Non-Gaussian Discriminative Factor Models via the Max-Margin Rank-Likelihood

Xin Yuan[*] EIEXYUAN@GMAIL.COM
Ricardo Henao[*] R.HENAO@DUKE.EDU
Ephraim L. Tsalik E.T@DUKE.EDU
Duke University, Durham, NC, 27708, USA
Raymond J. Langley RLANGLEY@LRRI.ORG
Department of Immunology, Lovelace Respiratory Research Institute, Albuquerque, NM 87108, USA
Lawrence Carin LCARIN@DUKE.EDU
Duke University, Durham, NC, 27708, USA

Abstract

We consider the problem of discriminative factor analysis for data that are in general *non-Gaussian*. A Bayesian model based on the *ranks* of the data is proposed. We first introduce a new *max-margin* version of the rank-likelihood. A discriminative factor model is then developed, integrating the max-margin rank-likelihood and (linear) Bayesian support vector machines, which are also built on the max-margin principle. The discriminative factor model is further extended to the *nonlinear* case through mixtures of local linear classifiers, via Dirichlet processes. Fully local conjugacy of the model yields efficient inference with both Markov Chain Monte Carlo and variational Bayes approaches. Extensive experiments on benchmark and real data demonstrate superior performance of the proposed model and its potential for applications in computational biology.

1. Introduction

Modern applications in computational biology and bioinformatics routinely involve data coming from different sources, measured and quantified in different ways, e.g., averaged intensities in gene expression, cytokines and proteomics, or fragment counts in RNA and microRNA sequencing. However, they all share a common trait: data are rarely Gaussian, and they are often discrete, the latter due to digital technologies used for quantification. Nevertheless, a large proportion of statistical analyses performed on these data assume Gaussianity in one way or another. This is because customary preprocessing pipelines employ normalization and/or domain transformation approaches aimed at making the data as Gaussian, or at least as symmetric, as possible. For example, one popular yet simple strategy for RNA sequencing data is to rescale each sample to correct for technical variability, followed by log-transformation or quantile normalization (Dillies et al., 2013). This and many other examples have the same rationale: the data transformations are *order preserving*, while also trying to achieve a desired distribution, typically Gaussian. Figure 1 shows

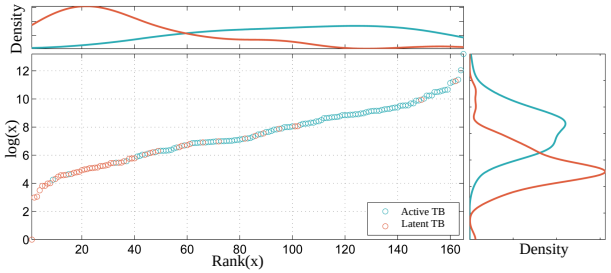

Figure 1. Intuition behind data modeling with ranks. Top and right panels are group-wise empirical distributions for $\mathrm{rank}(\mathbf{x})$ and $\log(\mathbf{x})$, respectively.

expression for a particular gene and two phenotypes (active and latent tuberculosis), from the dataset described in Section 4.2. The horizontal and vertical axes show respectively $\log(\mathbf{x})$ (log-transformed) and $\mathrm{rank}(\mathbf{x})$ (ranked) gene intensities. We see that from either axis we could derive a decision rule to separate the two groups, so that we can predict the label of new data, without worrying too much about the actual values or scaling of the axes. In fact, from their group-wise empirical distributions, we see that $\log(\mathbf{x})$ and $\mathrm{rank}(\mathbf{x})$ seem to have nearly the same optimal decision rule. Note that in general we are not required to log-transform the data, and in principle we could either use raw

[*]Equal contribution.

Proceedings of the 31st International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

data or any other monotone transformation of the gene intensities while still being able to build a classifier, if the data support it. Motivated by this fact, and also by the success of standard nonparametric statistical approaches based on ranks, such as Spearman's rank correlation and Wilcoxon rank-sum test (Lehmann & D'Abrera, 2006), in this paper, we propose a new *discriminative factor model* based directly on the *ranks* of observed data as opposed to their values. This new model enjoys three significant benefits:

(1) Since we do not model actual values, we could treat ordinal, continuous and discrete data within the same framework.

(2) We can in principle make weaker assumptions about the distribution of the data.

(3) We can jointly identify subsets of variables with similar (rank) correlation structures, some of which might be able to (partially) separate different classes and could be combined to build a classification model.

These advantages come with the price of not being able to account for the actual values of the data, which is not such a big disadvantage, as long as we are only interested in parameters of the model involving *relative* differences or similarities between elements of a given dataset (which is typically the case when building classifiers).

Modeling with ranks is not a new idea, in fact Pettitt (1982) presented a linear regression model using a likelihood function based on the ranks of observed data, coined by the authors as *rank-likelihood*. More recently it was also used by Hoff (2007) to estimate the correlation matrix of data from disparate types, e.g., binary, discrete and continuous. Here we employ the rank-likelihood as a building block for a discriminative factor model, with the ultimate goal of being able to jointly perform feature extraction and classification while decreasing the effort required to preprocess raw data.

The contributions of this paper are three-fold:

(1) We introduce a new *max-margin* version of the rank-likelihood geared towards Bayesian factor modeling, and we present a novel data augmentation scheme that allows for fast inference due to local conjugacy.

(2) We propose a discriminative factor model by integrating max-margin rank-likelihood, (linear) Bayesian support vector machines (SVMs) and global-local shrinkage priors. One key feature of our model is that likelihood functions for both data and labels have the max-margin property.

(3) We extend the discriminative factor model to *nonlinear* decision functions, through a mixture of local linear classifiers implemented via a Dirichlet process imposed on the latent space of the factor model.

Experiments on benchmark and real data, namely USPS, MNIST, gene expression and RNA sequencing, demonstrate that the proposed model often performs better than competing approaches. Results on the real data demonstrate the potential of our model for applications in computational biology, not only for well-established, high-throughput technologies such as gene expression and metabolomics, but also in emerging ones such as RNA sequencing and proteomics.

2. Max-margin rank-likelihood

Ordinal probit model Consider N data samples, each a d-dimensional vector with ordinal values; the discussion of ordinal data helps to motivate and explain the model, which is subsequently generalized to real-valued data. The data are represented by the $d \times N$ matrix $\mathbf{X}$, the nth column of which represents the nth data vector. Let $x_{i,n}$ represent element (i,n) of $\mathbf{X}$, corresponding to component i of the nth data vector, modeled as

$$\begin{aligned} x_{i,n} &= g_i(w_{i,n}) \,, \\ w_{i,n} &= \mathbf{a}_i^\top \mathbf{z}_n + v_{i,n} \,, \quad v_{i,n} \sim \mathcal{N}(0,1) \,, \end{aligned} \tag{1}$$

where $\mathbf{A} = [\mathbf{a}_1 \ \ldots \ \mathbf{a}_d]^\top \in \mathbb{R}^{d \times K}$ is the factor loadings matrix with K factors, the factor scores for all N data samples are represented by $\mathbf{Z} = [\mathbf{z}_1 \ \ldots \ \mathbf{z}_N] \in \mathbb{R}^{K \times N}$, $w_{i,n}$ is element (i,n) of $\mathbf{W} \in \mathbb{R}^{d \times N}$, and $g_i(\cdot)$ is a non-decreasing function (such that the rankings of the N realizations of component i are preserved). Specifically, large values in $\mathbf{x}_i$ (rows of $\mathbf{X}$) correspond to large values in $\mathbf{w}_i$ (rows of $\mathbf{W}$).

Assume that component i of each data vector takes values in the set $\{1, \ldots, J_i\}$. Then function $g_i(\cdot)$ can be fully specified by $J_i - 1$ ordered parameters $h_{i,1} < \cdots < h_{i,J_i-1}$, often called "cut points", yielding

$$x_{i,n} = g_i(w_{i,n}) = j \quad \text{if} \quad h_{i,j-1} < w_{i,n} < h_{i,j} \,, \tag{2}$$

where $h_{i,0} = -\infty$, $h_{i,J_i} = \infty$ and $\mathbf{h}_i = [h_{i,1} \ \ldots \ h_{i,J_i-1}]$ is a vector of thresholds for the ith row of $\mathbf{W}$. Equations (1) and (2) define a typical probit factor model for ordinal vector data (Hoff, 2009).

Inferring $\mathbf{W}$ for the model in (1) and (2) is not complicated, because its conditional posterior corresponds to a truncated Gaussian distributions, i.e., $w_{i,n} \sim \mathcal{TN}(\mathbf{a}_i^\top \mathbf{z}_n, 1, h_{i,j-1}, h_{i,j})$, where $h_{i,j-1} < w_{i,n} < h_{i,j}$ for $j = x_{i,n}$. Nevertheless, the model has three important shortcomings: (i) Specifying a prior distribution for $\{\mathbf{h}_i\}_{i=1}^d$ might be difficult because often such information is not available to the practitioner; (ii) if J_i is large, the number of parameters of the model that need to be estimated increases substantially, making prior specification and inference harder; (iii) sampling from a truncated Gaussian distribution can be relatively expensive, and may be

prone to numerical instabilities, especially when samples lie near the tails of the distribution.

Rank-likelihood model Provided that $g_i(w_{i,n})$ is non-decreasing by assumption, we know that given $x_{i,n} < x_{i,n'}$, then $g_i(w_{i,n}) < g_i(w_{i,n'})$ and $w_{i,n} < w_{i,n'}$, thus

$$R(\mathbf{x}_i) = \{\mathbf{w}_i \in \mathbb{R}^N : w_{i,n} < w_{i,n'} \text{ if } x_{i,n} < x_{i,n'}\}, \quad (3)$$

where $R(\mathbf{x}_i)$ is the set of all possible vectors $\mathbf{w}_i$ such that $\text{rank}(\mathbf{x}_i) = \text{rank}(\mathbf{w}_i)$. Given $\mathbf{x}_i$, since neither $\mathbf{w}_i$ nor $p(\mathbf{w}_i \in R(\mathbf{x}_i))$ depend on $g_i(\mathbf{w}_i)$, we can formulate inference for $\mathbf{A}$ and $\mathbf{Z}$ directly in terms of $\mathbf{w}_i \in R(\mathbf{x}_i)$ through the *rank-likelihood* representation $p(\mathbf{w}_i \in R(\mathbf{x}_i)|\mathbf{A}, \mathbf{Z})$ (Pettitt, 1982). Specifically, we can write the joint probability distribution of the model above as

$$\prod_{i=1}^d p(\mathbf{w}_i \in R(\mathbf{x}_i), \mathbf{a}_i, \mathbf{Z}) =$$
$$p(\mathbf{A})p(\mathbf{Z}) \prod_{i=1}^d \left\{ \int_{R(\mathbf{x}_i)} \prod_{n=1}^N \mathcal{N}(w_{i,n}; \mathbf{a}_i^\top \mathbf{z}_n, 1) dw_{i,n} \right\}. \quad (4)$$

The integrals to the right hand side of (4) are in general difficult to compute. However, it is not necessary to do so, because we can estimate the posterior of parameters $\{\mathbf{A}, \mathbf{Z}, \mathbf{W}\}$ via Gibbs sampling, by iteratively cycling through their conditional posterior distributions. For $\mathbf{A}$ and $\mathbf{Z}$ we need to sample from $p(\mathbf{Z}|\mathbf{W}, \mathbf{A})$ and $p(\mathbf{A}|\mathbf{W}, \mathbf{Z})$, respectively, where we instantiate $\mathbf{W}$ such that $\mathbf{w}_i \in R(\mathbf{x}_i)$ for $i = 1, \ldots, d$. For $\mathbf{w}_i$ we only need to be able to sample $\mathbf{w}_i$ from $p(\mathbf{w}_i \in R(\mathbf{x}_i)|\mathbf{a}_i, \mathbf{Z})$. We can write $p([w_{i,n}\ \mathbf{w}_{i\backslash n}] \in R(\mathbf{x}_i), \mathbf{a}_i, \mathbf{Z})$, where $\mathbf{w}_{i\backslash n}$ contains all elements from $\mathbf{w}_i$ but $w_{i,n}$. From (1) and (3), $w_{i,n}$ is Gaussian and restricted to the set $R(\mathbf{x}_i)$, respectively. Conditioning on $\mathbf{z}_{i\backslash n}$, we have

$$\begin{aligned} p(w_{i,n}|\mathbf{w}_{i\backslash n}, \mathbf{a}_i, \mathbf{z}_n) &= p(w_{i,n}|w_{i,n}^l, w_{i,n}^u, \mathbf{a}_i, \mathbf{z}_n) \\ &= \mathcal{TN}(\mathbf{a}_i^\top \mathbf{z}_n, 1, w_{i,n}^l, w_{i,n}^u), \end{aligned}$$

where $w_{i,n}^l = \max\{w_{i,n'} : x_{i,n'} < x_{i,n}\}$ and $w_{i,n}^u = \min\{w_{i,n'} : x_{i,n} < x_{i,n'}\}$, which jointly guarantee that $[w_{i,n}\ \mathbf{w}_{i\backslash n}] \in R(\mathbf{x}_i)$. Note that given $\{w_{i,n}^l, w_{i,n}^u\}$, $w_{i,n}$ is conditionally independent of $\mathbf{w}_i \backslash \{w_{i,n}^l, w_{i,n}^u\}$ and also that the conditional posteriors for the ordered probit and rank-likelihood based models are identical except that for the former, constraints for $w_{i,n}$ come from $\mathbf{w}_{i\backslash n}$ as opposed to thresholds $\mathbf{h}_i$. In fact, the rank-likelihood model can be seen as an alternative to the ordered probit model in which the threshold variables have been marginalized out (Hoff, 2009). It is important to point out that in applications when the connection between observed and latent variables, $g_i(\mathbf{w}_i)$, is of interest, the rank-likelihood is not applicable. Fortunately, in factor models we are usually only interested in $\mathbf{A}$ and $\mathbf{Z}$, the loadings and the factor scores, respectively (see Murray et al. (2013), for example).

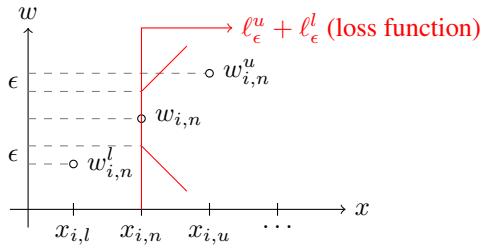

Figure 2. Graphical representation of the loss function associated to the max-margin rank-likelihood in (6), where $\ell_\epsilon^u + \ell_\epsilon^l = \ell_\epsilon(w_{i,n} - w_{i,n}^u) + \ell_\epsilon(w_{i,n}^l - w_{i,n})$. Note that $w_{i,n}^l + \epsilon < w_{i,n} < w_{i,n}^u - \epsilon$ is not penalized by the loss function.

Max-margin rank-likelihood One disadvantage of the rank-likelihood model is that differences between elements of $\mathbf{w}_i$ can be *arbitrarily small*, as there is no mechanism in the prior distribution of $\mathbf{w}_i$ to prevent this from happening. In the ordered probit model we can do so via the prior for the thresholds $\mathbf{h}_i$, however this is not necessarily easily accomplished. Fortunately, for the rank-likelihood we can alleviate this issue by modifying (3) as

$$R_{\text{mm}}(\mathbf{x}_i) = \{\mathbf{w}_i \in \mathbb{R}^N : w_{i,n} < w_{i,n'} - \epsilon \text{ if } x_{i,n} < x_{i,n'}\}, \quad (5)$$

where we have made explicit that any two distinct elements of $\mathbf{w}_i$ must be separated by a gap of size no smaller than $\epsilon > 0$. Furthermore, $\max\{w_{i,n'} : x_{i,n'} < x_{i,n}\} + \epsilon < w_{i,n} < \min\{w_{i,n'} : x_{i,n} < x_{i,n'}\} - \epsilon$. From (5) we can write a pseudo-likelihood for $w_{i,n}$ as

$$L_i(w_{i,n}|\mathbf{w}_{i\backslash n}) = \exp\left\{-\ell_\epsilon(w_{i,n} - w_{i,n}^u) - \ell_\epsilon(w_{i,n}^l - w_{i,n})\right\}, \quad (6)$$

where $w_{i,n} = \mathbf{a}_i^\top \mathbf{z}_n$, $w_{i,n}^u = \mathbf{a}_i^\top \mathbf{z}_n^u$, $w_{i,n}^l = \mathbf{a}_i^\top \mathbf{z}_n^l$ and $\ell_\epsilon(u) = 2\text{max}(0, u + \epsilon)$ can be interpreted as the "one-sided" ϵ-sensitive loss. From (6) this means that $\ell_\epsilon(u) > 0$ only if $w_{i,n} < w_{i,n}^l + \epsilon$ or $w_{i,n} > w_{i,n}^u - \epsilon$; it also means that this loss function does not penalize $w_{i,n}^l + \epsilon < w_{i,n} < w_{i,n}^u - \epsilon$ and ϵ is called the *margin*. See Figure 2 for a graphical representation of the proposed composite loss function. Maximizing (6) is equivalent to finding $\mathbf{w}_i \in R_{\text{mm}}(\mathbf{x}_i)$ such that differences between neighbor elements of $\mathbf{w}_i$ are maximal given ϵ, hence the term *max-margin* is used.

Recent work by Polson & Scott (2011b) has shown that $\ell_\epsilon(u)$ admits a location-scale mixture of Gaussians, specifically, they showed that $\exp\{-2\text{max}(0, u)\} = \int \mathcal{N}(u; -\lambda, \lambda) d\lambda$, thus we can rewrite (6) as

$$\begin{aligned} L_i(w_{i,n}|\mathbf{w}_{i\backslash n}) = \int & \mathcal{N}(w_{i,n} - w_{i,n}^u; -\epsilon - \lambda_{i,n}^u, \lambda_{i,n}^u) \\ & \times \mathcal{N}(w_{i,n}^l - w_{i,n}; -\epsilon - \lambda_{i,n}^l, \lambda_{i,n}^l) d\lambda_{i,n}^u d\lambda_{i,n}^l, \quad (7) \end{aligned}$$

where $\mathcal{N}(u; \cdot)$ is the density function of a Gaussian distribution, and we have introduced two sets of latent variables $\{\lambda_{i,n}^u\}$ and $\{\lambda_{i,n}^l\}$. This *data augmentation* scheme

implies that the pseudo-likelihood before marginalization, $L_i(w_{i,n}|\mathbf{w}_{i\backslash n}, \lambda^u_{i,n}, \lambda^l_{i,n})$, is conjugate to a Gaussian distribution, just as in the original rank-likelihood formulation, but without the difficulty of truncated Gaussians, because $w_{i,n}$ is now exactly $\mathbf{a}_i^\top \mathbf{z}_n$, not a random variable. Note that as a result of this, we have transferred the *uncertainty* between the rank of $x_{i,n}$ and the factorization $\mathbf{a}_i^\top \mathbf{z}_n$ from $w_{i,n}$ in the rank-likelihood (and the ordered probit) to the set of location-scale parameters $\{\lambda^u_{i,n}, \lambda^l_{i,n}\}$ in our max-margin formulation.

Discrete and continuous data So far we have assumed that we have ordinal data (the cut points of the ordinal model help explain the associated mechanics of the rank-likelihood model). However, we can also use the rank-likelihood with discrete or continuous data, as long as we acknowledge that likelihood and posteriors derived from them only contain information about the order of the observations and not their actual values.

In general terms, factor models are concerned with learning about the covariance structure of observed data via a low rank matrix decomposition, $\mathbf{AZ}$. In this sense, the role of the likelihood is to define the way in which covariances are measured. This means that one important difference between standard Gaussian and rank-likelihood based factor models is that they use different notions of covariance; very much in the same spirit of the differences between Pearson and Spearman correlations. Another important difference is the generative mechanism implied by the likelihood. In the rank-likelihood, we can generate a statistic, namely the rank, based on a sample population but not their values. This happens because we ignore the part of the model that links the statistic with actual data, specifically

$$\begin{aligned} p(\mathbf{x}_i|\mathbf{a}_i, \mathbf{Z}, \mathbf{h}_i) &= p(\mathrm{rank}(\mathbf{x}_i), \mathbf{x}_i|\mathbf{a}_i, \mathbf{Z}, \mathbf{h}_i) \\ &= p(\mathrm{rank}(\mathbf{x}_i)|\mathbf{a}_i, \mathbf{Z}) p(\mathbf{x}_i|\mathrm{rank}(\mathbf{x}_i), \mathbf{a}_i, \mathbf{Z}, \mathbf{h}_i)\,. \end{aligned}$$

We infer $\mathbf{a}_i$ and $\mathbf{Z}$ strictly from $p(\mathrm{rank}(\mathbf{x}_i)|\mathbf{a}_i, \mathbf{Z})$, the marginal likelihood, via $\mathbf{w}_i \in R_{\mathrm{mm}}(\mathbf{x}_i)$. Since we ignore $p(\mathbf{x}_i|\mathrm{rank}(\mathbf{x}_i), \mathbf{a}_i, \mathbf{Z}, \mathbf{h}_i)$, we do not infer the thresholds $\mathbf{h}_i$, thus in the strictest sense we are not using all information provided by $\mathbf{x}_i$, i.e., its values, however we are assuming that ranks alone contain enough information to be able to characterize the covariance structure of the data so we can reliably estimate $\mathbf{A}$ and $\mathbf{Z}$. Additional examples and further discussion of Bayesian analysis employing similar marginal likelihood strategies can be found in Monahan & Boos (1992).

3. Bayesian SVM based discriminative factor model

When the data being analyzed belong to two different classes, encoded as $\{-1, 1\}$, labels $\mathbf{y} = [y_1 \ \ldots \ y_N]^\top \in \{-1, 1\}$ will encourage our factor model to learn discriminative features (loadings and scores) from the data, then these features can be used to make predictions for new data. This modeling approach is commonly known as supervised dictionary learning or discriminative factor analysis (Mairal et al., 2008). From a Bayesian perspective, factor models and probit/logit link based classifiers have been already successfully combined; see for instance Salazar et al. (2012); Quadrianto et al. (2013).

Unlike previous work, we continue with the max-margin theme and develop a supervised factor model using Bayesian support vector machines (SVMs). The same result from Polson & Scott (2011b) used above to derive the max-margin rank-likelihood provides a pseudo-likelihood for the hinge loss, traditionally employed in the context of SVMs (Polson & Scott, 2011b). Specifically,

$$\begin{aligned} L_n(y_n|\boldsymbol{\beta}, \mathbf{z}_n) &= \exp\{-2\max(0, u_n)\} \\ &= \int_0^\infty \frac{1}{\sqrt{2\pi\lambda^c_n}} \exp\left(-\frac{1}{2}\frac{(u_n + \lambda^c_n)^2}{\lambda^c_n}\right) d\lambda^c_n\,, \end{aligned} \quad (8)$$

where $u_n = 1 - y_n \boldsymbol{\beta}^\top \mathbf{z}_n$, $\boldsymbol{\beta} \in \mathbb{R}^K$ is a vector of classifier coefficients and $\{\lambda^c_n\}$ is a vector of latent variables, with superscript c denoting the classifier. In Polson & Scott (2011b) covariates, $\mathbf{z}_n$, are observed while here they are latent variables (factor scores) that need to be estimated jointly with the remaining parameters of a factor model. It has been shown empirically that linear margin-based classifiers, SVMs being a special case, often perform better than those using logit or probit links (Polson & Scott, 2011a; Henao et al., 2014).

Interestingly, in our factor model the max-margin mechanism plays two roles, i.e., data and labels are both connected to the factor model core via max-margin pseudo-likelihoods: rank-likelihood for the data and hinge loss for the labels. Furthermore, for the loadings, since shrinkage for $\mathbf{A}$ is usually a requirement for interpretability or when $N \ll d$, here we use a three-parameter-beta normal prior ($\mathcal{TPBN}$) (Armagan et al., 2011), a fairly general global-local shrinkage prior (Polson & Scott, 2010), for which it has been demonstrated that it has better mixing properties than priors such as spike-slab (Carvalho et al., 2010). Shrinkage for the elements of $\boldsymbol{\beta}$ is also employed, because it allows us to identify the features of $\mathbf{A}$ that contribute to the classification task. Intuitively, we can see $\mathbf{A}$ as a dictionary with K features, each feature explaining a subset of the input variables due to shrinkage; via separate shrinkage within the model, $\boldsymbol{\beta}$ selects from the K features to build a predictor for labels $\mathbf{y}$. Being able to specify global and local properties independently makes the $\mathcal{TPBN}$ prior attractive for high-dimensional settings, such as gene expression and RNA sequencing, which are precisely the types of data we will focus on in our experiments.

Linear discriminative factor model By imposing the max-margin rank-likelihood construction in (5) to $\mathbf{X}$ and the hinge loss to $\mathbf{y}$ via pseudo-likelihoods in (7) and (8), respectively, one possible prior specification for the supervised factor model parameterized by $\{\mathbf{A}, \mathbf{Z}, \boldsymbol{\beta}\}$ is

$$a_{i,k} \sim \mathcal{TPBN}(r_a, s_a, \Phi_k^{(a)}), \quad \mathbf{z}_n \sim \mathcal{N}(0, \mathbf{I}_K),$$
$$\beta_k \sim \mathcal{TPBN}(r_\beta, s_\beta, \Phi^{(\beta)}),$$

where $\Phi_k^{(a)}, \Phi^{(\beta)}$ are global shrinkage parameters for loadings $\mathbf{A}$ and classifier coefficients $\boldsymbol{\beta}$. Furthermore, for the $\mathcal{TPBN}$ prior we can write

$$a_{i,k} \sim \mathcal{N}(0, \xi_{i,k}), \qquad \beta_k \sim \mathcal{N}(0, b_k),$$
$$\xi_{i,k} \sim \mathrm{Ga}(r_a, \eta_{i,k}), \qquad b_k \sim \mathrm{Ga}(r_\beta, e_k),$$
$$\eta_{i,k} \sim \mathrm{Ga}(s_a, \Phi_k^{(a)}), \qquad e_k \sim \mathrm{Ga}(s_\beta, \Phi^{(\beta)}).$$

Setting $r_a = s_a = \frac{1}{2}$ (for $\boldsymbol{\beta}$, it is $r_\beta = s_\beta = \frac{1}{2}$), a special case of $\mathcal{TPBN}$ corresponds to the widely known horseshoe prior (Carvalho et al., 2010). Note that each column of the loadings has a different global shrinkage parameter $\Phi_k^{(a)}$, thus allowing them to have different degrees of shrinkage. We can also infer $\Phi_k^{(a)}$ (and $\Phi^{(\beta)}$) by letting $\Phi_k^{(a)} \sim \mathrm{Ga}(\frac{1}{2}, \tilde{\Phi})$ and $\tilde{\Phi} \sim \mathrm{Ga}(\frac{1}{2}, 1)$. As a result of having individual shrinkage parameters for each column of $\mathbf{A}$, we could say that the prior is capable of "turning off" unnecessary factors, hence having an automatic relevance determination flavor to it (MacKay, 1995; Wipf & Nagarajan, 2008). This is indeed the behavior we see in practice; there are other ways to select the number of factors, e.g., by adding a multiplicative gamma prior to matrix $\mathbf{A}$ (Bhattacharya & Dunson, 2011).

Non-Linear discriminative factor model When the latent space for factor scores, $\mathbf{z}_n$, is not linearly separable, nonlinear classification approaches might be more appropriate. One may use a kernel to extend the linear SVM to its nonlinear counterpart. However, from a Bayesian factor modeling perspective, adding kernel-based nonlinear classifiers is nontrivial, because they tend to make inference complicated and computationally expensive due to loss of conjugacy for the parameters involved in the nonlinear component of the model, i.e., the kernel function. From a different perspective, it is still possible to build a *global nonlinear* decision rule as a mixture of *local linear* classifiers (Shahbaba & Neal, 2009; Fu et al., 2010). The basic idea is to assume factor scores as coming from a mixture model, in which each mixture component has an associated *local* linear Bayesian SVM. Here we use a Dirichlet process (DP) in its stick-breaking construction (Sethuraman, 2001), represented as

$$G = \sum_{t=1}^{\infty} q_t \delta_{\boldsymbol{\theta}_t^*}, \qquad q_t = \nu_t \prod_{l=1}^{t-1}(1-\nu_l),$$
$$\nu_t \sim \mathrm{Beta}(1, \alpha), \qquad \boldsymbol{\theta}_t^* \sim G_0, \tag{9}$$

where $\sum_{t=1}^{\infty} q_t = 1$, $\delta_{\boldsymbol{\theta}_t^*}$ represents a point measure at $\boldsymbol{\theta}_t^*$ and α is the concentration parameter. Applied to our model, factor scores and labels are drawn from a parametric model $y_n, \mathbf{z}_n \sim f(\boldsymbol{\theta}_n)$ with parameters $\boldsymbol{\theta}_n$, where $\boldsymbol{\theta}_n \sim G$. For G as in (9) and a finite number of samples N, many of the $\{y_n, \mathbf{z}_n\}$ share the same parameters, therefore making $\{y_n, \mathbf{z}_n\}$ a draw from a mixture model. Specifically, we make $f(\boldsymbol{\theta}_n) = L_n(y_n|\boldsymbol{\beta}_n, \mathbf{z}_n)\mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}_n, \psi_n^{-1}\mathbf{I}_K)$, $G_0 = \mathcal{TPBN}(\boldsymbol{\beta}|r_\beta, s_\beta, \Phi^{(\beta)}) \times \mathcal{N}(\boldsymbol{\mu}|\mathbf{0}, \mathbf{I}_K) \times \mathrm{Ga}(\psi|\psi_s, \psi_r)$ and $\{\boldsymbol{\beta}_n, \boldsymbol{\mu}_n, \psi_n\} = \{\boldsymbol{\beta}_t, \boldsymbol{\mu}_t, \psi_t\}$ if sample n belongs to the t-th component of the mixture. In practice we truncate the sum in (9) to T terms to make inference easier (Ishwaran & James, 2001) and set $\psi_s = 1.1$ and $\psi_r = 0.001$ (i.e., a non-informative prior).

Predictions Making predictions for new data using our model is conceptually simple. We use the pair $\{\mathbf{y}, \mathbf{X}\}$ to estimate the parameters of the model (training), namely $\{\mathbf{A}, \mathbf{Z}, \boldsymbol{\beta}\}$, then given a test point $\mathbf{x}_\star$, we go through three steps: (i) Compare $\mathbf{x}_\star$ to $\mathbf{X}$ to determine the rank of each component of $\mathbf{x}_\star$ w.r.t. to the training data, which amounts to finding $\{w_{i,\star}^l, w_{i,\star}^u\}$, for $i = 1, \ldots, d$. (ii) For fixed $\{\mathbf{A}, w_{i,\star}^l, w_{i,\star}^u\}$, estimate $\mathbf{z}_\star$ from its conditional posterior. (iii) Make a prediction for $\mathbf{x}_\star$ using $\mathrm{sign}(\boldsymbol{\beta}^\top \mathbf{z}_\star)$.

The first step of this prediction process is exclusive to the proposed rank-likelihood model, and implies that we are required to keep the training data in order to make predictions. This is in the same spirit of supervised kernel methods, in the sense that predictions are a function of the data used to fit the model (Scholkopf & Smola, 2001). Note, however, that for a single component of a test point, $x_{i,\star}$, we only need two elements of the training set: the two elements from $\mathbf{x}_i$ closest to $x_{i,\star}$ from above and below, which is closely related to the k-nearest neighbor paradigm (rather than k nearest neighbors, we only require the two training neighbors "to the left and right" of a test data component). Intuitively, what our model does at prediction is to find a latent representation $\mathbf{z}_\star$ such that $\mathbf{x}_\star$ is in between but as far as possible from its upper and lower bounds w.r.t. to $\mathbf{X}$. This is a very unique characteristic of our model.

Inference Due to fully local conjugacy, we can write the conditional posterior distribution for all parameters of our model in closed form, making Markov Chain Monte Carlo (MCMC) inference based on Gibbs sampling a straightforward procedure. Space limitations prevent us from presenting the complete set of conditionals, however below we show expressions for the parameters involving the max-margin rank-likelihood in (7), namely $\mathbf{A}$ and $\mathbf{Z}$. For convenience, we denote

$$\Gamma_{k,n} = \frac{y_n \beta_k [1 + \lambda_n^c - y_n(\boldsymbol{\beta}^\top \mathbf{z}_n)_{\setminus k}]}{\lambda_n^c},$$
$$\lambda_{i,n} = (\lambda_{i,n}^l)^{-1} + (\lambda_{i,n}^u)^{-1},$$
$$(\boldsymbol{\beta}^\top \mathbf{z}_n)_{\setminus k} = \boldsymbol{\beta}^\top \mathbf{z}_n - \beta_k z_{k,n},$$

In the following conditional posterior-distributions, "·" refers to the conditioning parameters of the distributions.

Sampling $\mathbf{A}$:

$$\begin{aligned} p(a_{i,k}|\cdot) &= \mathcal{N}(\mu_{a_{i,k}}, \sigma^2_{a_{i,k}}) \,, \\ \sigma^{-2}_{a_{i,k}} &= \xi^{-1}_{i,k} + \textstyle\sum_{n=1}^{N} z^2_{k,n} \lambda^{-1}_{i,n} \,, \\ \mu_{a_{i,k}} &= \sigma^2_{a_{i,k}} \textstyle\sum_{n=1}^{N} z_{k,n} \Delta^{(k)}_{i,n} \,, \end{aligned}$$

Sampling $\mathbf{Z}$:

$$\begin{aligned} p(z_{k,n}|\cdot) &= \mathcal{N}(\mu_{z_{k,n}}, \sigma^2_{z_{k,n}}) \,, \\ \sigma^{-2}_{z_{k,n}} &= 1 + \textstyle\sum_{i=1}^{d} a^2_{i,k} \lambda^{-1}_{i,n} + \frac{\beta^2_k}{\lambda^c_n} \,, \\ \mu_{z_{k,n}} &= \sigma^2_{z_{k,n}} \left(\textstyle\sum_{i=1}^{d} a_{i,k} \Delta^{(k)}_{i,n} + \Gamma_{k,n} \right) , \end{aligned}$$

where

$$\Delta^{(k)}_{i,n} = \left(\frac{w^l_{i,n} + \epsilon - w_{i,n}}{\lambda^l_{i,n}} - \frac{w_{i,n} + \epsilon - w^u_{i,n}}{\lambda^u_{i,n}} \right) + a_{i,k} z_{k,n} \lambda_{i,n} \,.$$

Conditional posteriors for the remaining parameters: $\{\lambda^l_{i,n}, \lambda^u_{i,n}, \lambda^c_n, \boldsymbol{\beta}\}$ and $\{\xi_{i,k}, \eta_{i,k}, \Phi^{(a)}_k, b_k, e_k, \Phi^{(\beta)}\}$ can be found in Polson & Scott (2011b) and Armagan et al. (2011), respectively. In our experiments we set $\epsilon = 0.05$, however a conjugate prior (gamma distribution) exists hence ϵ can be inferred if desired. Inference details for the DP specification for the factor scores can be found for instance in Ishwaran & James (2001); Neal (2000).

In applications where speed is important, we can use all conditional posteriors including those above to derive a variational Bayes (VB) inference algorithm for our model, which loosely amounts to replacing the conditioning on variables with their corresponding moments. Details of the conditionals are not shown here for brevity, and details of the VB procedure are found in the Supplementary Material.

Other related work For ordinal data, Xu et al. (2013) presented a factor model using the ordered probit mechanism, but in which the probit link is replaced with a max-margin pseudo-likelihood. Inference is very efficient, but they still have to infer the thresholds $\{\mathbf{h}_i\}$. However, in their collaborative prediction applications, variables only take one of six possible values. For count data, Chib et al. (1998) proposed a generalized-linear-model inspired Bayesian model for Poisson regression, that can be easily extended to a factor model. However, expensive Metropolis-Hastings sampling algorithms need to be used, due to the non-conjugacy between the prior for $\mathbf{A}$ and the log link. More recently, Zhou et al. (2012) presented a novel formulation of Poisson factor analysis (PFA), based on the beta-negative binomial process, for which inference is efficient. Furthermore, none of the approaches discussed above consider discriminative factors models, and for PFA this is very difficult, because in that case the prior for the factor scores is not a Gaussian distribution and is thus not conjugate to the SVM or probit-based likelihoods. As a result, building discriminative factor models under that framework is challenging, at least not without Metropolis-Hastings style inference.

Table 1. Composition of different methods.

Method	Likelihood	Classifier	DPM
G-L-Probit	Gaussian	probit	No
G-L-BSVM	Gaussian	BSVM	No
OR-L-Probit	ordinary rank	probit	No
OR-L-BSVM	ordinary rank	BSVM	No
R-L-BSVM	max-margin rank	BSVM	No
G-NL-BSVM	Gaussian	BSVM	Yes
R-NL-BSVM	max-margin rank	BSVM	Yes

4. Experiments

We present numerical results on two benchmark (USPS and MNIST) and two real (gene expression and RNA sequencing) datasets, using part of or all methods summarized in Table 1; inference is performed via VB. The data *likelihood* can be either *Gaussian*, *rank* or the *max-margin* rank-likelihood. The labels (classifier) can be modeled using the *probit* link or the Bayesian SVM (BSVM) pseudo-likelihood. When the DP mixture (DPM) model is used, the classifier results in a nonlinear (NL) decision function. Everywhere we set $K = 20$, $T = 5$ and performance measures were averaged over 5 repetitions (standard deviations are also presented). We verified empirically that further increasing K or T does not significantly change the outcome of any of our models. All code used in the experiments was written in Matlab and executed on a 3.3GHz desktop with 16Gb RAM.

In the following experiments we focus on comparing discriminative factor models against each other to show how each component of the model contributes to the end performance produced by our full model. In particular, we show that the Bayesian SVM, max-margin rank-likelihood and nonlinear decision function all improve the overall performance on their own, when compared to standard approaches such as probit regression and Gaussian models on log-transformed data. It is important to take into consideration that our model is at the same time trying to explain the data and to build a classifier via a linear latent representation of ranks, thus we will not attempt to match results obtained by more sophisticated state-of-the-art classification models (e.g., a nonlinear SVM applied directly to raw data may yield a good classifier, but it does not afford the generative interpretability of a factor model, the latter particularly relevant to biological applications). Our model is ultimately trying to find a good *balance* between covari-

Table 2. Mean error rates (%) and runtime in seconds for the test data of the USPS 3 vs. 5 subtask.

	G-L-Probit	G-L-BSVM	OR-L-Probit	OR-L-BSVM	R-L-BSVM	G-NL-BSVM	R-NL-BSVM
Error	5.95±0.005	5.86±0.008	5.05±0.013	4.92±0.027	**4.53±0.026**	3.88±0.017	**3.23±0.025**
Runtime	8.64	10.29	14.07	14.19	16.05	23.81	36.63

Table 3. Mean error rates (%) and runtime in seconds for the test data of the MNIST 3 vs. 5 subtask.

	G-L-BSVM	R-L-BSVM	L-SVM	G-NL-BSVM	R-NL-BSVM	NL-SVM
Error	5.05±0.053	4.84±0.014	**4.68**	4.21±0.010	2.10±0.007	**2.00**
Runtime	150	220	140	400	600	304

ance structure modeling, interpretability through shrinkage and classification performance. All of this is done with the very important additional benefit of not requiring distributional assumptions about the values of the data, as this information is usually not known in practice (as in the subsequent biological experiments below). However, in those cases where the distribution is known *a priori* it should certainly be reflected in likelihood function.

4.1. Handwritten digits

Digitized images are a good example of essentially non-Gaussian data traditionally modeled using Gaussian noise models in the context of factor models and dictionary learning (Mairal et al., 2008). However, depending on preprocessing, they can be naturally treated either as continuous variables representing pixel intensities when filtering/smoothing is pre-applied, or as discrete variables representing pixel values when raw data is available. Our running hypothesis here is that a rank-likelihood representation for pixels is more expressive than its Gaussian counterpart. Intuitively, a discriminative factor model should be able to find features (subsets of representative pixels) that separate image subtypes. However, without the Gaussian assumption for observations, our rank model might be able to adapt to more general conditions, e.g., skewed or heavy-tailed distributions. In our results we use classification error on a test set as a quantitative measure of performance.

USPS First we consider the models in Table 1 to the well known 3 vs. 5 subtask of the USPS handwritten digits dataset. It consists of 1540 smoothed gray scale 16×16 images rescaled to fit within the $[-1, 1]$ interval. Each observation is a 256-dimensional vector of scaled pixel intensities. Here we use the resampled version, which is 767 images for model fitting and the remaining 773 for testing. Results in Table 2 show that consistently: rank-likelihood based models outperform Gaussian models, BSVM outperforms the probit link, and nonlinear outperforms linear classifiers. Furthermore, the proposed max-margin rank likelihood model performs best in both variants, linear and nonlinear. In every case inference took less than 1 minute.

MNIST Next we consider the same 3 vs. 5 task, this time on a larger dataset, the MNIST database. The dataset is composed by 11552 training images and 1902 test images. Unlike USPS, MNIST consists of 28×28 raw 8-bit encoded images, so each observation is a 784-dimensional vector of pixel values (discrete values from $\{0, \ldots, 255\}$). Results for four out of six methods from Table 1 are summarized in Table 3. Results for probit based models were not as good as those for BSVM, thus not showed here, nor in the upcoming experiments. Instead, we include results for a linear (L-SVM) and nonlinear (NL-SVM) SVM with RBF kernel directly applied to the data as baselines, without the factor model. From Table 3 we see that the proposed model works better than the Gaussian model, and that the results for R-NL-BSVM are close to that for NL-SVM. This is not surprising, as a pure classification model (e.g., NL-SVM) does not attempt to explain the data but only to maximize classification performance. In this case, the most expensive approach, namely R-NL-BSVM has a runtime in the neighborhood of 10 minutes which is deemed acceptable considering the size of the dataset. Visualizations of the factor loadings, $\mathbf{A}$, learned by various models are presented in the Supplementary Material.

4.2. Gene expression

We applied our model to a newly published tuberculosis study from Anderson *et. al.* (2014), consisting of gene expression intensities for 47323 genes and 334 subjects (GEO accession series GSE39941). These subjects can be partitioned in three phenotypes: active tuberculosis (TB) (111), latent TB (54) and other infectious diseases (169), and in whether they are positive (107) or negative (227) for HIV. The raw data were preprocessed with background correction, sample-wise scaling and gene filtering. For the analysis we keep the top 4732 genes with largest intensity profiles. Results for three binary classification tasks using a *one vs. the rest* scheme, the HIV classifier and 10-fold cross-validation are summarized in Table 4 (error bars for accuracies omitted due to space limitations). We also present area under the ROC curve (AUC) to account for subset imbalance. As an additional baseline, we included a Poisson Factor Analysis (PFA) model (Zhou et al.,

Table 4. AUC (with error bars), accuracy and runtime in seconds for gene expression data.

Methods	PFA-L-BSVM	G-L-BSVM	R-L-BSVM	G-NL-BSVM	R-NL-BSVM
TB vs. Others	0.740±0.102, 0.683	0.766±0.093, 0.704	0.814±0.052, 0.740	0.847±0.061, 0.778	**0.872±0.025, 0.781**
Active TB vs. Others	0.802±0.070, 0.775	0.857±0.050, 0.784	0.896±0.028, 0.832	0.921±0.034, 0.853	**0.948±0.021, 0.880**
Latent TB vs. Others	0.849±0.051, 0.802	0.907±0.037, 0.841	0.923±0.041, 0.868	0.934±0.029, 0.874	**0.954±0.025, 0.889**
HIV(+) vs. HIV(-)	0.850±0.056, 0.793	0.879±0.055, 0.844	0.900±0.055, 0.856	0.915±0.041, 0.850	**0.959±0.051, 0.901**
One fold time	130	141	180	330	450

2012) with Bayesian SVMs, as a 2-step procedure. For the Gaussian models we log-transform intensities, and for PFA we round them to the closest integer value (raw intensities become floating point values after background correction and scaling). We can see that our models outperform the others in each of the classification subtasks, and R-NL-BSVM performs the best overall with a reasonable computational cost. It is important to mention that we are not building separate discriminative factor models for each subtask, instead a single factor model jointly learns the four predictors, meaning that all classifiers share the same loadings and factor scores. As a result, our model operates here as a *multi-tasking learning* scheme. Figure 3

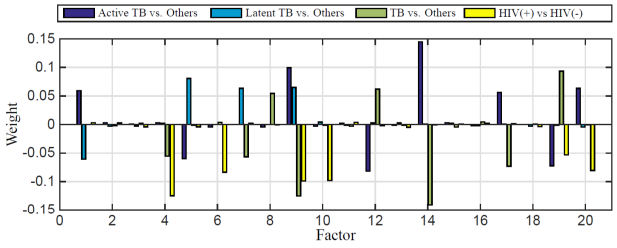

Figure 3. The learned coefficients β for the 4 classifiers based on gene expression data.

shows the coefficients of the classifier learned using R-L-BSVM. The leading coefficients reveal that certain factors are key to different classes. For instance, factor 14 is specific to TB vs. others, factors 1 and 5 are specific to TB vs. latent TB, and factors 9 and 4 are specific to TB vs. others including HIV(+). We performed a pathway association analysis using DAVID (Huang et al., 2009) on the top 200 genes from each factor. We found interesting associations. Factor 14: ubiquitin-protein, ligase activity and immunodeficiency. Factor 9: immune response, lymphocite/leukocite/T cell activation and apoptosis. Factor 5: proteasome complex, response to stress, response to antibiotic. Factor 4: ribonucleoprotein, proteasome, ubiquitin-protein, ligase activity. The complete gene lists and the inferred gene networks are provided in the Supplementary Material.

4.3. RNA sequencing

Finally, we consider a new RNA sequencing (RNAseq) sepsis study (**?**). The dataset consists of 133 subjects and 15158 genes (after removing genes with more than 25% zero entries). Data preprocessing consists of sample-wise rescaling to compensate for differences in library size, log-transform for Gaussian models and rounding for PFA. Subjects are split into three different groups, namely systemic inflammatory response (SIRS) (26), sepsis survivors (SeS) (78) and sepsis complications leading to death (SeD) (29). Three binary comparisons are the main interest of the study: SIRS vs (all) sepsis, SeS vs. SeD and SIRS vs. SeS. Being able to classify this sub-groupings is important for two reasons: (i) these tasks are known to be hard classification problems, and (ii) a recently published study by Liu et al. (2014) showed that approximately 40% of hospital mortality is sepsis related. Classification results for 10-fold cross-validation including AUC, accuracy and runtime per fold are summarized in Table 5. Once again our model performs the best. We also tried the nonlinear version of our model but figures were omitted due to very minor improvements in performance.

Table 5. AUC, accuracy and runtime(s) for RNAseq data.

Methods	PFA-L-BSVM	G-L-BSVM	R-L-BSVM
SIRS vs. Se	0.70±0.04, 0.73	0.78±0.02, 0.76	**0.86±0.01, 0.81**
SeD vs. SeS	0.76±0.05, 0.70	0.76±0.01, 0.75	**0.82±0.02, 0.78**
SIRS vs. SeS	0.75±0.02, 0.71	0.87±0.01, 0.71	**0.91±0.01, 0.87**
One fold time	179	175	226

5. Conclusion

We have developed a Bayesian discriminative factor model for data that are generally non-Gaussian. This is achieved via the integration of a new max-margin rank likelihood, Bayesian support vector machines, global-local shrinkage priors, and a Dirichlet process mixture model. The proposed model is built on the ranks of the data, opening the door to treat ordinal, continuous and discrete data (e.g., count data) within the same framework. Experiments have demonstrated that the proposed factor model achieves better performance than widely used log-transformed-plus-Gaussian models and a Poisson model, on both gene expression and RNA sequencing data. These results highlight the potential of the proposed model in a variety of applications, especially computational biology.

Our rank based models are relatively more computationally expensive than Gaussian models on log-transformed data. However, in applications such as gene expression or sequencing that constitute the real data used in our experiments, runtimes are still significantly lower when compared to the time needed to generate the data. For biological studies, the quality and interpretability of the results are of paramount importance, with speed a secondary issue.

Acknowledgments

The research reported here was funded in part by ARO, DARPA, DOE, NGA and ONR.

References

Anderson *et. al.*, S. T. Diagnosis of childhood tuberculosis and host RNA expression in Africa. *The New England Journal of Medicine*, 370(18):1712–1723, 2014.

Armagan, A., Clyde, M., and Dunson, D. B. Generalized beta mixtures of gaussians. In *NIPS 24*, 2011.

Bhattacharya, A. and Dunson, D. B. Sparse Bayesian infinite factor models. *Biometrika*, 98(2):291–306, 2011.

Blei, D. M. and Jordan, M. I. Variational inference for dirichlet process mixtures. *Bayesian Analysis*, 1:121–144, 2005.

Carvalho, C. M., Polson, N. G., and Scott, J. G. The horseshoe estimator for sparse signals. *Biometrika*, 97(2): 465–480, 2010.

Chib, S., Greenberg, E., and Winkelmann, R. Posterior simulation and Bayes factors in panel count data models. *Journal of Econometrics*, 86:33–54, 1998.

Dillies, M.-A. et al. A comprehensive evaluation of normalization methods for illumina high-throughput rna sequencing data analysis. *Briefings in bioinformatics*, 14 (6):671–683, 2013.

Fu, Z., Robles-Kelly, A., and Zhou, J. Mixing linear SVMs for nonlinear classification. *IEEE-TNN*, 2010.

Henao, R., Yuan, X., and Carin, L. Bayesian nonlinear support vector machines and discriminative factor modeling. In *NIPS*, 2014.

Hoff, P. D. Extending the rank likelihood for semiparametric copula estimation. *The Annals of Applied StatisticsS*, 1(1):265–283, 2007.

Hoff, P. D. *A First Course in Bayesian Statistical Methods*. Springer, 2009.

Huang, D. W., Sherman, B. T., and Lempicki, R.A. Systematic and integrative analysis of large gene lists using david bioinformatics resources. *Nature Protoc.*, 1:44–57, 2009.

Ishwaran, H. and James, L. F. Gibbs sampling methods for stick-breaking priors. *JASA*, 96(453):161–173, 2001.

Lehmann, E. L. and D'Abrera, H. J. M. *Nonparametrics: statistical methods based on ranks*. Springer New York, 2006.

Liu, V., Escobar, G J., and Greene *et al.* , J. D. Hospital deaths in patients with sepsis from 2 independent cohorts. *Journal of the American Medical Association*, 2014.

MacKay, D. J. C. Probable networks and plausible predictions – A review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6(3):469–505, 1995.

Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A. Supervised dictionary learning. In *NIPS 21*, 2008.

Monahan, J. F. and Boos, D. D. Proper likelihoods for Bayesian analysis. *Biometrika*, 79(2):271–278, 1992.

Murray, J. S., Dunson, D. B., Carin, L., and Lucas, J. E. Bayesian Gaussian copula factor models for mixed data. *JASA*, 108(502):656–665, 2013.

Neal, R. M. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

Pettitt, A. N. Inference for the linear model using a likelihood based on ranks. *JRSS-B*, 44(2):234–243, 1982.

Polson, N. G. and Scott, J. G. Shrink globally, act locally: sparse Bayesian regularization and prediction. *Bayesian Statistics*, 9:501–538, 2010.

Polson, N. G. and Scott, S. L. Rejoinder: Data augmentation for support vector machines. *Bayesian Analysis*, 6 (1):43–48, 2011a.

Polson, N. G. and Scott, S. L. Data augmentation for support vector machines. *Bayesian Analysis*, 6(1):1–23, 2011b.

Quadrianto, N., Sharmanska, V., Knowles, D. A., and Ghahramani, Z. The supervised IBP: Neighbourhood preserving infinite latent feature models. *CoRR*, 2013.

Salazar, E., Cain, M. S., Mitroff, S. R., and Carin, L. Inferring latent structure from mixed real and categorical relational data. In *ICML*, 2012.

Scholkopf, B. and Smola, A. J. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

Sethuraman, J. A constructive definition of the Dirichlet prior. *Statistica Sinica*, 4:639–650, 2001.

Shahbaba, B. and Neal, R. M. Nonlinear models using Dirichlet process mixtures. *JMLR*, 10:1829–1850, 2009.

Wipf, D. and Nagarajan, S. A new view of automatic relevance determination. In *NIPS 21*, 2008.

Xu, M., Zhu, J., and Zhang, B. Fast max-margin matrix factorization with data augmentation. In *ICML*, 2013.

Zhou, M., Hannah, L., Dunson, D., and Carin, L. Beta-negative binomial process and Poisson factor analysis. In *AISTATS*, 2012.

6. Model

The full Bayesian model is:

$$
\begin{aligned}
x_{i,n} &= g_i(w_{i,n}), \quad i = 1, \dots, d; \;\; n = 1, \dots, N; \\
L_i(w_{i,n}|\mathbf{w}_{i\backslash n}) &= \int \mathcal{N}(w_{i,n} - w^u_{i,n}; -\epsilon - \lambda^u_{i,n}, \lambda^u_{i,n}) \\
&\quad \times \mathcal{N}(w^l_{i,n} - w_{i,n}; -\epsilon - \lambda^l_{i,n}, \lambda^l_{i,n}) d\lambda^u_{i,n} d\lambda^l_{i,n}\,, \\
L_n(y_n|\boldsymbol{\beta}, \mathbf{z}_n) &= \int_0^\infty \frac{1}{\sqrt{2\pi\lambda^c_n}} \\
&\quad \times \exp\left(-\frac{1}{2}\frac{(1 - y_n\boldsymbol{\beta}^\top\mathbf{z}_n + \lambda^c_n)^2}{\lambda^c_n}\right) d\lambda^c_n\,, \\
\mathbf{z}_n &\sim \mathcal{N}(\boldsymbol{\mu}_{t(n)}, \psi^{-1}_{t(n)}\mathbf{I}_K)\,, \\
\boldsymbol{\mu}_t &\sim (\mathbf{0}, \mathbf{I}_K), \\
\psi_t &\sim \mathrm{Ga}(\psi_s, \psi_r), \\
t(n) &\sim \mathrm{Mult}(1; q_1, \dots, q_T), \\
q_t &= \nu_t \prod_{l=1}^{t-1}(1 - \nu_l)\,, \\
\nu_t &\sim \mathrm{Beta}(1, \alpha), \quad \alpha \sim \mathrm{Ga}(\alpha_s, \alpha_r), \\
a_{i,k} &\sim \mathcal{N}(0, \xi_{i,k})\,, \quad \xi_{i,k} \sim \mathrm{Ga}(r_a, \eta_{i,k})\,, \\
\eta_{i,k} &\sim \mathrm{Ga}(s_a, \Phi^{(a)}_k)\,, \\
\Phi^{(a)}_k &\sim \mathrm{Ga}(1/2, \tilde{\Phi}^{(a)}), \quad \tilde{\Phi}^{(a)} \sim \mathrm{Ga}(1/2, 1), \\
\beta_k &\sim \mathcal{N}(0, b_k)\,, \qquad b_k \sim \mathrm{Ga}(r_\beta, e_k)\,, \\
e_k &\sim \mathrm{Ga}(s_\beta, \Phi^{(\beta)})\,, \\
\Phi^{(\beta)} &\sim \mathrm{Ga}(1/2, \tilde{\Phi}^{(\beta)}), \quad \tilde{\Phi}^{(\beta)} \sim \mathrm{Ga}(1/2, 1).
\end{aligned}
$$

7. MCMC inference

For convenience, we denote

$$
\begin{aligned}
\lambda_{i,n} &= (\lambda^l_{i,n})^{-1} + (\lambda^u_{i,n})^{-1}, \\
\Delta^{(k)}_{i,n} &= \left(\frac{w^l_{i,n} + \epsilon - w_{i,n}}{\lambda^l_{i,n}} - \frac{w_{i,n} + \epsilon - w^u_{i,n}}{\lambda^u_{i,n}}\right) \\
&\quad + a_{i,k} z_{k,n} \lambda_{i,n},
\end{aligned}
$$

$$
\begin{aligned}
(\boldsymbol{\beta}^\top\mathbf{z}_n)_{\backslash k} &= \boldsymbol{\beta}^\top\mathbf{z}_n - \beta_k z_{k,n}, \\
\Gamma_{k,n} &= \frac{y_n\beta_k[1 + \lambda^c_n - y_n(\boldsymbol{\beta}^\top\mathbf{z}_n)_{\backslash k}]}{\lambda^c_n}.
\end{aligned}
$$

In the following conditional posterior-distributions, "$\cdot$" refers to the conditioning parameters of the distributions, $\mathrm{IG}(a, b)$ denotes the inverse Gaussian distribution, $\mathrm{Ga}(a, b)$ the gamma distribution, and $\mathrm{GIG}(a, b, p)$ the generalized inverse Gaussian distribution.

In the *linear* case, when the Dirichlet process mixture (DPM) model is not used, the conditional posterior distributions are:

1. $\mathbf{A}$:

$$
\begin{aligned}
p(a_{i,k}|\,\cdot) &= \mathcal{N}(\mu_{a_{i,k}}, \sigma^2_{a_{i,k}})\,, \\
\sigma^{-2}_{a_{i,k}} &= \xi^{-1}_{i,k} + \sum_{n=1}^N z^2_{k,n}\lambda^{-1}_{i,n}\,, \\
\mu_{a_{i,k}} &= \sigma^2_{a_{i,k}} \sum_{n=1}^N z_{k,n}\Delta^{(k)}_{i,n}\,, \\
p(\xi_{i,k}|\cdot) &= \mathrm{GIG}(2\eta_{i,k}, a^2_{i,k}, r_a - 0.5)\,, \\
p(\eta_{i,k}|\cdot) &= \mathrm{Ga}(r_a + s_a, \xi_{i,k} + \Phi^{(a)}_k)\,, \\
p(\Phi^{(a)}_k|\cdot) &= \mathrm{Ga}\left(\frac{1}{2} + s_a d, \tilde{\Phi}^{(a)} + \frac{1}{2}\sum_i \eta_{i,k}\right), \\
p(\tilde{\Phi}^{(a)}|\cdot) &= \mathrm{Ga}\left(1, \sum_k \Phi^{(a)}_k + 1\right).
\end{aligned}
$$

2. $\mathbf{Z}$:

$$
\begin{aligned}
p(z_{k,n}|\,\cdot) &= \mathcal{N}(\mu_{z_{k,n}}, \sigma^2_{z_{k,n}})\,, \\
\sigma^{-2}_{z_{k,n}} &= 1 + \sum_{i=1}^d a^2_{i,k}\lambda^{-1}_{i,n} + \frac{\beta^2_k}{\lambda^c_n}\,, \\
\mu_{z_{k,n}} &= \sigma^2_{z_{k,n}}\left(\sum_{i=1}^d a_{i,k}\Delta^{(k)}_{i,n} + \Gamma_{k,n}\right).
\end{aligned}
$$

3. $\boldsymbol{\Lambda}^l, \boldsymbol{\Lambda}^u, \boldsymbol{\lambda}^c$:

$$
\begin{aligned}
p\left((\lambda^l_{i,n})^{-1}|\cdot\right) &= \mathrm{IG}\left(|w^l_{i,n} + \epsilon - w_{i,n}|^{-1}, 1\right), \\
p\left((\lambda^u_{i,n})^{-1}|\cdot\right) &= \mathrm{IG}\left(|w_{i,n} + \epsilon - w^u_{i,n}|^{-1}, 1\right), \\
p\left((\lambda^c_n)^{-1}|\cdot\right) &= \mathrm{IG}\left(|1 - y_n\boldsymbol{\beta}^\top\mathbf{z}_n|^{-1}, 1\right).
\end{aligned}
$$

4. $\boldsymbol{\beta}$:

$$p(\beta_k|\cdot) = \mathcal{N}(\mu_{\beta_k}, \sigma^2_{\beta_k}), \quad \sigma^{-2}_{\beta_k} = b_k^{-1} + \sum_{n=1}^{N} \frac{z^2_{k,n}}{\lambda^c_n},$$
$$\mu_{\beta_k} = \sigma^2_{\beta_k} \sum_{n=1}^{N} \frac{y_n z_{k,n} \left[1 + \lambda^c_n - y_n(\boldsymbol{\beta}^\top \mathbf{z}_n)_{\setminus k}\right]}{\lambda^c_n},$$
$$p(b_k|\cdot) = \mathrm{GIG}(2e_k, \beta^2_k, r_\beta - 0.5),$$
$$p(e_k|\cdot) = \mathrm{Ga}(r_\beta + s_\beta, b_k + \Phi^{(\beta)}_k),$$
$$p(\Phi^{(\beta)}|\cdot) = \mathrm{Ga}\left(\frac{1}{2} + s_\beta K, \tilde{\Phi}^{(\beta)} + \frac{1}{2}\sum_k e_k\right),$$
$$p(\tilde{\Phi}^{(\beta)}|\cdot) = \mathrm{Ga}\left(1, \Phi^{(\beta)} + 1\right).$$

In the *nonlinear* case, when the DPM is used:

1. $t(n)$ (mixture component index for n-th observation):

$$p(t(n) = t|\cdot) \propto q_t \mathcal{N}(\mathbf{z}_n; \boldsymbol{\mu}_t, \psi_t^{-1}\mathbf{I}_K).$$

2. DPM parameters:

$$p(\mu_{t,k}|\cdot) = \mathcal{N}(\mu_{\mu_{t,k}}, \sigma^2_{\mu_{t,k}}), \quad \sigma^{-2}_{\mu_{t,k}} = 1 + \sum_{n:t(n)=t} \psi_t,$$
$$\mu_{\mu_{t,k}} = \sigma^2_{\mu_{t,k}} \psi_t \sum_{n:t(n)=t} z_{k,n},$$
$$p(\psi_t|\cdot) = \mathrm{Ga}\left(\psi_s + 0.5K, \;\; \psi_r + 0.5\sum_k \mu^2_{t,k}\right),$$
$$p(\nu_t|\cdot) = \mathrm{Beta}\left(1 + \sum_{n:t(n)=t} 1, \;\; \alpha + \sum_{n:t(n)>t} 1\right),$$
$$p(\alpha|\cdot) = \mathrm{Ga}\left(\alpha_s + T - 1, \;\; \alpha_r - \sum_{t=1}^{T-1} \log(1 - \nu_t)\right).$$

In this case, $\boldsymbol{\beta}$ in 2) and 4) should be replaced by $\boldsymbol{\beta}^{(t)}$, for $t = 1, \ldots, T$, and the summation over n in 4) will only account for $\{n : t(n) = t\}$.

8. VB inference

Since the model is fully local conjugate, the VB update equations can be obtained using the moments of the above conditional posterior distributions. Here we present the moments for the model without DPM, and for the VB inference of the DP mixture model, please refer to (Blei & Jordan, 2005). In the following expressions, $\langle\cdot\rangle$ denotes expectation, $\mathcal{K}_p(\cdot)$ is the modified Bessel function of the second kind, $\langle w_{i,n}\rangle = \langle \mathbf{a}_i^\top\rangle\langle\mathbf{z}_n\rangle$, $\langle w^l_{i,n}\rangle = \langle \mathbf{a}_i^\top\rangle\langle\mathbf{z}^l_n\rangle$ and $\langle w^u_{i,n}\rangle = \langle \mathbf{a}_i^\top\rangle\langle\mathbf{z}^u_n\rangle$.

1. $\mathbf{A}$:

$$\langle a_{i,k}\rangle = \langle\sigma^2_{a_{i,k}}\rangle \sum_{n=1}^{N} \langle z_{k,n}\rangle\langle\Delta^{(k)}_{i,n}\rangle,$$
$$\langle\sigma^2_{a_{i,k}}\rangle = \left(\langle\xi^{-1}_{i,k}\rangle + \sum_{n=1}^{N}\langle z^2_{k,n}\rangle\langle\lambda^{-1}_{i,n}\rangle\right)^{-1},$$
$$\langle a^2_{i,k}\rangle = \langle a_{i,k}\rangle^2 + \langle\sigma^2_{a_{i,k}}\rangle,$$
$$\langle\Delta^{(k)}_{i,n}\rangle = \langle(\lambda^l_{i,n})^{-1}\rangle\left(\langle w^l_{i,n}\rangle + \epsilon - \langle w_{i,n}\rangle\right) - \langle(\lambda^u_{i,n})^{-1}\rangle\left(\langle w_{i,n}\rangle + \epsilon - \langle w^u_{i,n}\rangle\right) + \langle a_{i,k}\rangle\langle z_{k,n}\rangle\left[\langle(\lambda^l_{i,n})^{-1}\rangle + \langle(\lambda^u_{i,n})^{-1}\rangle\right],$$
$$\langle\xi_{i,k}\rangle = \frac{\sqrt{\langle a^2_{i,k}\rangle}\mathcal{K}_{r_a+0.5}\left(\sqrt{2\langle\eta_{i,k}\rangle\langle a^2_{i,k}\rangle}\right)}{\sqrt{2\eta_{i,k}}\mathcal{K}_{r_a-0.5}\left(\sqrt{2\langle\eta_{i,k}\rangle\langle a^2_{i,k}\rangle}\right)},$$
$$\langle\xi^{-1}_{i,k}\rangle = \frac{\sqrt{2\langle\eta_{i,k}\rangle}\mathcal{K}_{r_a-0.5}\left(\sqrt{2\langle\eta_{i,k}\rangle\langle a^2_{i,k}\rangle}\right)}{\sqrt{\langle a^2_{i,k}\rangle}\mathcal{K}_{r_a-1.5}\left(\sqrt{2\langle\eta_{i,k}\rangle\rangle a^2_{i,k}\rangle}\right)},$$
$$\langle\eta_{i,k}\rangle = \frac{r_a + s_a}{\langle\xi_{i,k}\rangle + \langle\Phi^{(a)}_k\rangle},$$
$$\langle\Phi^{(a)}_k\rangle = \frac{0.5 + ds_a}{\langle\tilde{\Phi}^{(a)}\rangle + 0.5\sum_i\langle\eta_{i,k}\rangle},$$
$$\langle\tilde{\Phi}^{(a)}\rangle = \frac{1}{1 + \sum_k\langle\Phi^{(a)}_k\rangle}.$$

2. $\mathbf{Z}$:

$$\langle z_{k,n}\rangle = \langle\sigma^2_{z_{k,n}}\rangle\left(\sum_{i=1}^{d}\langle a_{i,k}\rangle\langle\Delta^{(k)}_{i,n}\rangle + \langle\Gamma_{k,n}\rangle\right),$$
$$\langle\Gamma_{k,n}\rangle = \langle(\lambda^c_n)^{-1}\rangle\left\{y_n\langle\beta_k\rangle[\langle(\lambda^c_n)^{-1}\rangle + 1 - \langle(\lambda^c_n)^{-1}\rangle y_n(\langle\boldsymbol{\beta}^\top\rangle\langle\mathbf{z}_n\rangle)_{\setminus k}]\right\},$$
$$\langle\sigma^2_{z_{k,n}}\rangle = \left(1 + \sum_{i=1}^{d}\langle a^2_{i,k}\rangle\langle\lambda^{-1}_{i,n}\rangle + \langle\beta^2_k\rangle\langle(\lambda^c_n)^{-1}\rangle\right)^{-1},$$
$$\langle z^2_{k,n}\rangle = \langle z_{k,n}\rangle^2 + \langle\sigma^2_{z_{k,n}}\rangle.$$

3. $\boldsymbol{\Lambda}^l, \boldsymbol{\Lambda}^u, \boldsymbol{\lambda}^c$:

$$\langle(\lambda^l_{i,n})^{-1}\rangle = \left|\langle w^l_{i,n}\rangle + \epsilon - \langle w_{i,n}\rangle\right|^{-1},$$
$$\langle(\lambda^u_{i,n})^{-1}\rangle = \left|\langle w_{i,n}\rangle + \epsilon - \langle w^u_{i,n}\rangle\right|^{-1},$$
$$\langle(\lambda^c_n)^{-1}\rangle = \left|1 - y_n\langle\boldsymbol{\beta}^\top\rangle\langle\mathbf{z}_n\rangle\right|^{-1}.$$

4. $\boldsymbol{\beta}$:

$$\begin{aligned}
\langle \beta_k \rangle &= \langle \sigma^2_{\beta_k} \rangle \sum_{n=1}^{N} \Big\{ y_n \langle z_{k,n} \rangle [\langle (\lambda^c_n)^{-1} \rangle + 1 \\
&\quad - \langle (\lambda^c_n)^{-1} \rangle y_n (\langle \boldsymbol{\beta}^\top \rangle \langle \mathbf{z}_n \rangle)_{\setminus k}] \Big\}, \\
\langle \sigma^2_{\beta_k} \rangle &= \langle b_k^{-1} \rangle + \sum_{n=1}^{N} \langle z^2_{k,n} \rangle \langle (\lambda^c_n)^{-1} \rangle, \\
\langle \beta_k^2 \rangle &= \langle \beta_k \rangle^2 + \langle \sigma^2_{\beta_k} \rangle, \\
\langle b_k \rangle &= \frac{\sqrt{\langle \beta_k^2 \rangle} \mathcal{K}_{r_\beta + 0.5} \left(\sqrt{2 \langle e_k \rangle \langle \beta_k^2 \rangle} \right)}{\sqrt{2 e_k} \mathcal{K}_{r_\beta - 0.5} \left(\sqrt{2 \langle e_k \rangle \langle \beta_k^2 \rangle} \right)}, \\
\langle b_k^{-1} \rangle &= \frac{\sqrt{2 e_k} \mathcal{K}_{r_\beta - 0.5} \left(\sqrt{2 \langle e_k \rangle \langle \beta_k^2 \rangle} \right)}{\sqrt{\langle \beta_k^2 \rangle} \mathcal{K}_{r_\beta - 1.5} \left(\sqrt{2 \langle e_k \rangle \langle \beta_k^2 \rangle} \right)}, \\
\langle e_k \rangle &= \frac{r_\beta + s_\beta}{\langle b_k \rangle + \langle \Phi_k^{(\beta)} \rangle}, \\
\langle \Phi^{(\beta)} \rangle &= \frac{0.5 + 0.5 s_\beta}{\langle \tilde{\Phi}^{(\beta)} \rangle + 0.5 \sum_k \langle e_k \rangle}, \\
\langle \tilde{\Phi}^{(\beta)} \rangle &= \frac{1}{\langle \Phi^{(\beta)} \rangle + 1}.
\end{aligned}$$

9. Inferred Factor Loadings on the Handwritten Digits

We plotted the factor loadings $\mathbf{A}$ learned from USPS and MNIST datasets in Figures 5 and 4, respectively. Four models, G-L-BSVM, R-L-BSVM, G-NL-BSVM and R-NL-BSVM are used as examples. It can be be seen that the Gaussian model is trying to learn the dictionaries to reconstruct images while the rank model is trying to learning the differences (focusing on the edges).

10. Results on Gene Expression Data

We show the results of our model for gene expression data. $K = 20$ factors are used and here we only show the results generated by the proposed max-margin rank model without DP, *i.e.*, using linear Bayesian SVM as the classifier. Figure 6 shows the coefficients $\boldsymbol{\beta}$ of the learned classifiers and Figure 7 the inferred gene network from the learned factor loading matrix $\mathbf{A}$.

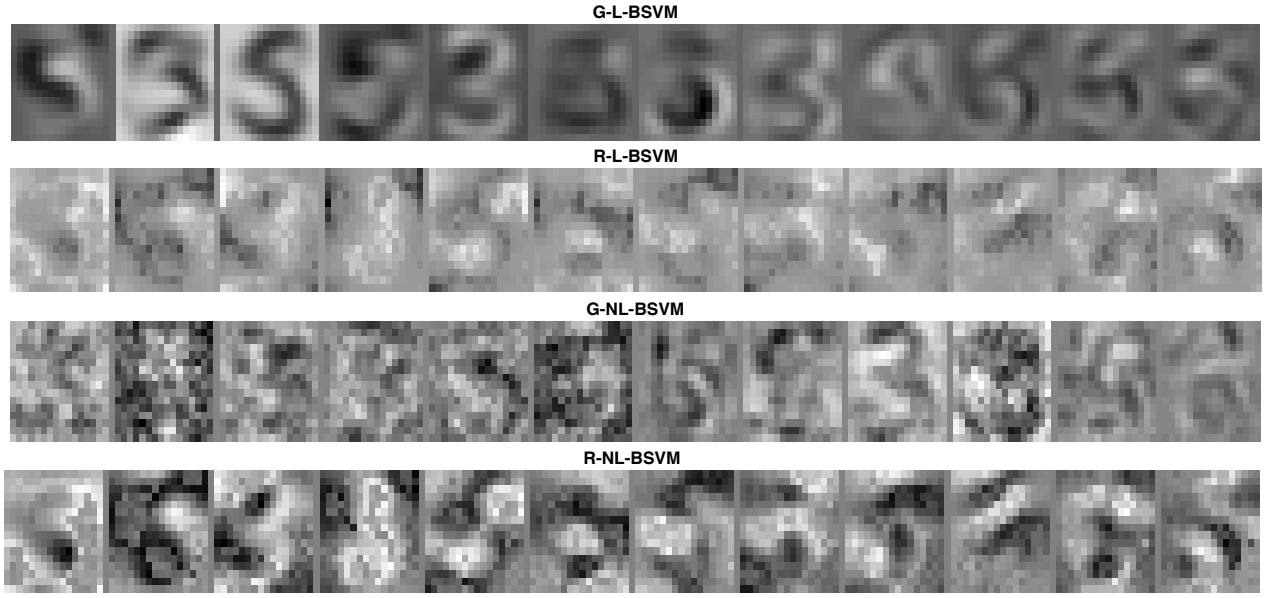

Figure 4. Inferred factor loading matrix $\mathbf{A}$ from USPS 3 vs. 5. The first 12 columns are reshaped and plotted.

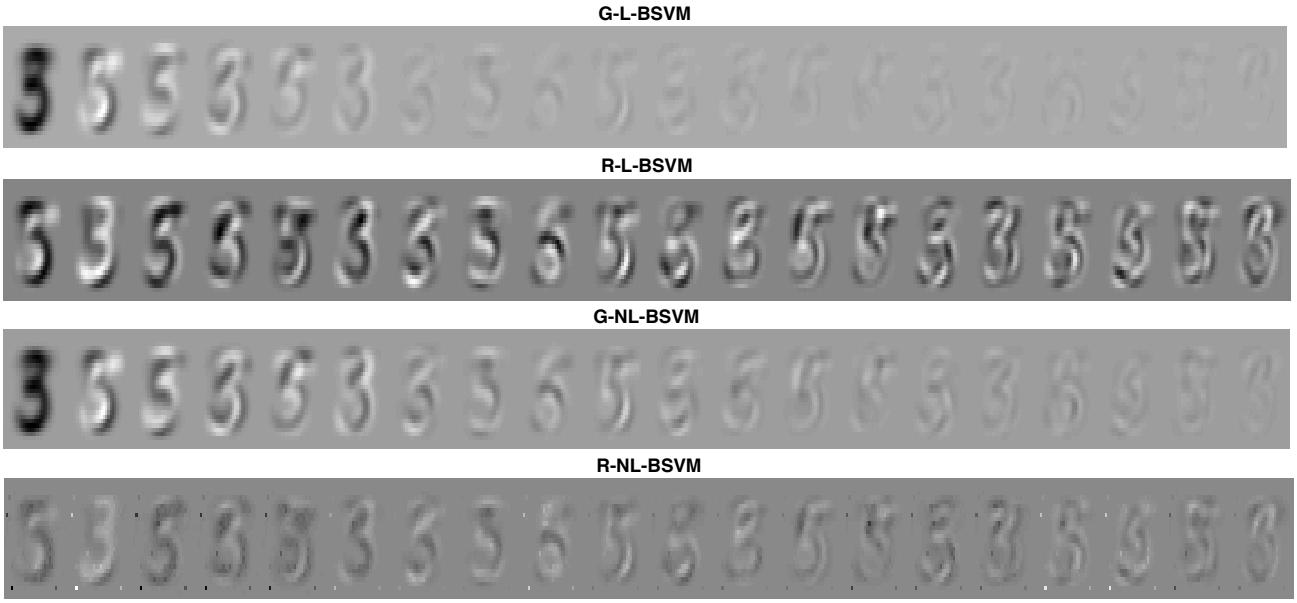

Figure 5. Inferred factor loading matrix $\mathbf{A}$ from MNIST 3 vs. 5. The 20 columns are reshaped and plotted.

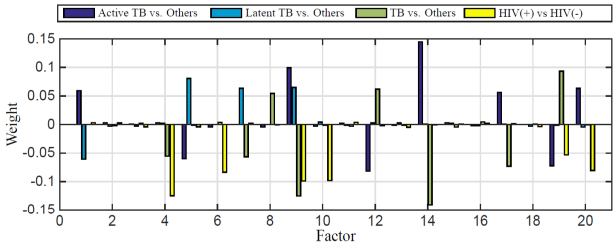

Figure 6. The learned classifier coefficients β of the 4 classifiers for the gene expression data.

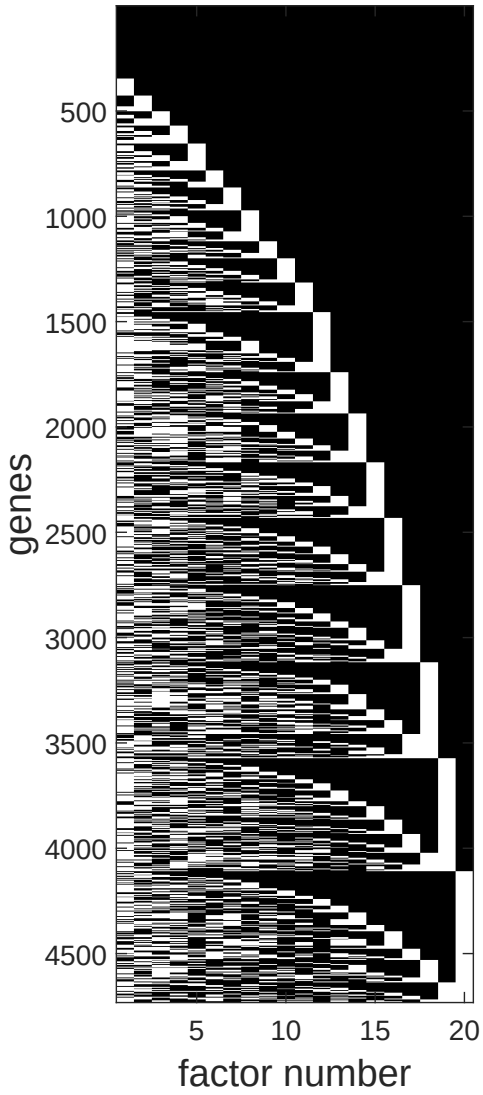

Figure 7. The learned gene network inferred from the factor loading matrix $\mathbf{A}$.