Supplementary Materials

A MARGINAL LIKELIHOOD IN BAYESIAN LINEAR REGRESSION

For ease of notation, we drop the j subscript, and therefore $\mathbf{Y} \to \mathbf{y} = [y_1 \dots y_N]^\top$ and $\boldsymbol{\beta}_j \to \boldsymbol{\beta}$. Consider the linear regression model

$$\mathbf{y} = \varphi_{\mathbf{W}}(\mathbf{X})\boldsymbol{\beta}^\top + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}_N(\mathbf{0}, \sigma^2\mathbf{I}), \tag{18}$$

where $\varphi_{\mathbf{W}}(\mathbf{X}) = [\varphi(\mathbf{x}_1, \mathbf{w}) \dots \varphi(\mathbf{x}_N, \mathbf{w})]^\top$, an $N \times M$ matrix. A common conjugate prior on $\boldsymbol{\beta}$ is a normal–inverse–gamma distribution,

$$\begin{aligned} \boldsymbol{\beta} \mid \sigma^2 &\sim \mathcal{N}_M(\boldsymbol{\beta}_0, \sigma^2\mathbf{S}_0^{-1}) \\ \sigma^2 &\sim \text{InvGamma}(a_0, b_0), \end{aligned} \tag{19}$$

We can write the functional form of the posterior and prior terms in (19) as

$$\begin{aligned} p(\mathbf{y} \mid \varphi_{\mathbf{W}}(\mathbf{X}), \boldsymbol{\beta}, \sigma^2) &= (2\pi\sigma^2)^{-N/2} \exp\Big(-\frac{1}{2\sigma^2}(\mathbf{y} - \varphi_{\mathbf{W}}(\mathbf{X})\boldsymbol{\beta})^\top(\mathbf{y} - \varphi_{\mathbf{W}}(\mathbf{X})\boldsymbol{\beta})\Big) \\ p(\boldsymbol{\beta} \mid \sigma^2) &= (2\pi\sigma^2)^{-M/2}\big|\mathbf{S}_0\big|^{1/2} \exp\Big(-\frac{1}{2\sigma^2}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)^\top\mathbf{S}_0(\boldsymbol{\beta} - \boldsymbol{\beta}_0)\Big) \\ p(\sigma_\beta^2) &= \frac{b_0^{a_0}}{\Gamma(a_0)}(\sigma_\beta^2)^{-(a_0+1)} \exp\Big(\frac{-b_0}{\sigma^2}\Big). \end{aligned} \tag{20}$$

We can combine the likelihood's Gaussian kernel with the prior's kernel in the following way:

$$\begin{aligned} &(\mathbf{y} - \varphi_{\mathbf{W}}(\mathbf{X})\boldsymbol{\beta})^\top(\mathbf{y} - \varphi_{\mathbf{W}}(\mathbf{X})\boldsymbol{\beta}) + (\boldsymbol{\beta} - \boldsymbol{\beta}_0)^\top\mathbf{S}_0(\boldsymbol{\beta} - \boldsymbol{\beta}_0) \\ &= \mathbf{y}^\top\mathbf{y} + \boldsymbol{\beta}_0^\top\mathbf{S}_0\boldsymbol{\beta}_0 - \boldsymbol{\beta}_N^\top\mathbf{S}_N\boldsymbol{\beta}_N + (\boldsymbol{\beta} - \boldsymbol{\beta}_N)^\top\mathbf{S}_N(\boldsymbol{\beta} - \boldsymbol{\beta}_N). \end{aligned} \tag{21}$$

where $\boldsymbol{\beta}_N$ and $\mathbf{S}_N$ are defined as

$$\begin{aligned} \mathbf{S}_N &= \varphi_{\mathbf{W}}(\mathbf{X})^\top\varphi_{\mathbf{W}}(\mathbf{X}) + \mathbf{S}_0 \\ \boldsymbol{\beta}_N &= \mathbf{S}_N^{-1}(\boldsymbol{\beta}_0^\top\mathbf{S}_0 + \varphi_{\mathbf{W}}(\mathbf{X})^\top\mathbf{y}). \end{aligned} \tag{22}$$

Now our posterior can be written as

$$\begin{aligned} p(\mathbf{y} \mid \varphi_{\mathbf{W}}(\mathbf{X}), \boldsymbol{\beta}, \sigma^2) \propto (2\pi)^{-M/2}\big|\mathbf{S}_0\big|^{1/2} &\exp\Big(-\frac{1}{2\sigma^2}\Big[(\boldsymbol{\beta} - \boldsymbol{\beta}_N)^\top\mathbf{S}_N(\boldsymbol{\beta} - \boldsymbol{\beta}_N)\Big]\Big) \\ (2\pi\sigma^2)^{-N/2} &\exp\Big(-\frac{1}{2\sigma^2}\Big[\mathbf{y}^\top\mathbf{y} + \boldsymbol{\beta}_0^\top\mathbf{S}_0\boldsymbol{\beta}_0 - \boldsymbol{\beta}_N^\top\mathbf{S}_N\boldsymbol{\beta}_N\Big]\Big) \\ \frac{b_0^{a_0}}{\Gamma(a_0)}(\sigma^2)^{-(a_0+1)} &\exp\Big(\frac{-b_0}{\sigma^2}\Big). \end{aligned} \tag{23}$$

We can see that we have an M-variate normal distribution on the first line. If we ignore $(2\pi)^{-N/2}$ and inverse–gamma prior normalizer, we can combine the bottom two lines to be proportional to an inverse–gamma distribution,

$$(\sigma^2)^{-(a_0+N/2+1)} \exp\Big(-\frac{1}{\sigma^2}\Big[b_0 + \frac{1}{2}\Big\{\mathbf{y}^\top\mathbf{y} + \boldsymbol{\beta}_0^\top\mathbf{S}_0\boldsymbol{\beta}_0 - \boldsymbol{\beta}_N^\top\mathbf{S}_N\boldsymbol{\beta}_N\Big\}\Big]\Big). \tag{24}$$

Now define a_N and b_N as

$$\begin{aligned} a_N &= a_0 + \frac{N}{2} \\ b_N &= b_0 + \frac{1}{2}(\mathbf{y}^\top\mathbf{y} + \boldsymbol{\beta}_0^\top\mathbf{S}_0\boldsymbol{\beta}_0 - \boldsymbol{\beta}_N^\top\mathbf{S}_N\boldsymbol{\beta}_N). \end{aligned} \tag{25}$$

Thus, we can write our posterior as

$$\begin{aligned} p(\boldsymbol{\beta}, \sigma^2 \mid \varphi_{\mathbf{W}}(\mathbf{X}), \mathbf{y}) &\propto p(\boldsymbol{\beta} \mid \varphi_{\mathbf{W}}(\mathbf{X}), \mathbf{y}_j, \sigma^2) p(\sigma^2 \mid \varphi_{\mathbf{W}}(\mathbf{X}), \mathbf{y}) \\ &\text{where} \\ \boldsymbol{\beta} \mid \varphi_{\mathbf{W}}(\mathbf{X}), \mathbf{y}, \sigma^2 &\sim \mathcal{N}_M(\boldsymbol{\beta}_N, \mathbf{S}_N) \\ \sigma^2 \mid \mathbf{y}, \varphi_{\mathbf{W}}(\mathbf{X}) &\sim \text{InvGamma}(a_N, b_N). \end{aligned} \tag{26}$$

Now to compute the log marginal likelihood, we want

$$p(\mathbf{y} \mid \varphi_{\mathbf{W}}(\mathbf{X}), a_0, b_0) = \iint p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\beta}, \sigma^2) p(\boldsymbol{\beta}), \sigma^2 \mid a_0, b_0) \, \mathrm{d}^M \boldsymbol{\beta} \, \mathrm{d}\sigma^2. \tag{27}$$

Using the definitions in (22) and (25), we can write the joint as

$$\begin{aligned} p(\mathbf{y}, \boldsymbol{\beta}, \sigma^2) = (2\pi\sigma^2)^{-P/2} \big|\mathbf{S}_0\big|^{1/2} \exp\Big(- \frac{1}{2\sigma^2} \Big[(\boldsymbol{\beta} - \boldsymbol{\beta}_N)^\top \mathbf{S}_N (\boldsymbol{\beta} - \boldsymbol{\beta}_N) \Big] \Big) \\ (\sigma^2)^{-(a_N+1)} \exp\Big(- \frac{b_N}{\sigma^2} \Big) \\ (2\pi)^{-N/2} \frac{b_0^{a_0}}{\Gamma(a_0)}. \end{aligned} \tag{28}$$

The integral over $\boldsymbol{\beta}$ is only over the Gaussian kernel, which allows us to compute it immediately:

$$(2\pi\sigma^2)^{M/2} \big|\mathbf{S}_N\big|^{-1/2} = \int \exp\Big(- \frac{1}{2} (\boldsymbol{\beta} - \boldsymbol{\beta}_N)^\top \Big[\frac{1}{\sigma^2} \mathbf{S}_N \Big] (\boldsymbol{\beta} - \boldsymbol{\beta}_N) \Big) \mathrm{d}^M \boldsymbol{\beta}. \tag{29}$$

The terms $(2\pi\sigma^2)^{M/2}$ in (28) cancel, and the first line of (28) reduces to

$$\sqrt{\frac{|\mathbf{S}_0|}{|\mathbf{S}_N|}}. \tag{30}$$

We can compute the second integral in (27) because we know the normalizing constant of the gamma kernel,

$$\frac{\Gamma(a_N)}{b_N^{a_N}} = \int (\sigma^2)^{-(a_N+1)} \exp\Big(- \frac{b_N}{\sigma^2} \Big) \mathrm{d}\sigma^2. \tag{31}$$

Putting everything together, we see that the marginal likelihood is

$$p(\mathbf{y} \mid \varphi_{\mathbf{W}}(\mathbf{X}), a_0, b_0) = \frac{1}{(2\pi)^{N/2}} \cdot \sqrt{\frac{|\mathbf{S}_0|}{|\mathbf{S}_N|}} \cdot \frac{b_0^{a_0}}{b_N^{a_N}} \cdot \frac{\Gamma(a_N)}{\Gamma(a_0)}. \tag{32}$$

B NEGATIVE BINOMIAL GIBBS SAMPLER UPDATES

B.1 Sampling $\boldsymbol{\beta}_j$

Let ω be a Pólya-Gamma distributed random variable with parameters $b > 0$ and $c \in \mathbb{R}$, denoted $\omega \sim \text{PG}(b, c)$. Polson et al. (2013) proved two useful properties of Pólya-Gamma variables. First,

$$\frac{(e^{\psi})^a}{(1+e^{\psi})^b} = 2^{-b} e^{\kappa\psi} \int_0^\infty e^{-\omega\psi^2/2} p(\omega) \mathrm{d}\omega, \tag{33}$$

where $\kappa = a - b/2$ and $p(\omega) = \text{PG}(\omega \mid b, 0)$. And second,

$$p(\omega \mid \psi) \sim \text{PG}(b, \psi). \tag{34}$$

Now consider an NB likelihood on $\mathbf{Y}$,

$$p(\mathbf{Y} \mid —) = \prod_{n=1}^{N} \prod_{j=1}^{J} \frac{(\exp\{\boldsymbol{\beta}_j^\top \varphi_{\mathbf{W}}(\mathbf{x}_n)\})^{y_{nj}}}{(1+\exp\{\boldsymbol{\beta}_j^\top \varphi_{\mathbf{W}}(\mathbf{x}_n)\})^{y_{nj}+r_j}}. \tag{35}$$

Using (33), we can express the nj-th term in the negative binomial likelihood using the following variable substitutions,

$$\psi = \boldsymbol{\beta}_j^\top \varphi_{\mathbf{W}}(\mathbf{x}_n), \quad a = y_{nj}, \quad b = y_{nj} + r_j, \quad \kappa = \frac{y_{nj} - r_j}{2}. \tag{36}$$

This gives us

$$\begin{aligned} &\frac{(\exp\{\boldsymbol{\beta}_j^\top \varphi_{\mathbf{W}}(\mathbf{x}_n)\})^{y_{nj}}}{(1+\exp\{\boldsymbol{\beta}_j^\top \varphi_{\mathbf{W}}(\mathbf{x}_n)\})^{y_{nj}+r_j}} \\ &\propto \exp\Big\{\frac{y_{nj}-r_j}{2}\boldsymbol{\beta}_j^\top \varphi_{\mathbf{W}}\mathbf{x}_n)\Big\} \int_0^\infty \exp\Big\{-\omega_{nj}\frac{(\boldsymbol{\beta}_j^\top \varphi_{\mathbf{W}}(\mathbf{x}_n))^2}{2}\Big\} p(\omega_{nj})\mathrm{d}\omega_{nj} \\ &= \exp\Big\{-\frac{\omega_{nj}}{2}\Big(\boldsymbol{\beta}_j^\top \varphi_{\mathbf{W}}(\mathbf{x}_n) - z_{nj}\Big)^2\Big\} \end{aligned} \tag{37}$$

where

$$z_{nj} = \frac{y_{nj} - r_j}{2\omega_{nj}}. \tag{38}$$

Finally, note that

$$\omega \mid \Psi \sim \text{PG}(b, \Psi) \implies \omega_{nj} \mid \boldsymbol{\beta}_j \sim \text{PG}\big(y_{nj} + r_j, \boldsymbol{\beta}_j^\top \varphi_{\mathbf{W}}(\mathbf{x}_n)\big). \tag{39}$$

If we vectorize across N, we can sample each $\boldsymbol{\beta}_j$ following Polson et al. (2013)'s proposed Gibbs sampler:

$$\begin{aligned} \boldsymbol{\beta}_j \mid \boldsymbol{\omega}_j &\sim \mathcal{N}(\mathbf{m}_{\boldsymbol{\omega}_j}, \mathbf{V}_{\boldsymbol{\omega}_j}) \\ \boldsymbol{\omega}_j \mid \boldsymbol{\beta}_j &\sim \text{PG}(\mathbf{y}_j + r_j, \varphi_{\mathbf{W}}(\mathbf{X})\boldsymbol{\beta}_j) \end{aligned} \tag{40}$$

where

$$\begin{aligned} \boldsymbol{\Omega}_j &= \text{diag}([\boldsymbol{\omega}_{1j}, \ldots, \boldsymbol{\omega}_{Nj}]) \\ \mathbf{V}_{\boldsymbol{\omega}_j} &= (\varphi_{\mathbf{W}}(\mathbf{X})^\top \boldsymbol{\Omega}_j \varphi_{\mathbf{W}}(\mathbf{X}) + \mathbf{B}_0^{-1})^{-1}, \\ \mathbf{m}_{\boldsymbol{\omega}_j} &= \mathbf{V}_{\boldsymbol{\omega}_j}(\varphi_{\mathbf{W}}(\mathbf{X})^\top \boldsymbol{\kappa}_j + \mathbf{B}_0^{-1}\boldsymbol{\beta}_0), \\ \boldsymbol{\kappa}_j &= (\mathbf{y}_j - r_j)/2 \end{aligned} \tag{41}$$

B.2 Sampling r_j

Consider the hierarchical model

$$\begin{aligned} y_{nj} &\sim \text{NB}(r_j, p_{nj}) \\ r_j &\sim \text{Ga}(a_0, 1/h) \\ h &\sim \text{Ga}(b_0, 1/g_0). \end{aligned} \tag{42}$$

Zhou and Carin (2012) showed we can sample r as follows:

$$r_j \sim \mathrm{Ga}\Big(L_j, \frac{1}{-\sum_{n=1}^{N}\log(\max(1-p_{nj}, -\infty))}\Big). \tag{43}$$

where

$$L_j = \sum_{n=1}^{N}\sum_{t=1}^{\ell_j} u_{n\ell}, \qquad u_{n\ell} \sim \log(p_{nj}), \qquad \ell_j \sim \mathrm{Poisson}(-r_j \ln(1-p_{nj})). \tag{44}$$

Zhou has released code[3].

[3] `https://mingyuanzhou.github.io/Softwares/LGNB_Regression_v0.zip`

C MULTINOMIAL GIBBS SAMPLER UPDATES

In order to derive a Gibbs sampler for the multinomial likelihood, we first must use the reparameterization of the likelihood developed in Holmes et al. (2006). We may rewrite the likelihood as

$$\begin{aligned} p(\mathbf{Y}|-) &= \prod_{i=1}^{N} \frac{\Gamma\left(\sum_{j=1}^{J} y_{ij} + 1\right)}{\prod_{j=1}^{J} \Gamma\left(y_{ij}+1\right)} \prod_{j=1}^{J} \left(\frac{\exp\left\{\varphi_{\mathbf{W}}(\mathbf{x}_i)\boldsymbol{\beta}_j\right\}}{\sum_{j=1}^{J} \exp\left\{\varphi_{\mathbf{W}}(\mathbf{x}_i)\boldsymbol{\beta}_j\right\}} \right)^{y_{ij}} \\ &\propto \prod_{i=1}^{N} \prod_{j=1}^{J} \frac{\left(\exp\left\{\varphi_{\mathbf{W}}(\mathbf{x}_i)\boldsymbol{\beta}_j - \xi_{ij}\right\}\right)^{y_{ij}}}{\left(1 + \exp\left\{\varphi_{\mathbf{W}}(\mathbf{x}_i)\boldsymbol{\beta}_j - \xi_{ij}\right\}\right)^{y_{ij} + \sum_{j=1}^{J} y_{ij}}} \end{aligned} \tag{45}$$

Where $\xi_{ij} = \log \sum_{j' \neq j} \exp\{\varphi_{\mathbf{W}}(\mathbf{x}_i)\boldsymbol{\beta}_{j'}\}$. By convention and for identifiability purposes, we set $\boldsymbol{\beta}_J = 0$. We let $\kappa_{ij} = y_{ij} - \sum_{j=1}^{J} y_{ij}/2$. Now that we have written the likelihood in this form, we may use the Pólya-Gamma augmentation trick again:

$$\frac{\left(\exp\left\{\varphi_{\mathbf{W}}(\mathbf{x}_i)\boldsymbol{\beta}_j - \xi_{ij}\right\}\right)^{y_{ij}}}{\left(1 + \exp\left\{\varphi_{\mathbf{W}}(\mathbf{x}_i)\boldsymbol{\beta}_j - \xi_{ij}\right\}\right)^{y_{ij} + \sum_{j=1}^{J} y_{ij}}} \propto \exp\left\{\kappa_{ij}\left(\varphi_{\mathbf{W}}(\mathbf{x}_i)\boldsymbol{\beta}_j - \xi_{ij}\right) - \frac{\omega_{nj}}{2}\left(\varphi_{\mathbf{W}}(\mathbf{x}_i)\boldsymbol{\beta}_j - \xi_{ij}\right)^2\right\}. \tag{46}$$

So this gives us a posterior w.r.t. $\boldsymbol{\beta}_j$ as

$$p(\boldsymbol{\beta}_j \mid \mathbf{y}_j, \mathbf{X}) \propto p(\boldsymbol{\beta}_j) \prod_{n=1}^{N} \exp\left\{\boldsymbol{\kappa}_i\left(\varphi_{\mathbf{W}}(\mathbf{x}_i)\boldsymbol{\beta}_j - \boldsymbol{\xi}_j\right) - \frac{1}{2}\left(\varphi_{\mathbf{W}}(\mathbf{x}_i)\boldsymbol{\beta}_j - \boldsymbol{\xi}_j\right)^T \boldsymbol{\Omega}_n \left(\varphi_{\mathbf{W}}(\mathbf{x}_i)\boldsymbol{\beta}_j - \boldsymbol{\xi}_j\right)\right\} \tag{47}$$

which we can rewrite into a closed form update as

$$\boldsymbol{\beta}_j \mid \boldsymbol{\omega}_j \sim \mathcal{N}(\mathbf{m}_{\boldsymbol{\omega}_j}, \mathbf{V}_{\boldsymbol{\omega}_j}) \tag{48}$$

where

$$\begin{aligned} \boldsymbol{\Omega}_j &= \text{diag}([\boldsymbol{\omega}_{1j}, \ldots, \boldsymbol{\omega}_{Nj}]) \\ \mathbf{V}_{\boldsymbol{\omega}_j} &= (\varphi_{\mathbf{W}}(\mathbf{X})^\top \boldsymbol{\Omega}_j \varphi_{\mathbf{W}}(\mathbf{X}) + \mathbf{B}_0^{-1})^{-1}, \\ \mathbf{m}_{\boldsymbol{\omega}_j} &= \mathbf{V}_{\boldsymbol{\omega}_j}(\varphi_{\mathbf{W}}(\mathbf{X})^\top(\boldsymbol{\kappa}_j + \boldsymbol{\xi}_j^T \boldsymbol{\Omega}_j) + \mathbf{B}_0^{-1}\boldsymbol{\beta}_0), \\ \boldsymbol{\kappa}_j &= \mathbf{y}_j - \frac{1}{2}\sum_{j=1}^{J} y_{ij} \end{aligned} \tag{49}$$

and we sample $\boldsymbol{\Omega}_j$ with

$$\boldsymbol{\omega}_j \mid \boldsymbol{\beta}_j \sim \text{PG}\left(\sum_{j=1}^{J} y_{ij}, \varphi_{\mathbf{W}}(\mathbf{X})\boldsymbol{\beta}_j - \boldsymbol{\xi}_j\right) \tag{50}$$

Although we can sample the $\boldsymbol{\beta}$ parameters in closed form with the Pólya-Gamma augmentation, we still face a problem with obtaining the MAP of $\mathbf{X}$ through optimization when we assume the likelihood is multinomial. Baker (1994) discusses the optimization problem of learning the maximum likelihood estimate (MLE) of the regression parameters in a multinomial logistic link regression problem. It is difficult to optimize parameters with respect to an objective function where the parameters are pushed through the normalization constant of a softmax function. To avoid this problem, we may write the

$$p(\mathbf{y}_i|-) \propto \prod_{j=1}^{J} \left(\frac{\exp\left\{\varphi_{\mathbf{W}}(\mathbf{x}_i)\boldsymbol{\beta}_j\right\}}{\sum_{j=1}^{J} \exp\left\{\varphi_{\mathbf{W}}(\mathbf{x}_i)\boldsymbol{\beta}_j\right\}} \right)^{y_{ij}}, \tag{51}$$

as a Poisson probability mass function with an additional N-dimensional nuisance parameter, $\mathbf{h}$, that we must learn through optimization,

$$p(\mathbf{y}_i|-) \propto \prod_{j=1}^{J} \left(\exp\left\{\mathbf{h} + \varphi_{\mathbf{W}}(\mathbf{x}_i)\boldsymbol{\beta}_j\right\}\right)^{y_{ij}} \exp\left\{-\exp\left\{\mathbf{h} + \varphi_{\mathbf{W}}(\mathbf{x}_i)\boldsymbol{\beta}_j\right\}\right\} \tag{52}$$

where the MLE for the parameters in this Poisson reparamaterization are equal to the MLE learned in the original multinomial likelihood. In our implementation, we use this Poisson parameterization to learn the MAP of $\mathbf{X}$.

D EXPERIMENTS

D.1 Additional results

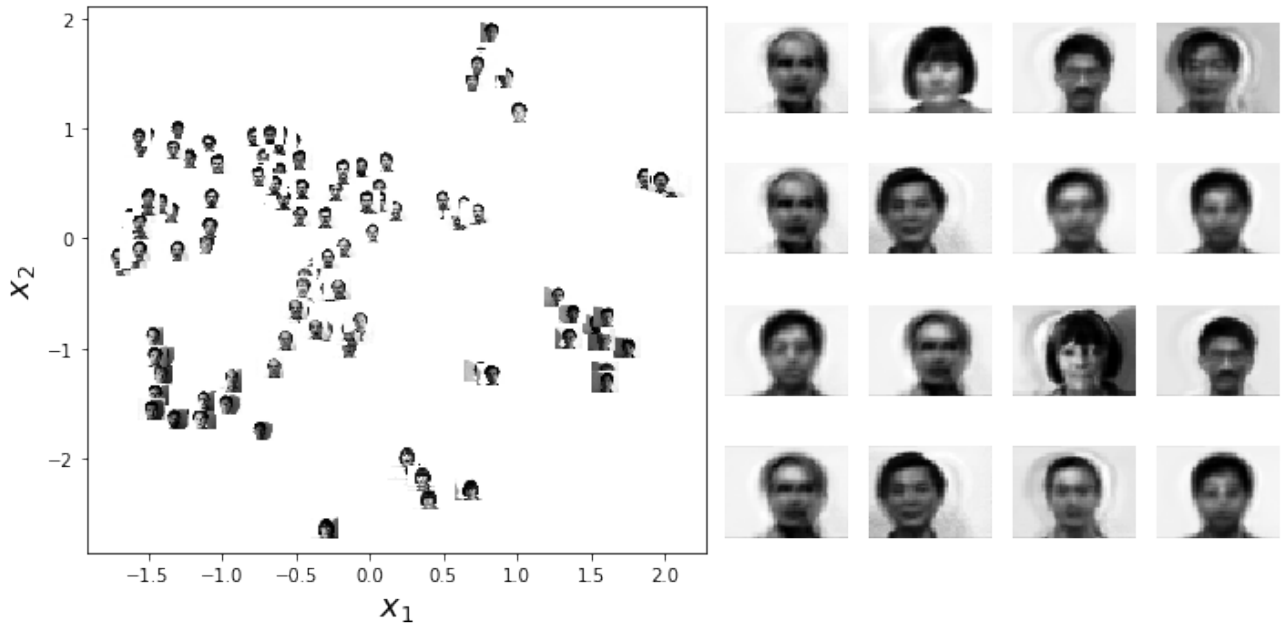

Figure 7: Latent space and generated faces for the Yale dataset using a Poisson RFLVM.

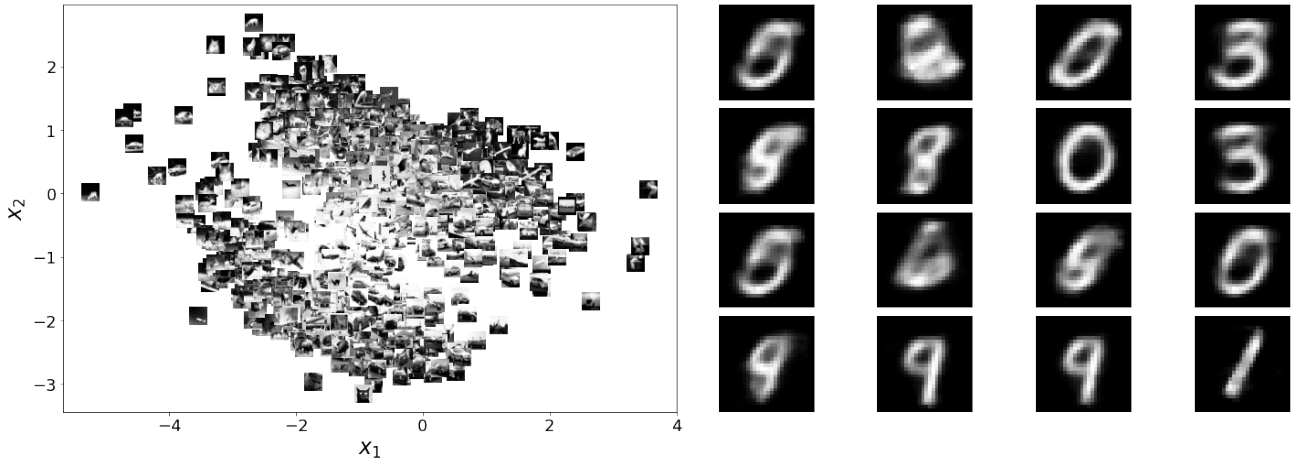

Figure 8: Latent space for CIFAR-10 and generated digits for MNIST using a Poisson RFLVM.

D.2 Data descriptions and preprocessing

- **Bridges:** We used the number of bicycle crossing per day over four East River bridges in New York City[4]. Since these data are unlabeled, we used weekday vs. weekend as binary labels since such information is correlated with bicycle counts (Fig. 9, left).

- **CIFAR-10:** We limited the classes to $[1-5]$ and subampled 400 images for each class for a final dataset of size 2000. We converted the images to grayscale and resized them from 32×32 down to 20×20 pixels.

[4] `https://data.cityofnewyork.us/Transportation/Bicycle-Counts-for-East-River-Bridges/gua4-p9wg`

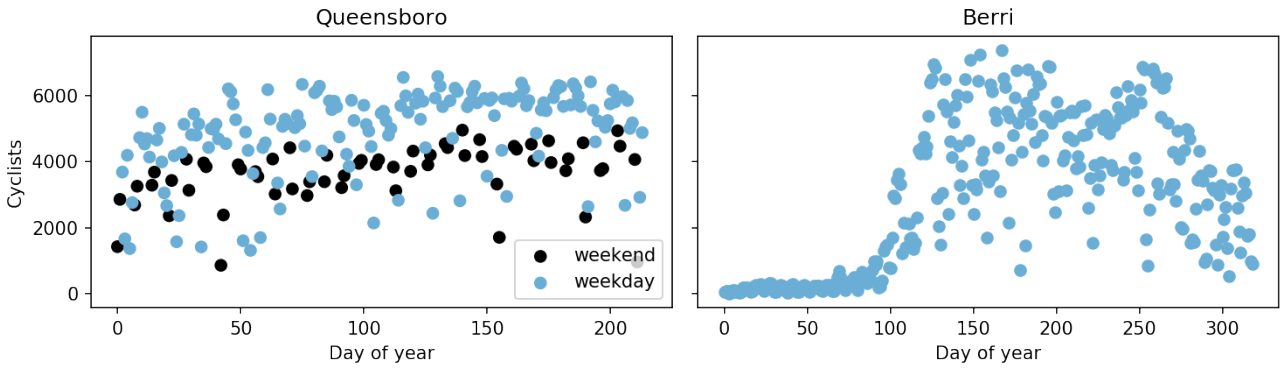

Figure 9: (Left) The number of bicycle crossings over the Queensboro Bridge from April through November 2017. (Right) The number of cyclists on Berri St. in Montreal throughout 2015.

- **Congress:** The word frequency counts from individual members of the 109th Congress (Gentzkow and Shapiro, 2010). Labels are political party: *Democrat*, *Independent*, *Republican*.
- **MNIST:** We limited the dataset size by randomly subsampling 1000 images.
- **Montreal:** We use the number of cyclists per day on eight bicycle lanes in Montreal.[5]. Since these data are unlabeled, we used the four seasons as labels, since seasonality is correlated with bicycle counts (Fig. 9, right).
- **Newsgroups:** The 20 Newsgroups Dataset[6]. We limited the classes to *comp.sys.mac.hardware*, *sci.med*, and *alt.atheism*, and limited the vocabulary to words with document frequencies in the range $10 - 90\%$.
- **Spam:** The SMS Spam dataset from the UCI Machine Learning Repository[7]. Emails are labeled *spam* or *ham* (not spam).
- **Yale:** The Yale Faces Dataset[8]. We used subject IDs as labels.

D.3 Scalability

To assess scalability of RFLVMs, we computed the wall-time in minutes required to fit both RFLVMs and the benchmarks (Table 2). For both the VAE and deep count autoencoder, we trained the neural networks for 2000 iterations (default used in software package[9]). For DLA-GPLVM, we ran the optimizer for 50 iterations (default used in software package[10]). For RFLVMs, we ran the Gibbs samplers for 100 iterations. While results in Table 1 were run for 2000 Gibbs sampling iterations to ensure convergence for all datasets, we found empirically that reducing the number of iterations to 100 did not significantly change the results. We find that RFLVMs are indeed slower than most methods, but not significantly so. For example, on the CIFAR-10 dataset, a VAE takes 23.7 minutes, while a Poisson RFLVM takes 22.9 minutes and a negative binomial RFLVM takes 55.7 minutes. The DLA-GPLVM is slowest, taking 69.8 minutes.

D.4 Miscellany

GPLVM baselines: We used GPy's implementation `BayesianGPLVMMiniBatch`, which supports inducing points and prediction on held out data.

[5] `http://donnees.ville.montreal.qc.ca/dataset/f170fecc-18db-44bc-b4fe-5b0b6d2c7297/resource/64c26fd3-0bdf-45f8-92c6-715a9c852a7b`
[6] `http://qwone.com/~jason/20Newsgroups/`
[7] `https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection`
[8] `http://vision.ucsd.edu/content/yale-face-database`
[9] `https://github.com/theislab/dca`
[10] `https://github.com/waq1129/LMT`

Table 2: Wall-time in minutes for model fitting. Mean and standard error were computed by running each experiment five times.

	PCA	NMF	HPF	LDA	VAE	DCA
Bridges	0.0186 ± 0.0005	0.0182 ± 0.0012	0.0273 ± 0.0002	0.0528 ± 0.0067	1.8193 ± 0.0708	0.5740 ± 0.0255
CIFAR-10	0.4398 ± 0.0743	0.4151 ± 0.0123	1.0894 ± 0.0500	0.8674 ± 0.0199	23.6770 ± 0.3789	1.1341 ± 0.0540
Congress	0.0244 ± 0.0002	0.0245 ± 0.0007	0.7296 ± 0.0824	0.0846 ± 0.0221	4.2919 ± 0.0539	0.5448 ± 0.0134
MNIST	0.2368 ± 0.0064	0.2522 ± 0.0273	1.0004 ± 0.1880	0.3264 ± 0.0237	15.3385 ± 1.8402	0.8719 ± 0.0618
Montreal	0.0171 ± 0.0008	0.0164 ± 0.0001	0.0523 ± 0.0350	0.0632 ± 0.0065	2.0585 ± 0.0947	0.5028 ± 0.0120
Newsgroups	0.0219 ± 0.0006	0.0227 ± 0.0000	0.1757 ± 0.0215	0.1163 ± 0.0344	6.8089 ± 0.7869	0.8551 ± 0.0527
Spam	0.0230 ± 0.0004	0.0235 ± 0.0012	0.3039 ± 0.0419	0.1262 ± 0.0381	6.8448 ± 0.7796	0.7146 ± 0.0453
Yale	0.0884 ± 0.0003	0.0984 ± 0.0064	0.3774 ± 0.0181	0.1381 ± 0.0072	5.5177 ± 0.1645	0.6410 ± 0.0223
	NBVAE	Isomap	DLA-GPLVM	Poisson RFLVM	Neg. binom. RFLVM	Multinomial RFLVM
Bridges	0.0867 ± 0.0157	0.0098 ± 0.0018	0.5182 ± 0.0206	0.3318 ± 0.0135	0.4915 ± 0.0502	0.5715 ± 0.0473
CIFAR-10	2.1002 ± 0.0594	0.4366 ± 0.0034	69.7889 ± 4.2406	22.9299 ± 1.2624	55.6701 ± 2.6837	59.8926 ± 9.9910
Congress	1.5898 ± 0.0725	0.0236 ± 0.0005	45.8584 ± 22.9771	9.8935 ± 0.1041	20.4514 ± 0.3995	94.0656 ± 2.7319
MNIST	2.1104 ± 0.1020	0.2148 ± 0.0019	26.4795 ± 1.5429	17.8148 ± 0.0493	33.8967 ± 4.1385	74.3100 ± 2.1778
Montreal	0.0819 ± 0.0009	0.0080 ± 0.0001	0.8723 ± 0.0237	0.5006 ± 0.0143	0.9291 ± 0.0434	0.8769 ± 0.0376
Newsgroups	0.7432 ± 0.0248	0.0721 ± 0.0008	1088.2659 ± 35.5089	2.6502 ± 0.4063	3.2600 ± 0.0892	2.8393 ± 0.1525
Spam	1.8411 ± 0.0283	0.0795 ± 0.0036	440.5963 ± 26.7444	10.6939 ± 0.4018	17.9958 ± 2.8573	19.0018 ± 2.4612
Yale	0.7931 ± 0.0589	0.0402 ± 0.0026	6.7210 ± 0.1193	9.8992 ± 0.5530	21.6030 ± 0.8839	45.4209 ± 4.4139