Stochastic Inference for Scalable Probabilistic Modeling of Binary Matrices

UNIVERSITY OF CAMBRIDGE

José Miguel Hernández-Lobato[1,2], Neil Houlsby[1,2] and Zoubin Ghahramani[1]

1. Department of Engineering, Cambridge University. 2. Equal contributors.

Neural Information Processing Systems Foundation

1. Introduction

Motivation: Probabilistic matrix factorizations are a powerful tool for modelling matrix **X**.

- They are **robust to overfitting**.
- They can account for different **data types** (continuous, ordinal, count, etc...).
- Fast approximate inference is easily implemented using **variational Bayes**.
- They scale with the number of entries observed in **X**, which is usually low, and not with the size of **X** which can be very large.

Problem: Many real-world **binary** matrices are fully observed. Probabilistic approaches are **infeasible** in this case because they are based on **batch variational algorithms** that require processing all the entries in **X** before producing a single parameter update.

Solution: A novel stochastic algorithm for variational inference on big binary matrices:

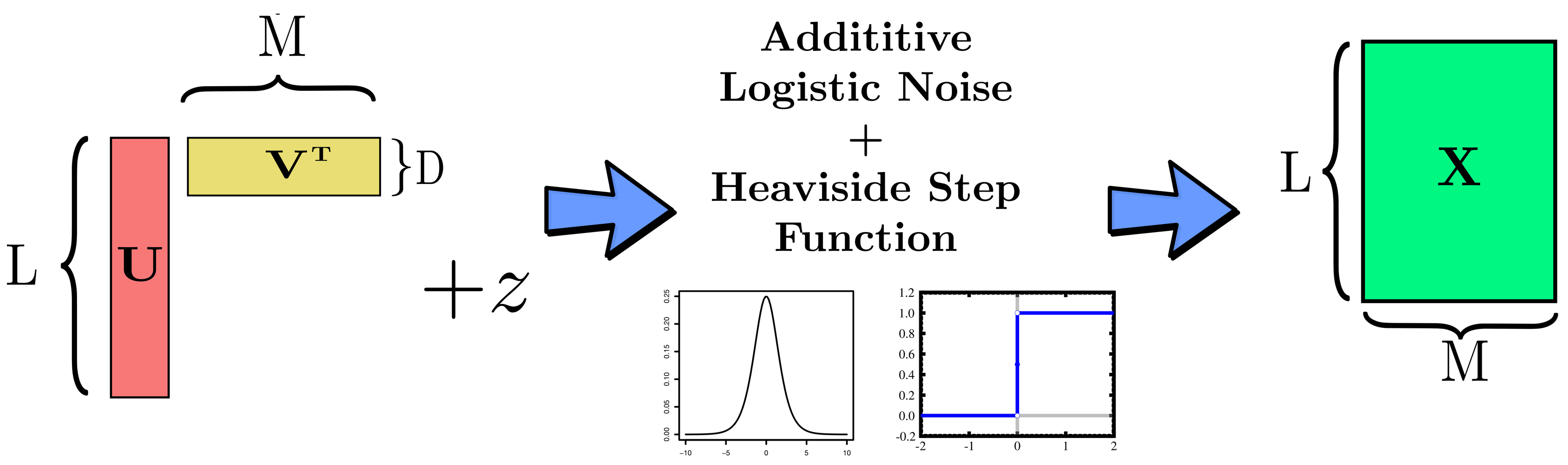
- We apply the **SVI** method of Hoffman et al., 2013 to matrix factorization models.
- We subsample **matrix entries** instead of individual data instances.
- We use **non-uniform** data subsampling strategies which lead to improved predictions.
- We use minibatches to speed up convergence and **adjust the minibatch size on-line**.

2. A Probabilistic Model for Binary Matrices

We use a **logistic likelihood** and a **global bias** parameter.

$$p(\mathbf{X}|\mathbf{U},\mathbf{V},z) = \prod_{i=1}^{L}\prod_{j=1}^{M} p(x_{i,j}|\mathbf{u}_i,\mathbf{v}_j,z) = \prod_{i=1}^{L}\prod_{j=1}^{M}\left[\sigma(\mathbf{u}_i\mathbf{v}_j^{\mathrm{T}}+z)^{x_{i,j}}\sigma(-\mathbf{u}_i\mathbf{v}_j^{\mathrm{T}}-z)^{1-x_{i,j}}\right],$$

$$p(\mathbf{U}) = \prod_{i=1}^{L}\prod_{d=1}^{D}\mathcal{N}(u_{i,d}|\bar{u}^0_{i,d},\tilde{u}^0_{i,d}),\quad p(\mathbf{V}) = \prod_{j=1}^{M}\prod_{d=1}^{D}\mathcal{N}(v_{j,d}|\bar{v}^0_{j,d},\tilde{v}^0_{j,d}),\quad p(z) = \mathcal{N}(z|\bar{z}^0,\tilde{z}^0).$$



3. Variational Bayes

We approximate the posterior with a tractable $q(\mathbf{U},\mathbf{V},z)$ indexed by variational parameters $\mathbf{\Phi}$. We optimize q by maximizing the **Evidence Lower Bound (ELBO)** with respect to $\mathbf{\Phi}$.

Jensen's Inequality — The ELBO

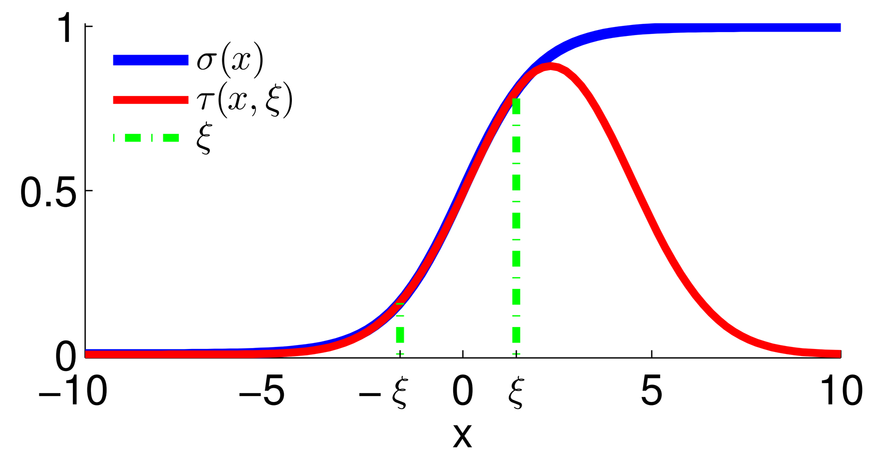
$$\log p(\mathbf{X}) = \log\int p(\mathbf{X},\mathbf{U},\mathbf{V},z)d\mathbf{U}d\mathbf{V}dz \geq \mathbb{E}_{q(\mathbf{U},\mathbf{V},z)}\left[\log\frac{p(\mathbf{X},\mathbf{U},\mathbf{V},z)}{q(\mathbf{U},\mathbf{V},z)}\right] \triangleq \mathcal{L}(\mathbf{\Phi}).$$

$$q(\mathbf{U},\mathbf{V},z) = \left[\prod_{i=1}^{L}\prod_{d=1}^{D}\mathcal{N}(u_{i,d}|\bar{u}_{i,d},\tilde{u}_{i,d})\right]\left[\prod_{j=1}^{M}\prod_{d=1}^{D}\mathcal{N}(v_{j,d}|\bar{v}_{j,d},\tilde{v}_{j,d})\right]\mathcal{N}(z|\bar{z},\tilde{z}).$$

$$\mathbf{\Phi} = \{\{\{\bar{u}_{i,d},\tilde{u}_{i,d},\}_{i=1}^{L},\{\bar{v}_{j,d},\tilde{v}_{j,d}\}_{j=1}^{M}\}_{d=1}^{D},\bar{z},\tilde{z}\}.$$

4. Local Variational Approximation

We lower bound each logistic function in the ELBO with a Gaussian: $\sigma(x) \geq \tau(x,\xi)$.



(Jaakkola & Jordan, 1997)

Figure: blue = logistic function, red = lower bound, tight at $x = \pm\xi$.

We add an **extra variational parameter** $\xi_{i,j}$ for each matrix entry: $\mathbf{\Xi} = \{\{\xi_{i,j}\}_{i=1}^{L}\}_{j=1}^{M}$.

The model is now **conjugate** with Gaussian complete conditionals.

5. Stochastic Inference

We use stochastic gradient descent to optimize $\mathcal{L}(\mathbf{\Phi}) \triangleq \arg\max_{\mathbf{\Xi}}\mathcal{L}(\mathbf{\Phi},\mathbf{\Xi})$.

1 - Sample a matrix entry $x_{i,j}$ with probability $p(i,j)$.

2 - Compute a noisy estimate of $\mathcal{L}(\mathbf{\Phi})$ which includes only a few of the terms in $\mathcal{L}(\mathbf{\Phi})$:

$$\mathcal{L}_{\text{noisy}}(\mathbf{\Phi}) = c_{i,j}\underbrace{f(x_{i,j},\xi_{i,j},\mathbf{\Phi}_{i,j})}_{\text{likelihood}} + \sum_{d=1}^{D}\underbrace{g(\bar{u}_{i,d},\tilde{u}_{i,d})}_{\text{prior on } u_{i,d}} + \sum_{d=1}^{D}\underbrace{g(\bar{v}_{j,d},\tilde{v}_{j,d})}_{\text{prior on } v_{j,d}} + \underbrace{g(\bar{z},\tilde{z})}_{\text{prior on z}}.$$

3 - Optimize $\xi_{i,j}$ and choose the values of the scaling constant $c_{i,j}$.

4 - Update $\mathbf{\Phi}_{i,j} = \{\{\bar{u}_{i,d},\tilde{u}_{i,d},\bar{v}_{j,d},\tilde{v}_{j,d}\}_{d=1}^{D},\{\bar{z},\tilde{z}\}\}$ by taking a small step in the direction of the gradient of $\mathcal{L}_{\text{noisy}}$.

6. Natural Gradients and Minibatches

We work with **natural parameters**: $\mathring{\mathbf{u}}_{i,d} = [\bar{u}_{i,d}/\tilde{u}_{i,d},\tilde{u}_{i,d}^{-1}]^{\mathrm{T}}$. Let $\mathring{\mathbf{u}}^\star_{i,d}$ be the maximizer of $\mathcal{L}_{\text{noisy}}$ with respect $\mathring{\mathbf{u}}_{i,d}$. The **natural gradient** with respect to this parameter is

$$\hat{\nabla}\mathcal{L}_{\text{noisy}}(\mathring{\mathbf{u}}_{i,d}) = \mathring{\mathbf{u}}^\star_{i,d} - \mathring{\mathbf{u}}_{i,d}.$$

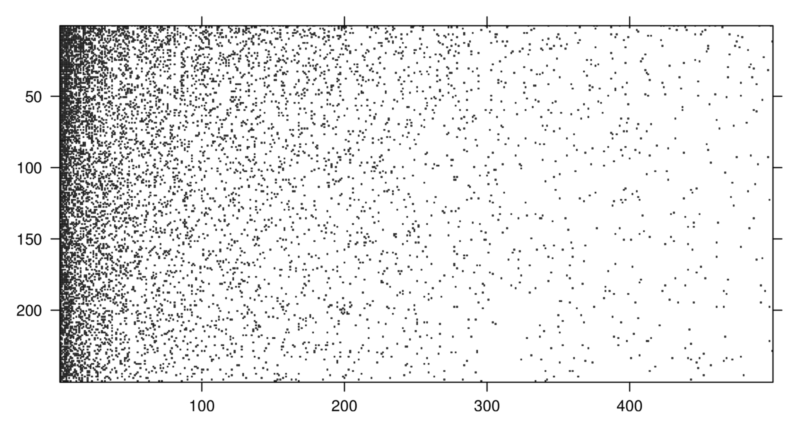
The stochastic update of step size ρ in the direction of the natural gradient is then

$$\mathring{\mathbf{u}}^{\text{new}}_{i,d} = \mathring{\mathbf{u}}^{\text{old}}_{i,d} + \rho\hat{\nabla}\mathcal{L}_{\text{noisy}}(\mathring{\mathbf{u}}_{i,d}) = (1-\rho)\mathring{\mathbf{u}}^{\text{old}}_{i,d} + \rho\mathring{\mathbf{u}}^\star_{i,d}.$$

To use **minibatches** of size S, we replace $\mathring{\mathbf{u}}^\star_{i,d}$ with $\mathring{\mathbf{u}}^{\star,\text{avg}}_{i,d} = \frac{1}{n(i)}\sum_{s=1}^{n(i)}\mathring{\mathbf{u}}^{\star,s}_{i,d}$, where $n(i)$ is the number of entries from the i-th row found in the last S subsampled entries and $\mathring{\mathbf{u}}^{\star,s}_{i,d}$ is the maximizer of $\mathcal{L}_{\text{noisy}}$ when the s-th of those entries in the i-th row is subsampled.

7. Non-uniform Data Subsampling Strategies

Real-world binary matrices are usually very **sparse**, with frequencies for ones and zeros that **change considerably** across rows and across columns.



We use different subsampling strategies:

- **S-Uniform**: $p(i,j) = 1/(LM)$.
- **S-Balanced**: $p(i,j) = 1/(2\sum_{a=1}^{L}\sum_{b=1}^{M}\mathbb{I}[x_{i,j}=x_{a,b}])$.
- **S-Biased**: $p(i,j) = r_i^{(1-x_{i,j})}c_j^{(1-x_{i,j})}[2\sum_{a=1}^{L}\sum_{b=1}^{M}\mathbb{I}[x_{i,j}=x_{a,b}]r_a^{(1-x_{a,b})}c_b^{(1-x_{a,b})}]^{-1}$.

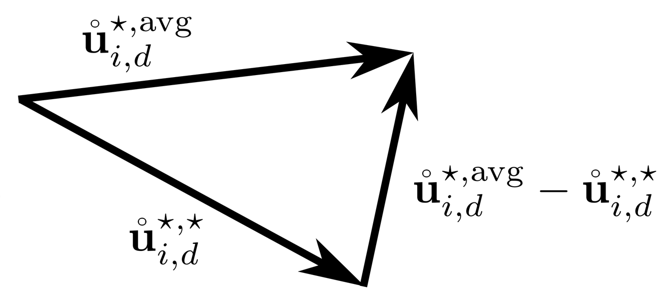
$r_i^{(0)}$ and $r_i^{(1)}$ are the number of zeros and ones in the i-th row of **X** and likewise $c_j^{(0)}$ and $c_j^{(1)}$ count the number of zeros and ones in the j-th column.

8. Automatically Adjusting the Minibatch Size Online

The minibatch size S is important. **Trade off**: noise reduction vs. frequency of updates. We bound the relative error of $\mathring{\mathbf{u}}^{\star,\text{avg}}_{i,d}$ with respect to its expectation $\mathring{\mathbf{u}}^{\star,\star}_{i,d} = \mathbf{E}[\mathring{\mathbf{u}}^{\star,\text{avg}}_{i,d}]$.

Markov's Inequality

$$\delta = p\left(\frac{||\mathring{\mathbf{u}}^{\star,\text{avg}}_{i,d}-\mathring{\mathbf{u}}^{\star,\star}_{i,d}||_2^2}{||\mathring{\mathbf{u}}^{\star,\star}_{i,d}||_2^2}\geq\theta\right) \leq \frac{\mathbb{E}\left[||\mathring{\mathbf{u}}^{\star,\text{avg}}_{i,d}-\mathring{\mathbf{u}}^{\star,\star}_{i,d}||_2^2\right]}{\theta||\mathring{\mathbf{u}}^{\star,\star}_{i,d}||_2^2}$$



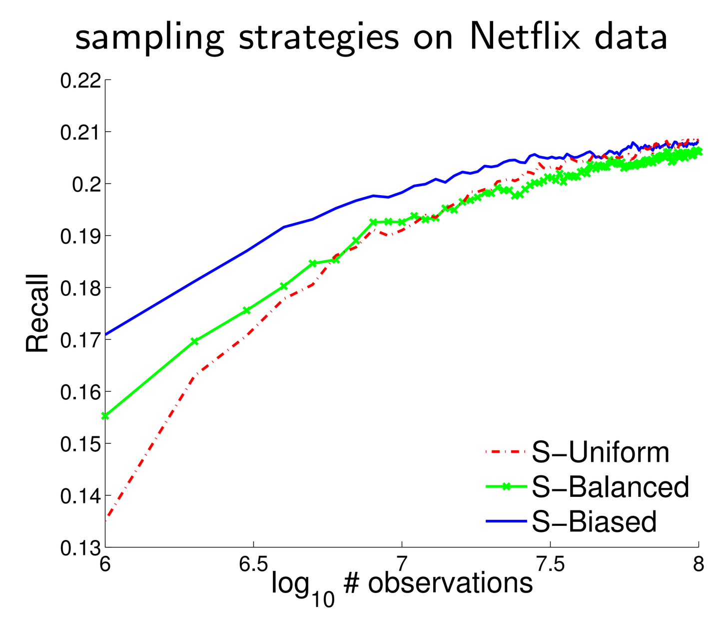
Solving for S, we obtain that S should be proportional to the **noise to signal ratio** in $\mathring{\mathbf{u}}^\star_{i,d}$.

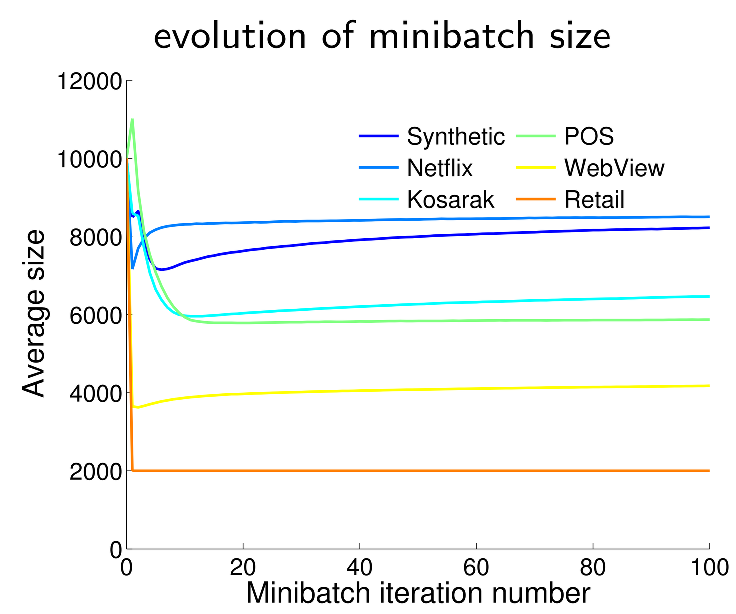
$$S = \frac{||\text{Var}[\mathring{\mathbf{u}}^\star_{i,d}]||_1}{\theta\delta p(i)||\mathbb{E}[\mathring{\mathbf{u}}^\star_{i,d}]||_2^2}.$$

- Only a single effective parameter $\theta\delta$.
- We estimate $\mathbf{E}[\mathring{\mathbf{u}}^\star_{i,d}]$ and $\text{Var}[\mathring{\mathbf{u}}^\star_{i,d}]$ online.
- We re-update S after S samples have been drawn.

9. Sampling Strategies and Evolution of MiniBatchSize

The strategy **S-Biased** performs best. The minibatch size S converges very quickly.





10. Results on Synthetic and Real-world Datasets