Learning Sigmoid Belief Networks via Monte Carlo Expectation Maximization: Supplemental Materials

Zhao Song[†] **Ricardo Henao**[†] **David Carlson**[‡] **Lawrence Carin**[†]
[†]Department of Electrical and Computer Engineering, Duke University
[‡]Department of Statistics and Grossman Center for Statistics of Mind, Columbia University

1 Properties of the Pólya-Gamma Distribution

We first provide the definition of the Pólya-Gamma (PG) distribution and summarize several of its key properties (Polson et al., 2013).

Definition 1. *A random variable X has a Pólya-Gamma distribution with parameters $b > 0$ and $c \in \mathbb{R}$, denoted as $X \sim \mathrm{PG}(b, c)$, if*

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

where $g_k \sim Gamma(b, 1)$ are independent gamma random variables.

The PG distribution has a closed form mean, i.e.,

$$\mathbb{E}(\omega) = \frac{b}{2c} \tanh\left(\frac{c}{2}\right) = \frac{b}{2c}\left(\frac{\exp(c)-1}{\exp(c)+1}\right) \tag{A1}$$

Binomial likelihoods parameterized by log-odds can be represented as (Polson et al., 2013),

$$\frac{\big[\exp(\psi)\big]^v}{\big[1+\exp(\psi)\big]^u} = 2^{-u} \exp(\kappa\psi) \int_0^\infty \exp(-\frac{\omega\psi^2}{2})\, p(\omega)\, d\omega \tag{A2}$$

where $\kappa = v - \frac{u}{2}$ and $\omega \sim$ PG $(u, 0)$. The conditional distribution is (Polson et al., 2013)

$$p(\omega|\psi) = \frac{\exp(-\omega\psi^2/2)\, p(\omega)}{\int_0^\infty \exp(-\omega\psi^2/2)\, p(\omega)\, d\omega} \tag{A3}$$

satisfies the PG distribution and is parameterized as $\omega|\psi \sim$ PG (b, ψ).

2 Derivation of Inner EM Algorithm

For clarity, we focus on one-layer model and omit prior and bias terms. We also set the number of samples $K = 1$ for notational simplicity.

Starting from (3) of the main text, we first need to compute the expected complete-data log likelihood $\widehat{\mathcal{Q}}$ in the inner expectation-maximization (EM) algorithm as

$$\begin{aligned}
&\widehat{\mathcal{Q}}(W|W^{(t)}) = \mathbb{E}_{\boldsymbol{\omega}|\boldsymbol{v},W,\boldsymbol{h}}\big[\sum_k \ln p(\boldsymbol{\omega}, \boldsymbol{v}|\boldsymbol{h}_k, W^{(t)})\big] \\
&\quad + \mathbb{E}_{\tau|W,\lambda}\big[\ln p(W^{(t)}, \tau|\lambda)\big] \\
=\;& \mathbb{E}_{\boldsymbol{\omega}|\boldsymbol{v},W,\boldsymbol{h}}\big[\sum_n \sum_i \ln p(v_{n,i}, \omega_{n,i}|\boldsymbol{h}_k, W^{(t)}) \\
&\quad + \sum_i \sum_j \mathbb{E}_{\tau_{i,j}|W_{i,j},\lambda}\big[\ln p(W_{i,j}^{(t)}, \tau_{i,j}|\lambda)\big] \\
=\;& \sum_n \sum_i \kappa_{n,i}^{(t)} \psi_{n,i}^{(t)} - \frac{1}{2} \widehat{\omega}_{n,i}^{(t+1)} (\psi_{n,i}^{(t)})^2 && \text{(A4)} \\
&\quad + \sum_i W_{i\cdot}\, \Phi_i^{(t+1)}\, (W_{i\cdot})^T + \text{const} && \text{(A5)}
\end{aligned}$$

where

$$\kappa_{n,i}^{(t)} = v_{n,i} - \frac{u_{n,i}}{2} \tag{A6}$$

$$\widehat{\omega}_{n,i}^{(t+1)} = \frac{u_{n,i}}{2\psi_{n,i}^{(t)}} \tanh(\frac{\psi_{n,i}^{(t)}}{2}) \tag{A7}$$

$$\Phi_i^{(t+1)} = \mathrm{diag}\left(\frac{\lambda}{|W_{i,1}^{(t)}|}, \ldots, \frac{\lambda}{|W_{i,J_0}^{(t)}|}\right) \tag{A8}$$

(A7) holds because $\omega_{n,i}|h_{n,i}, \boldsymbol{v}, W^{(t)} \sim \mathrm{PG}(u_{n,i}, \psi_{n,i}^{(t)})$. The derivation for (A8) comes from Figueiredo (2003).

Reordering terms in (A5) gives $\widehat{\mathcal{Q}}(W|W^{(t)})$ the following form:

$$\begin{aligned}
\widehat{\mathcal{Q}}(W|W^{(t)}) &= \frac{1}{2} \sum_i \big[W_{i\cdot}(\Phi_i^{(t+1)} + X_i^{(t+1)}) W_{i\cdot}^T\big] \\
&\quad - \sum_i W_{i\cdot} \boldsymbol{\eta}_i^{(t)} && \text{(A9)}
\end{aligned}$$

where

$$X_i^{(t+1)} = \sum_{n=1}^{N} \widehat{\omega}_{n,i}^{(t+1)} \boldsymbol{h}_n \boldsymbol{h}_n^T$$

$$\boldsymbol{\eta}_i^{(t)} = \sum_{n=1}^{N} \kappa_{n,i}^{(t)} \boldsymbol{h}_n$$

This now finishes derivations of all intermediate variables in the E step.

Since (A9) is a quadratic function of $W_{i\cdot}$, we can take the gradient with respect to W and set it to be zero, to obtain the following M step update: $\forall\, i = 1, \ldots, J_0$

$$\left[W_{i\cdot}^{(t+1)}\right]^T = \left[X_i^{(t+1)} + \Phi_i^{(t+1)}\right]^{-1} \boldsymbol{\eta}_i^{(t)}. \quad \text{(A10)}$$

3 Online MCEM Algorithm

Suppose that the *mini-batch* size is N_{mini} and the stepsize for the mth mini-batch is set to $\gamma_m = (m+2)^{-\alpha}$, as suggested in Liang and Klein (2009). Then, $\forall i = 1, \ldots, J_0$, we update the sufficient statistics as

$$\begin{aligned}
\widetilde{\kappa}_{i,m} &= (1-\gamma_m)\widetilde{\kappa}_{i,m-1} + \gamma_m \sum_{n=1}^{N_{\text{mini}}} \bar{\kappa}_{n,i,m} \\
\widetilde{X}_{i,m} &= (1-\gamma_m)\widetilde{X}_{i,m-1} \\
&\quad + \gamma_m \sum_{n=1}^{N_{\text{mini}}} \bar{\omega}_{n,i,m} \bar{\boldsymbol{h}}_{n,m} (\bar{\boldsymbol{h}}_{n,m})^T \quad \text{(A11)} \\
\widetilde{\boldsymbol{\eta}}_{i,m} &= (1-\gamma_m)\widetilde{\boldsymbol{\eta}}_{i,m-1} + \gamma_m \sum_{n=1}^{N_{\text{mini}}} \bar{\kappa}_{n,i,m} \bar{\boldsymbol{h}}_{n,m}
\end{aligned}$$

Subsequently, we summarize the online MCEM algorithm for the MAP estimate in Algorithm 1. The ML version can be derived accordingly.

Algorithm 1 Online MCEM algorithm for MAP estimate.

Input: Mini-batch size N_{mini}, dataset size N, learning rate α, initial parameters $\boldsymbol{\theta}^{(0)}$, $m = 0$.
repeat
  for $k = 1$ to N/N_{mini} **do**
    Read the kth mini-batch data $\boldsymbol{v}_k$.
    Set the stepsize $\gamma_m = (m+2)^{-\alpha}$.
    Compute the $\mathcal{Q}$ function as shown in (A9) with $\boldsymbol{v}_k$.
    Update the expected sufficient statistics shown in (A11).
    Update $\boldsymbol{\theta}$ by the M step shown in (A10).
    $m \leftarrow m + 1$.
  end for
until Convergence

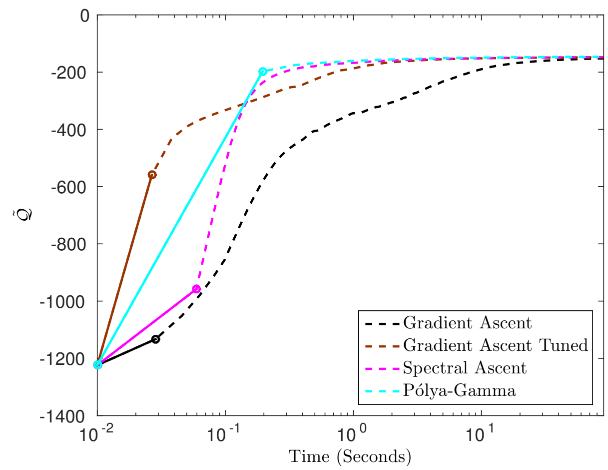

Figure A1: The value of $\tilde{\mathcal{Q}}$ as a function of running time for different optimization schemes.

4 Evaluation Details for Perplexities

Following Zhou et al. (2012); Gan et al. (2015), we split the test documents by a random 80/20% partition: 80% of the words are used to infer the document-specific local variables and the remaining 20% of the words are held out to compute the predictive perplexity. We denote the hold-out documents as a matrix $Y \in \mathbb{Z}_{\geq 0}^{P \times N}$ where P is the vocabulary size and N is the document size. Consequently, the distribution of the count vector $\boldsymbol{y}_n$ can be modelled as the following Replicated Softmax Model (RSM):

$$\boldsymbol{y}_n \sim \text{Multi}(D_n\,;\,\boldsymbol{\beta}_n)$$

$$\beta_{p,n} = \frac{\exp(W_{p\cdot}\boldsymbol{h}_n + c_p)}{\sum_{p'=1}^{P} \exp(W_{p'\cdot}\boldsymbol{h}_n + c_{p'})}$$

where $\boldsymbol{y}_n$ is the nth document in Y and $D_n = \sum_{p=1}^{P} Y_{p,n}$ is the number of words in document n, $\{W, \boldsymbol{b}, \boldsymbol{c}\}$ are the learned parameters from the training document. The test perplexity is then computed as Gan et al. (2015):

$$\exp\left(-\frac{1}{y_{\cdot\cdot}} \sum_{p=1}^{P} \sum_{n=1}^{N} Y_{p,n} \log \beta_{p,n}\right)$$

where $y_{\cdot\cdot} = \sum_{p=1}^{P} \sum_{n=1}^{N} Y_{p,n}$.

5 Additional Results

We recreated Figure 1 of the main text with running time in log-scale. Since $\tilde{\mathcal{Q}}$ is a concave function, all methods eventually converge to the same final maxima, as shown in Figure A1. A bias tern is added to the running time to ensure an appropriate starting point in the log-scale.

References

Figueiredo, M. A. (2003). Adaptive sparseness for supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*

Gan, Z., Chen, C., Henao, R., Carlson, D., and Carin, L. (2015). Scalable deep Poisson factor analysis for topic modeling. In *ICML*.

Liang, P. and Klein, D. (2009). Online EM for unsupervised models. In *NAACL*.

Polson, N. G., Scott, J. G., and Windle, J. (2013). Bayesian inference for logistic models using Pólya–Gamma latent variables. *J. Am. Statistical Association*.

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