, building upon a recent online implementation for latent Dirichlet allocation [51]. Online VB exploits the difference between *local* variables, which are observation dependent, and *global* variables, which are shared among the entire dataset. In the context of factor models, the data are the gene expression values of an individual, the global variables are the factor loadings (shared among all data samples), and the local variables are individual-specific factor scores.

In practice, stochastic optimization is applied to the variational objective function in the online VB representation [52]. The key observation is that the coordinate ascent updates in VB precisely correspond to the natural gradient of the variational objective function. Considering that we split the entire dataset with N observations into D mini-batches, the following steps are repeated to implement the online VB.

1. Randomly select one mini-batch; optimize its local variational parameters.
2. Obtain the current estimate of the global variational parameters, as though we were running classical coordinate ascent update on the dataset formed by repeating N/D times of the selected mini-batches.
3. Update the global variational parameters to be a weighted average of the current estimate and previous estimate.

To be more specific, the global variables are denoted as $\mathbf{\Theta_g} = \{a_{jk}, \lambda_k, \psi_j, \zeta_{jk}, \xi_{jk}, \phi_k, w, \delta_k, h_k\}$ and the local variables are denoted as $\mathbf{\Theta_l} = \{s_{kn}, \gamma_n\}$, for $j = 1, \ldots, p$, $k = 1, \ldots, K$ and $n = 1, \ldots, N$. The update of one global variable θ_g, after seeing the mini-batch indexed by l, becomes

$$\theta_g^{(l)} = (1 - \rho_l)\theta_g^{(l-1)} + \rho_l \theta_g^* \,,$$

where θ_g^* is the current estimate and $\rho_l = (\rho_0 + l)^{-\kappa}$ is an appropriately decreasing learning rate. To ensure that the global parameters converge to a stationary point, we set $\kappa \in (0.5, 1]$ and $\rho_0 > 0$. The update of local variational parameters are the same as the batch VB inference. This modeling and inference framework allows one to scale up the analysis to "big data," with a large number of biomarkers (large p) as well as a large number of sample N, although in practice we still typically deal with $N < p$.

Computational cost

The computational cost of the factor model with or without the classifier is roughly $\mathcal{O}(pK^2)$ per iteration. In our experience, we find that between 50 and 100 VB iterations are enough for the model to stabilize. It is important to take into account that the time needed to run any of the models proposed here is significantly smaller than the time needed to generate the data, thus statistical analyses performed using our models will not constitute a bottleneck in real-world applications.

Implementation and availability

All the code used for the experiments including implementations of the models considered was written in Matlab and can be found at:
`http://www.duke.edu/~rh137/host.html`.

1.7 Experiments

In this section, we present extensive experiments on two real-world microarray-based gene expression datasets, from two studies about infectious diseases. In the following, we briefly describe the data and performance measures used to evaluate the models being compared. Extensive numerical results are provided, with model interpretation delivered via pathway association analysis.

ARI dataset
The Acute Respiratory Infection (ARI) dataset is developed with the goal of differentiating bacterial from viral infections in the context of a relatively heterogeneous cohort, also containing subjects with non-infectious illnesses. It is composed of intensities of $p = 22277$ probes from Affymetrix HG-U133A 2.0 arrays with $N = 280$ subjects categorized into the following three groups: bacterial (70), viral (115) and non-infectious illness (88). For the analysis, we keep the top 25% (5569) GCRMA normalized [53] probes with largest intensity profiles.

TB dataset
This particular tuberculosis (TB) dataset[1] is the result of a recently published study [54], which consists of gene expression intensities for 47323 genes and $N = 491$ subjects measured using Illumina HumanHT-12 V4.0 expression beadchips, categorized in four phenotypes: active TB (190), latent TB (68), other diseases (233) and HIV positive (161). The raw data were preprocessed using background correction, quantile normalized signal intensities, log-transformation and gene filtering. For the analysis we keep the top $p = 4732$ genes with largest intensity profiles.

1.7.1 Performance measures

The performance of factor models is evaluated by the mean squared error (MSE) between the original observation matrix $\mathbf{X}$, and the reconstructed matrix $\hat{\mathbf{X}} = \mathbf{A}\boldsymbol{\Lambda}\mathbf{S}$, where $\mathbf{A}$, $\boldsymbol{\Lambda}$ and $\mathbf{S}$ are the inferred factor loadings, factor scaling matrix and factor scores, respectively. To be more precise, MSE is defined as $\frac{1}{N}\sum_{n=1}^{N}\sqrt{\sum_{j=1}^{p}(X_{jn} - \hat{X}_{jn})^2}$. Classification is done within a 10-fold cross validation (CV) framework and the receiver operating characteristic (ROC) [55], area under curve (AUC), classification accuracy (ACC), true positive rate (TPR) and true false rate (TNR) are reported as quantitative performance measures. CPU time is also recorded as proxy for computational cost.

1.7.2 Experimental setup

Inference is performed via VB and the maximum number of iterations is set to 60. We verified empirically that further increasing iterations does not significantly

[1] Available at: `http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE39941`.

change the outcome of any of our models. A Gibbs sampler is implemented as well, which is set to 2000 burn-in iterations and 1000 posterior collection samples. In order to address potential large-scale datasets, an on-line version of the VB inference is further developed. All computations are carried out on a 3.40GHz desktop with 12GB RAM.

When implementing the inference algorithms, we initialize **A** and **S** randomly using isotropic normals with standard deviation 1, and set $\mathbf{\Lambda}$ to be the identity matrix. To impose strong sparsity, the hyper-parameters ϕ_k, $k = 1, \ldots, K$ in the TPBN prior are all set to be 10^{-4} rather than inferred from the data. The hyper-parameters of the precision ψ_j, $j = 1, \ldots, p$ are $a_\psi = 1.1$ and $b_\psi = 10^{-3}$. For the ARI dataset, the hyper-parameters of the MGP are set to $a_1 = 2.1$ and $a_2 = 3.1$, and the truncation level of the MGP to $K = 50$. For the TB dataset, we set $a_1 = 1.1$, $a_2 = 2.1$ and $K = 100$.

1.7.3 Classification results

Recall that our model is built using a multi-task learning scheme, in which a factor model jointly learns several predictors, e.g., 4 in the TB dataset: HIV positive vs. HIV negative, active TB vs. others, latent TB vs. others and other diseases vs. all TB. The discriminative models are implemented either by probit, logistic or max-margin classifiers as previously shown in Table 1.1.

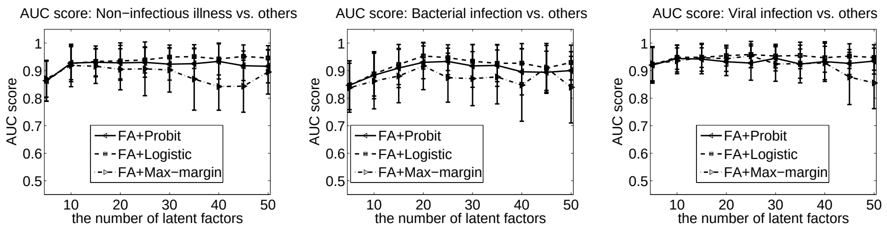

Figure 1.2 AUC values from 10-fold cross-validation on the ARI dataset using the two-step approach. (Left) Non-infectious illness. (Middle) Bacterial infection. (Right) Viral infection.

Two-step approach

For illustration, we first train the factor model and the one-vs-all classifiers separately in a two-step approach. The MGP is not utilized in the factor model in order to investigate the impact of the number of factors on the classification performance. Figures 1.2 and 1.3 show AUCs resulting from this two-step approach for the two datasets, ARI and TB, respectively. It can be seen that the logistic classifier consistently achieves a better classification performance when compared with the other two methods. Using a different number of factors results in different classification accuracies. The number of factors that yields the best performance is close 20 and 40 for ARI and TB data, respectively.

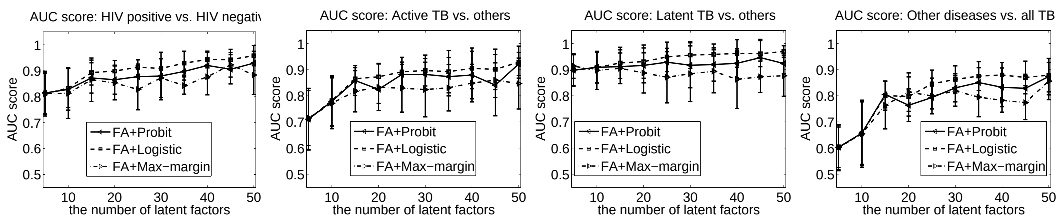

Figure 1.3 AUC values from 10-fold cross-validation on the TB dataset using the two-step approach. (Left) HIV positive. (Middle-left) Active TB. (Middle-right) Latent TB. (Right) other diseases.

Discriminative factor model

We show the results using the proposed discriminative factor model from Section 1.5 to verify that exploiting the label information during the process of factor modeling can further improve the classification performance. Further, that using the MGP prior can sidestep the model selection issue, and provides us with reasonable choices of the number of factors.

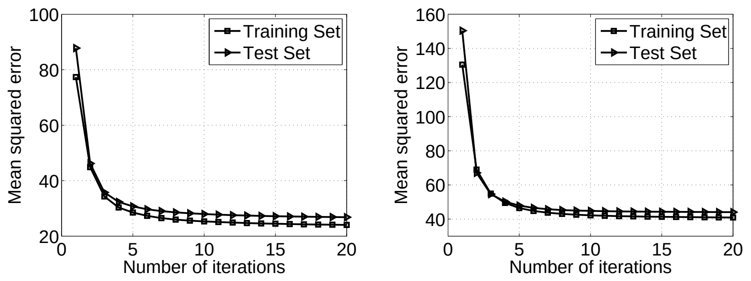

Figure 1.4 Mean squared error. (Left) ARI dataset. (Right) TB dataset. Training and test sets are displayed separately.

Figure 1.4 plots the mean squared error (MSE) between the model reconstruction and the original data. It also shows that VB inference converges quickly, i.e., 10 iterations provides a stable result. Furthermore, the MSE for training and test sets does not differ considerably, which indicates that our discriminative factor model has the ability to successfully prevent overfitting, due to the utilization of the TPBN shrinkage priors.

Since the logistic classifier empirically achieves the best performance, as discussed below, only the ROC curves for the logistic classifier are plotted in Figure 1.5. The AUCs, ACCs, TPRs and TNRs are detailed in Table 1.2, both for ARI and TB data. For each fold, we calculate AUC, ACC and MSE values, and report averaged results with corresponding error bars (standard deviations). The TPRs and TNRs are calculated on the full set of cross-validated predictions, thus no error bars are available.

From Table 1.2 we see that on average 20 and 42 factors are inferred from

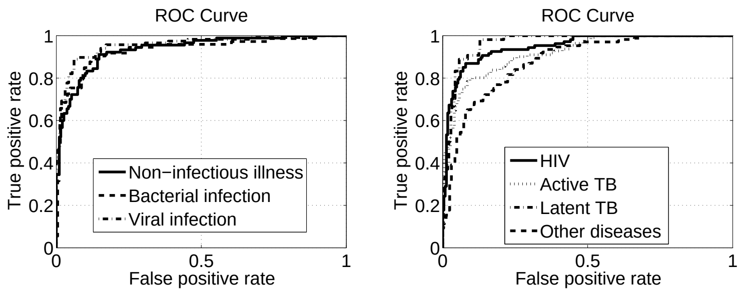

Figure 1.5 ROC curves from 10-fold cross validation. (Left) ARI dataset. (Right) TB dataset.

Table 1.2 10-fold cross-validation results for TB (left) and ARI (right) datasets.

Measure	Logistic	Max-margin	Probit	Logistic	Max-margin	Probit
	HIV positive vs. HIV negative			*Non-infectious illness vs. others*		
AUC (%)	95.18± 1.63	93.67± 3.42	92.48± 4.93	93.57± 5.39	91.35± 5.91	92.81± 5.90
ACC (%)	89.84± 3.71	87.10± 4.80	86.26± 5.90	87.50± 6.57	86.43± 6.48	85.36± 6.83
TPR (%)	85.98	82.24	79.44	90.00	81.11	84.44
TNR (%)	92.07	87.22	88.55	85.26	87.89	85.26
	Active TB vs. others			*Bacterial infection vs. others*		
AUC (%)	91.52± 6.50	86.89± 8.08	91.44± 4.87	93.85± 5.68	91.47± 9.32	92.51± 6.92
ACC (%)	86.27± 5.50	82.64± 5.26	85.95± 7.58	90.00± 6.48	84.64± 8.59	88.57± 4.39
TPR (%)	79.28	76.58	85.59	87.67	80.82	84.93
TNR (%)	89.69	82.96	78.48	87.92	87.92	88.89
	Latent TB vs. others			*Viral infection vs. others*		
AUC (%)	96.18± 4.17	93.47± 6.55	94.02± 5.28	95.61± 4.38	94.19± 5.05	94.90± 4.46
ACC (%)	93.13± 5.76	90.73± 5.56	90.42± 4.45	91.07± 5.12	89.64± 4.89	88.93± 7.23
TPR (%)	87.04	87.04	90.74	88.89	85.47	83.76
TNR (%)	94.29	85.00	84.29	92.64	92.02	95.09
	Other diseases vs. all TB					
AUC (%)	88.11± 3.88	84.93± 5.89	86.79± 4.61			
ACC (%)	79.64± 4.88	75.74± 5.96	79.05± 4.41			
TPR (%)	81.07	78.11	77.51			
TNR (%)	77.58	74.55	80.61			
Factors	42.0± 8.43	40.4± 8.88	43.6± 8.88	20.40± 3.27	19.80± 2.44	20.50± 1.96
Train MSE	39.84± 1.26	40.03± 1.42	39.59± 1.07	23.02± 0.82	23.19± 0.61	22.99± 0.56
Test MSE	45.01± 1.86	45.21± 1.86	44.95± 1.92	24.98± 1.08	25.09± 0.86	24.93± 0.78
Time (s)	3031.2± 44.2	3114.0± 25.5	3039.6± 43.4	2161.6± 15.2	2097.7± 8.2	2105.2± 22.5

ARI and TB data, respectively, which is consistent with the model selection results by choosing the model with the highest AUC scores in Figures 1.2 and 1.3. This implies our MGP prior has the ability to choose an adequate number of factors. In terms of classification, all the classifiers we have presented achieve comparable performances, among which the logistic classifier performs best. The good prediction performance indicates that our model has the ability to find the potentially important factors, thus it is able to successfully characterize the host response to the stimuli considered. Furthermore, since the TB dataset has more categories to predict, the number of inferred factors is larger compared to the ARI dataset as one may expected.

Inference via Gibbs sampling results in similar performances, thus those results are not presented, for brevity. Nevertheless, we show trace plots in Figure 1.6 by randomly selecting one element from the factor loadings $\mathbf{A}$, factor scores $\mathbf{S}$, noise precision Ψ and classifier coefficients $\mathbf{h}$, respectively. It is observed that the Gibbs sampler yields in general good mixing; the trace plot for $\mathbf{A}$ clearly demonstrates the shrinkage imposed on the factor loadings through the TPBN prior.

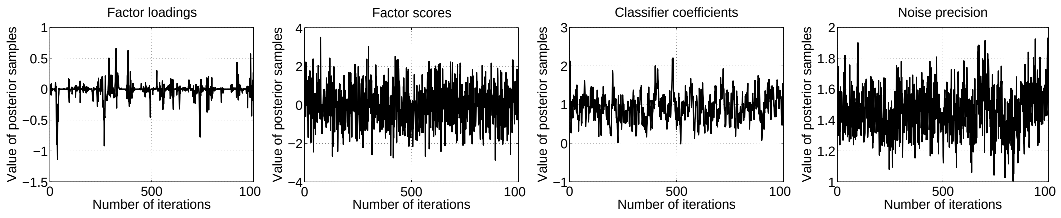

Figure 1.6 Trace plots for the Gibbs sampler. (Left) Factor loadings. (Middle-left) Factor scores. (Middle-right) Noise precision. (Right) classifier coefficients.

Online learning

To demonstrate the ability of our proposed model to scale up to large datasets, we present results on the ARI dataset, using our implementation of online VB with the logistic classifier. Local parameters are updated using 20 iterations per mini-batch, and results are shown for 4 epochs. The learning rate is $\rho_l = (\rho_0 + l)^{-\kappa}$, and we set $\rho_0 = 1$ and $\kappa = 0.5$. Results are summarized in Table 1.3. We see that online VB can achieve almost the same performance as batch VB, in terms of both factor modeling and classification; at the same time, online VB is considerably less computationally expensive.

Table 1.3 10-fold cross-validation results on the ARI dataset using online VB. The first column indicates the mini-batch size. Results in the second row are taken from Table 1.2. AUC scores correspond from left to right to non-infectious illness, bacterial and viral classifiers.

Size	CPU time (s)	MSE		AUC (%)	
252	2161.6± 15.2	24.98± 1.08	93.57± 5.39	93.85± 5.68	95.61± 4.38
126	1087.5± 3.9	25.72± 1.22	92.16± 7.51	90.62± 6.98	94.67± 4.16
63	881.3± 3.1	25.83± 1.11	92.81± 6.90	90.87± 7.41	94.63± 4.57
36	857.4± 3.7	26.68± 1.15	91.29± 6.08	90.14± 9.60	93.13± 5.89

1.7.4 Interpretation

Interpretation of our discriminative factor model is based on the factor loadings and classification weights, $\mathbf{A}$ and $\mathbf{h}$, respectively. As previously discussed, we can relate columns in $\mathbf{A}$, the factors, to networks of correlated genes and the magnitude of the weights in $\mathbf{h}$ as a measure of the relative importance of each factor to the classification outcome. Figure 1.7(left) shows classification weights for each of the three classifiers built for the ARI dataset. We see that individual factors contrast different phenotypes, for example factor 1 is specific to bacterial infection, factors 2 and 3 to viral infection, and so on. Figure 1.7(Right) shows a thresholded version of loadings matrix $\mathbf{A}$, where we have set to zero any of its elements such that $\{a_{jk} : |a_{jk}| < 3\mathrm{std}(a_{1k}, \ldots, a_{pk})\}$. This procedure not only eases visualization but helps subsequent interpretation by only focusing on the genes that have an important contribution to the factor scores, and thus to the discriminative models. Note that nearly half of the genes are not present in the thresholded loading matrix in Figure 1.7(right), meaning that they are for the most part considered by the model as "noise", thus explained by $\boldsymbol{\epsilon}_n$ rather than $\mathbf{A}\mathbf{s}_n$. After thresholding each of the 21 factors found by the model constitute gene networks of approximately 300 genes each. Interpretation of the

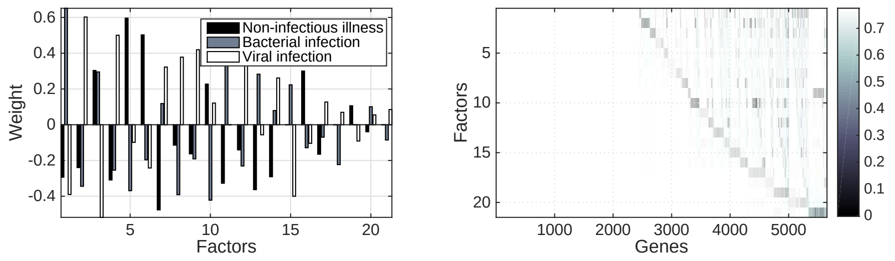

Figure 1.7 Model features for ARI data. (Left) Classification weights $\mathbf{h}$. (Right) Thresholded loading matrix $\mathbf{A}$.

gene networks encoded by the loadings matrix is done by means of a pathway association analysis using DAVID [56]. The idea is to statistically quantify the association between a set of genes (factor) to a biological theme, function or pathway. In our analysis we particularly focus on Gene Ontology (GO) terms related to molecular function, however if the application requires it one may rather target it to pathway databases or medical terms. Table 1.4 shows the size of each gene network and the top GO associations corresponding to the top 7 factors with largest absolute classification weights, as shown in Figure 1.7(left). In parentheses are the number of genes in the network associated with a particular pathway. All associations are significant at the 0.05 level with p-values corrected for multiple testing using Benjamin and Hochberg [57]. From Table 1.4 we see GO terms intimately related to host response to infectious diseases, such as, immune response, defense response and inflammatory response, but more interestingly we see some factors targeting particular biological functions, for example, factors 1 and 3 confirm our interpretation of the classifier weights from Figure 1.7(left), in terms of them being specific of bacterial and viral infections, respectively, whereas factor 6 has apoptosis as its main theme. Complete tables containing all associations, not only the statistically significant ones, p-values

Table 1.4 Pathway analysis for ARI data. GO terms associated with gene networks encoded by the columns of factor loadings matrix. All associations are significant at the 0.05 level.

ID	N	GO terms
1	252	Defense response (43), immune response (45), response to wounding (36), inflammatory response (26), positive regulation of immune system process (20), response to bacterium (18), innate immune response (15).
2	447	Golgi apparatus (46), nuclear lumen (65), intracellular organelle lumen (74), organelle lumen (75), endoplasmic reticulum (47).
3	286	Immune response (43), response to virus (13), nucleic acid transport (12), RNA transport (12), establishment of RNA localization (12), RNA localization (12).
4	301	Immune response (32), programmed cell death (27), death (28), defense response (26), positive regulation of immune system process (15)
5	291	Immune response (49), defense response (33), lymphocyte activation (16), innate immune response (13), leukocyte activation (17), cell activation (17), response to virus (11), response to vitamin (8), response to organic substance (28)
6	295	Negative regulation of cell death (21), negative regulation of programmed cell death (21), macromolecular complex subunit organization (31), immune response (31), nucleosome assembly (10), negative regulation of apoptosis (21)
7	302	Immune response (47), defense response (33), antigen processing and presentation (11), regulation of apoptosis (34), regulation of programmed cell death (34), regulation of cell death (34), vesicle-mediated transport (27)

with different correction methods and gene lists for each GO term can be found at: `http://www.duke.edu/~rh137/host.html`.

We also performed a thorough pathway association analysis for the TB dataset with results as biologically relevant as those in Table 1.4. Complete tables of association and plots similar to those in Figure 1.7 can be found at:
`http://www.duke.edu/~rh137/host.html`.

1.8 Closing remarks

This book chapter highlights the importance of Bayesian modeling for gene expression analysis. Discriminative factor models, which are the particular theme of this chapter, are presented within a principled framework to jointly build factor models and multiple classifiers. As an alternative to Bayesian classifiers based on the traditional probit link, we have integrated logistic regression and support vector classification into our modeling scheme, using novel variable augmentation techniques. We have equipped the factor models with global-local shrinkage priors, recently proposed within the machine learning community. Our model can infer the number of factors automatically from the data. We also provide extensions to multi-task learning. Inference is developed using both MCMC and variational Bayes algorithms, while online learning is further investigated to scale the model to large datasets.

It is understood that real-world datasets are getting larger and also more complex. One important research direction is to develop factor models with built-in nonlinear classifiers, since the data may not be always separable in the linear subspace implied by our factor model specification. We have made some progress with a discriminative factor model using nonlinear classifiers, based on mixtures of locally linear classifiers or support vector machines [58]. However, this new methodology still needs a considerable amount of study, thus is left as future work. Another possibility we have been considering is to relax the Gaussianity assumption implied by the likelihood function of our factor model, which may be useful in other types of omics data, such as proteomics, metabolomics and RNA sequencing. In particular, we are considering replacing the Gaussian likelihood in our model with a likelihood function for ranked data, so we can seamlessly treat ordinal, continuous and discrete data all within the same family of generalized discriminative factor models [59].

Acknowledgements

The research reported here was supported by the Defense Advanced Research Projects Agency (DARPA), under the Predicting Health and Disease (PHD) program. Ephraim L. Tsalik was supported by Award Number 1IK2CX000530 from the Clinical Science Research and Development Service of the VA Office of Research and Development. The views expressed in this article are those of the authors and do not necessarily represent the views of the Department of Veterans Affairs.

1.9 Inference details

Given all the conditional posterior distributions for all the parameters involved in (1.12), MCMC based on Gibbs sampling and mean-field VB inference can be easily implemented as briefly described in Section 1.6.

Gibbs sampling

We sample all parameters of the model from their corresponding conditional posterior distributions one at the time. We repeat this cycle of samples enough times for the model to mix; this is usually known as the “burn-in” period. Then, we keep sampling for a specified number of iterations in order to summarize the distributions of the parameters of interest (“collection” period).

Variational Bayes

Instead of sampling from the parameters of the model, we introduce a fully factorized approximation $q(\mathbf{\Theta}|\mathbf{x}_n, y_n)$ to the exact posterior $p(\mathbf{\Theta}|\mathbf{x}_n, y_n)$, for $n = 1, \ldots, N$, where $\mathbf{\Theta} = \{a_{jk}, \lambda_k, s_{kn}, \psi_j, \zeta_{jk}, \xi_{jk}, \phi_k, w, \delta_k, h_k, \gamma_n\}$, for $j = 1, \ldots, p$ and $k = 1, \ldots, K$. Inference proceeds by repeatedly updating the moments of the variational distributions until convergence is observed. In the case of local conjugacy, the moments of the variational approximation match the moments of the analytically computed conditional posteriors.

Conditional posteriors

In the remainder of this section we present the relevant conditional posteriors. For reference, $j = 1, \ldots, p$ indexes variables (genes), $n = 1, \ldots, N$ indexes observations and $k = 1, \ldots, K$ indexes factors.

Factor loadings

- Define $X_{jn}^{-k} = x_{jn} - \sum_{l \neq k}^{K} \lambda_l A_{jl} s_{ln}$, then $A_{jk}|- \sim \mathcal{N}(\mu_{jk}, \Sigma_{jk})$, where

$$\Sigma_{jk} = \left(\sum_{n=1}^{N} \psi_j \lambda_k^2 s_{kn}^2 + \zeta_{jk}^{-1} \right)^{-1}, \quad \mu_{jk} = \Sigma_{jk} \left(\sum_{n=1}^{N} \psi_j \lambda_k s_{kn} X_{jn}^{-k} \right).$$

- $\zeta_{jk}|- \sim \text{GIG}\left(0, 2\xi_{jk}, A_{jk}^2\right)$ and $\zeta_{jk}^{-1}|- \sim \text{GIG}\left(0, A_{jk}^2, 2\xi_{jk}\right)$.
- $\xi_{jk}|- \sim \text{Gamma}\left(1, \zeta_{jk} + \phi_k\right)$.
- $\phi_k|- \sim \text{Gamma}\left(\frac{1}{2}p + \frac{1}{2}, w + \sum_{j=1}^{p} \xi_{jk}\right)$.
- $w|- \sim \text{Gamma}\left(\frac{1}{2}K + \frac{1}{2}, 1 + \sum_{k=1}^{K} \phi_k\right)$.

Factor scalings

- $\lambda_k|- \sim \mathcal{N}\left(\mu_k, \sigma_k^2\right)$, where

$$\sigma_k^2 = \left(\sum_{n=1}^{N} \sum_{j=1}^{p} \psi_j A_{jk}^2 s_{kn}^2 + \tau_k \right)^{-1}, \quad \mu_k = \sigma_k^2 \left(\sum_{n=1}^{N} \sum_{j=1}^{p} \psi_j A_{jk} s_{kn} X_{jn}^{-k} \right).$$

- $\delta_1|- \sim \text{Gamma}\left(\hat{a}_1, \hat{b}_1\right)$, where

$$\hat{a}_1 = a_1 + \frac{K}{2}, \quad \hat{b}_1 = 1 + \frac{1}{2} \sum_{k=1}^{K} \tau_k^{(1)} \lambda_k^2.$$

- $\delta_h|- \sim \text{Gamma}\left(\hat{a}_h, \hat{b}_h\right)$, where $h \geq 2$ and

$$\hat{a}_h = a_2 + \frac{K - h + 1}{2}, \quad \hat{b}_h = 1 + \frac{1}{2} \sum_{k=h}^{K} \tau_k^{(h)} \lambda_k^2, \quad \tau_k^{(h)} = \prod_{l=1, l \neq h}^{k} \delta_l = \frac{\tau_k}{\delta_h}.$$

Noise variance

- $\psi_j|- \sim \text{Gamma}(g_j, h_j)$, where

$$g_j = g_0 + \frac{N}{2}\,, \quad h_j = h_0 + \frac{1}{2}\sum_{n=1}^{N}\left(x_{jn} - \mathbf{A}_j^\top \mathbf{\Lambda}\mathbf{s}_n\right)^2 .$$

When utilizing various classifiers considered, we have different update equations for $\mathbf{S}$ and $\mathbf{h}$, factor scores and classifier coefficients, respectively.

Max-margin classifier

- Define $T_{kn} = 1 - \sum_{l\neq k} y_n s_{ln} h_l$, then $s_{kn}|- \sim \mathcal{N}(\mu_{kn}, \Sigma_{kn})$, where

$$\Sigma_{kn} = \left(\sum_{j=1}^{p}\psi_j\lambda_k^2 A_{jk}^2 + 1 + \frac{h_k^2}{\gamma_n}\right)^{-1},$$
$$\mu_{kn} = \Sigma_{kn}\left(\sum_{j=1}^{p}\psi_j\lambda_k A_{jk} X_{jn}^{-k} + (1 + T_{kn}\gamma_n^{-1})y_n h_k\right).$$

- $h_k|- \sim \mathcal{N}\left(\mu_k, \sigma_k^2\right)$, where

$$\sigma_k^2 = \left(\sum_{n=1}^{N}\frac{s_{kn}^2}{\gamma_n} + \omega_k^{-1}\right)^{-1}, \quad \mu_k = \sigma_k^2\left(\sum_{n=1}^{N}(1 + T_{kn}\gamma_n^{-1})y_n s_{kn}\right).$$

- $\gamma_n^{-1}|- \sim \text{IG}\left(|1 - y_n\mathbf{h}^\top\mathbf{s}_n|^{-1}, 1\right)$

Logistic regression

- Define $T_{kn} = \gamma_n^{-1}(y_n - \frac{1}{2}) - \sum_{l\neq k} s_{ln} h_l$, then $s_{kn}|- \sim \mathcal{N}(\mu_{kn}, \Sigma_{kn})$, where

$$\Sigma_{kn} = \left(\sum_{j=1}^{p}\psi_j\lambda_k^2 A_{jk}^2 + 1 + \gamma_n h_k^2\right)^{-1},$$
$$\mu_{kn} = \Sigma_{kn}\left(\sum_{j=1}^{p}\psi_j\lambda_k A_{jk} X_{jn}^{-k} + \gamma_n h_k T_{kn}\right).$$

- $h_k|- \sim \mathcal{N}\left(\mu_k, \sigma_k^2\right)$, where

$$\sigma_k^2 = \left(\sum_{n=1}^{N}\gamma_n s_{kn}^2 + \omega_k^{-1}\right)^{-1}, \quad \mu_k = \sigma_k^2\left(\sum_{n=1}^{N}\gamma_n s_{kn} T_{kn}\right).$$

- $\gamma_n|- \sim \text{PG}\left(1, \mathbf{h}^\top\mathbf{s}_n\right)$

Probit regression

- Define $T_{kn} = \gamma_n - \sum_{l \neq k} s_{ln} h_l$, then $s_{kn}|- \sim \mathcal{N}(\mu_{kn}, \Sigma_{kn})$, where

$$\Sigma_{kn} = \left(\sum_{j=1}^{p} \psi_j \lambda_k^2 A_{jk}^2 + 1 + h_k^2 \right)^{-1},$$

$$\mu_{kn} = \Sigma_{kn} \left(\sum_{j=1}^{p} \psi_j \lambda_k A_{jk} X_{jn}^{-k} + h_k T_{kn} \right).$$

- $h_k|- \sim \mathcal{N}\left(\mu_k, \sigma_k^2\right)$, where

$$\sigma_k^2 = \left(\sum_{n=1}^{N} s_{kn}^2 + \omega_k^{-1} \right)^{-1}, \quad \mu_k = \sigma_k^2 \left(\sum_{n=1}^{N} T_{kn} s_{kn} \right).$$

- γ_n, where

$$\gamma_n | y_n = 1, - \sim \mathcal{N}(\mathbf{h}^\top \mathbf{s}_n, 1) I(\gamma_n \geq 0),$$
$$\gamma_n | y_n = -1, - \sim \mathcal{N}(\mathbf{h}^\top \mathbf{s}_n, 1) I(\gamma_n < 0).$$

References

[1]C. M. Carvalho, J. Chang, J. E. Lucas, J. R. Nevins, Q. Wang, and M. West, “High-dimensional sparse factor modeling: applications in gene expression genomics,” *Journal of the American Statistical Association*, vol. 103, no. 484, pp. 1438–1456, 2008.

[2]J. Lucas, C. Carvalho, and M. West, “A Bayesian analysis strategy for cross-study translation of gene expression biomarkers,” *Statistical Applications in Genetics and Molecular Biology*, vol. 8, no. 1, pp. 1–26, 2009.

[3]T. Speed, *Statistical analysis of gene expression microarray data.* CRC Press, 2003.

[4]S. Dudoit, J. Fridlyand, and T. P. Speed, “Comparison of discrimination methods for the classification of tumors using gene expression data,” *Journal of the American Statistical Association*, vol. 97, no. 457, pp. 77–87, 2002.

[5]I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” *Machine Learning*, vol. 46, pp. 389–422, 2002.

[6]I. Jolliffe, *Principal component analysis.* Wiley Online Library, 2005.

[7]D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Advances in Neural Information Processing Systems*, 2001.

[8]A. Hyvärinen, J. Karhunen, and E. Oja, *Independent component analysis.* John Wiley & Sons, 2004.

[9]L. Carin, J. L. Alfred Hero III, D. Dunson, M. Chen, R. Heñao, A. Tibau-Puig, A. Zaas, C. W. Woods, and G. S. Ginsburg, “High-dimensional longitudinal genomic data: An analysis used for monitoring viral infections,” *IEEE Signal Processing Magazine*, vol. 29, no. 1, pp. 108–123, 2012.

[10]M. E. Tipping, “Sparse Bayesian learning and the relevance vector machine,” *The Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[11]B. Krishnapuram, D. Williams, Y. Xue, A. Hartemink, L. Carin, and M. Figueiredo, “On semi-supervised classification,” in *Advances in Neural Information Processing Systems*, 2004.

[12]C. M. Bishop *et al.*, *Pattern recognition and machine learning.* springer New York, 2006.

[13]J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, “Bayesian factor regression models in the “large p, small n” paradigm,” *Bayesian Statistics*, vol. 7, pp. 733–742, 2003.

[14]A. M. Kagan, C. R. Rao, and Y. V. Linnik, *Characterization problems in mathematical statistics.* Wiley, 1973.

[15]H. Ishwaran and L. F. James, “Gibbs sampling methods for stick-breaking priors,” *Journal of the American Statistical Association*, vol. 96, no. 453, pp. 1–23, 2001.

[16]R. Henao and O. Winther, “Sparse linear identifiable multivariate modeling,” *The Journal of Machine Learning Research*, vol. 12, pp. 863–905, 2011.

[17]T. Park and G. Casella, “The Bayesian lasso,” *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 681–686, 2008.

[18]C. M. Carvalho, N. G. Polson, and J. G. Scott, “Handling sparsity via the horseshoe,” in *International Conference on Artificial Intelligence and Statistics*, 2009, pp. 73–80.

[19]N. G. Polson and J. G. Scott, “Shrink globally, act locally: sparse Bayesian regularization and prediction,” *Bayesian Statistics*, vol. 9, pp. 501–538, 2010.

[20]A. Armagan, D. B. Dunson, and M. Clyde, “Generalized beta mixtures of Gaussians,” in *Advances in Neural Information Processing Systems*, 2011.

[21]A. K. Zaas, M. Chen, J. Varkey, T. Veldman, A. O. Hero III, J. Lucas, Y. Huang, R. Turner, A. Gilbert, R. Lambkin-Williams *et al.*, “Gene expression signatures diagnose influenza and other symptomatic respiratory viral infections in humans,” *Cell Host & Microbe*, vol. 6, no. 3, pp. 207–217, 2009.

[22]B. Chen, M. Chen, J. Paisley, A. Zaas, C. Woods, G. S. Ginsburg, A. Hero, J. Lucas, D. Dunson, and L. Carin, “Bayesian inference of the number of factors in gene-expression analysis: application to human virus challenge studies,” *BMC Bioinformatics*, vol. 11, no. 1, pp. 1–16, 2010.

[23]M. Chen, D. Carlson, A. Zaas, C. W. Woods, G. S. Ginsburg, A. Hero, J. Lucas, and L. Carin, “Detection of viruses via statistical gene expression analysis,” *Biomedical Engineering, IEEE Transactions on*, vol. 58, no. 3, pp. 468–479, 2011.

[24]C. W. Woods, M. T. McClain, M. Chen, A. K. Zaas, B. P. Nicholson, J. Varkey, T. Veldman, S. F. Kingsmore, Y. Huang, R. Lambkin-Williams *et al.*, “A host transcriptional signature for presymptomatic detection of infection in humans exposed to influenza H1N1 or H3N2,” *PloS one*, vol. 8, no. 1, pp. e52 198, 1–9, 2013.

[25]R. Henao, J. W. Thompson, M. A. Moseley, G. S. Ginsburg, L. Carin, and J. E. Lucas, “Latent protein trees,” *Annals of Applied Statistics*, vol. 7, no. 2, pp. 691–713, 2013.

[26]A. K. Zaas, B. H. Garner, E. L. Tsalik, T. Burke, C. W. Woods, and G. S. Ginsburg, “The current epidemiology and clinical decisions surrounding acute respiratory infections,” *Trends in Molecular Medicine*, vol. 20, no. 10, pp. 579–588, 2014.

[27]J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Supervised dictionary learning,” in *Advances in Neural Information Processing Systems*, 2009.

[28]J. H. Albert and S. Chib, “Bayesian analysis of binary and polychotomous response data,” *Journal of the American Statistical Association*, vol. 88, no. 422, pp. 669–679, 1993.

[29]N. G. Polson, J. G. Scott, and J. Windle, “Bayesian inference for logistic models using Pólya–gamma latent variables,” *Journal of the American Statistical Association*, vol. 108, no. 504, pp. 1339–1349, 2013.

[30]N. G. Polson and S. L. Scott, “Data augmentation for support vector machines,” *Bayesian Analysis*, vol. 6, no. 1, pp. 1–23, 2011.

[31]N. Quadrianto, V. Sharmanska, D. A. Knowles, and Z. Ghahramani, “The supervised IBP: Neighbourhood preserving infinite latent feature models.” in *Uncertainty in Artificial Intelligence*, 2013.

[32]E. Salazar, M. S. Cain, S. R. Mitroff, and L. Carin, “Inferring latent structure from mixed real and categorical relational data,” in *International Conference on Machine Learning*, 2012.

[33]C. Hans, “Bayesian lasso regression,” *Biometrika*, vol. 96, no. 4, pp. 835–845, 2009.

[34]J. O. Berger and W. E. Strawderman, “Choice of hierarchical priors: admissibility in estimation of normal means,” *The Annals of Statistics*, pp. 931–951, 1996.

[35]J. E. Griffin and P. J. Brown, “Bayesian adaptive lassos with non-convex penalization,” University of Warwick. Centre for Research in Statistical Methodology, Tech. Rep., 2007.

[36]N. G. Polson and J. G. Scott, “Local shrinkage rules, Lévy processes and regularized regression,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 74, no. 2, pp. 287–311, 2012.

[37]T. L. Griffiths and Z. Ghahramani, “Infinite latent feature models and the Indian buffet process,” in *Advances in Neural Information Processing Systems*, 2005.

[38]A. Bhattacharya and D. B. Dunson, “Sparse Bayesian infinite factor models,” *Biometrika*, vol. 98, no. 2, pp. 291–306, 2011.

[39]H. F. Lopes and M. West, “Bayesian model assessment in factor analysis,” *Statistica Sinica*, vol. 14, no. 1, pp. 41–68, 2004.

[40]J. Paisley and L. Carin, “Nonparametric factor analysis with beta process priors,” in *International Conference on Machine Learning*, 2009.

[41]X. Zhang and L. Carin, “Joint modeling of a matrix with associated text via latent binary features,” in *Advances in Neural Information Processing Systems*, 2012.

[42]C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[43]D. F. Andrews and C. L. Mallows, “Scale mixtures of normal distributions,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 99–102, 1974.

[44]N. G. Polson, S. L. Scott *et al.*, “Data augmentation for support vector machines,” *Bayesian Analysis*, vol. 6, no. 1, pp. 1–23, 2011.

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

[46]S. Ji, D. Dunson, and L. Carin, “Multitask compressive sensing,” *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 92–106, 2009.

[47]S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.

[48]C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, “An introduction to MCMC for machine learning,” *Machine learning*, vol. 50, pp. 5–43, 2003.

[49]M. J. Beal, “Variational algorithms for approximate Bayesian inference,” *Ph.D. dissertation, University College London*, 2003.

[50]S. Kullback and R. A. Leibler, “On information and sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[51]M. Hoffman, F. R. Bach, and D. M. Blei, “Online learning for latent Dirichlet allocation,” in *Advances in Neural Information Processing Systems*, 2010.

[52]M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, “Stochastic variational inference,” *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.

[53]Z. Wu, R. A. Irizarry, R. Gentleman, F. Martinez-Murillo, and F. Spencer, “A model-based background adjustment for oligonucleotide expression arrays,” *Journal of the American Statistical Association*, vol. 99, no. 468, pp. 909–917, 2004.

[54]S. T. Anderson, M. Kaforou, A. J. Brent, V. J. Wright, C. M. Banwell, G. Chagaluka, A. C. Crampin, H. M. Dockrell, N. French, M. S. Hamilton *et al.*, “Diagnosis of childhood tuberculosis and host rna expression in africa,” *New England Journal of Medicine*, vol. 370, no. 18, pp. 1712–1723, 2014.

[55]T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[56]B. T. S. Da Wei Huang and R. A. Lempicki, “Systematic and integrative analysis of large gene lists using david bioinformatics resources,” *Nature Protocols*, vol. 4, no. 1, pp. 44–57, 2008.

[57]Y. Benjamini and Y. Hochberg, “Controlling the false discovery rate: a practical and powerful approach to multiple testing,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 289–300, 1995.

[58]R. Henao, X. Yuan, and L. Carin, “Bayesian nonlinear support vector machines and supervised factor modeling,” in *Advances in Neural Information Processing Systems*, 2014.

[59]X. Yuan, R. Henao, and L. Carin, “Non-Gaussian discriminative factor models via the max-margin rank likelihood,” *submitted*, 2014.