A Forest Mixture Bound for Block-Free Parallel Inference

Neal G. Lawton and **Greg Ver Steeg** and **Aram Galstyan**
Information Sciences Institute
University of Southern California
Marina del Rey, CA 90292

Abstract

Coordinate ascent variational inference is an important algorithm for inference in probabilistic models, but it is slow because it updates only a single variable at a time. Block coordinate methods perform inference faster by updating blocks of variables in parallel. However, the speed and convergence of these algorithms depends on how the variables are partitioned into blocks. In this paper, we give a convergent parallel algorithm for inference in deep exponential families that doesn't require the variables to be partitioned into blocks. We achieve this by lower bounding the ELBO by a new objective we call the *forest mixture bound* (FM bound) that separates the inference problem for variables within a hidden layer. We apply this to the simple case when all random variables are Gaussian and show empirically that the algorithm converges faster for models that are inherently more forest-like.

1 INTRODUCTION

Inference in directed models like deep exponential families (DEF's) [Ranganath et al., 2015] is complicated by the "explaining away effect": for a directed model with observed variables $\boldsymbol{x} \in \mathbb{R}^n$ and latent variables $\boldsymbol{y} \in \mathbb{R}^m$, independent "causes" y_j become dependent given an observed "effect" x_i. To handle this, the coordinate ascent variational inference (CAVI) algorithm iteratively updates the variational distribution for a single latent variable y_j while holding the variational distribution for all other latent variables fixed [Blei et al., 2017].

Though the y_j's are not conditionally independent given $\boldsymbol{x}$ except in exceedingly simple models, in many cases the y_j's are *nearly* conditionally independent. Is there a way to perform parallel inference in such models, or do we have to resort to the serial coordinate algorithm?

Block methods provide one avenue for parallel inference. These algorithms work by first partitioning the latent variables into a collection of blocks, and then iteratively updating a variable from each block in parallel. However, the speed (as in MCMC methods [Terenin et al., 2015]) or convergence (as in Hogwild methods [Recht et al., 2011]) of the resulting algorithm will depend on how the variables are blocked, and finding a good choice of blocking for an arbitrary model can be difficult.

The main contribution of this paper is a novel lower bound on log-likelihood we call the forest mixture bound (FM bound) that separates the problem of inference for each variable in a hidden layer. This allows all the variables in a layer to be updated in parallel, without the use of blocks. We call the resulting parallel inference algorithm the forest mixture algorithm (FM algorithm).

We study in detail the case when all the random variables in the DEF are Gaussian. We then demonstrate on both synthetic and real-world data the proposed method achieves faster convergence compared to existing methods.

2 RELATED WORK

Hogwild Block Methods There are two types of block methods for inference. The first is Hogwild-type algorithms [Recht et al., 2011][Sa et al., 2016] [Wang and Banerjee, 2014] [Zhao et al., 2014]. After partitioning the variables into blocks, these algorithms iteratively choose a single variable from each block and update as in CAVI, but in parallel [Sa et al., 2016]. These algorithms are guaranteed to converge only in certain cases, e.g., when the blocks are conditionally independent [Johnson et al., 2013].

Convergent Block Methods Instead of making CAVI updates in parallel, block algorithms may achieve convergence by making small parallel updates [Sontag and Jaakkola, 2009]. For example, "exact" asynchronous Gibbs sampling randomly rejects each block update according to an MCMC rejection ratio [Terenin et al., 2015]. If the blocks are chosen poorly, the rejection rate will increase and the rate of convergence will decrease [Singh et al., 2017].

In either type of block method, the performance of the algorithm depends on how the variables are blocked. In a distributed computation setting, blocking is necessary since each worker can only store a fraction of all variables in local memory. In this case, the FM bound provides a method for updating variables within a block or worker in parallel, instead of updating only a single variable in each block at a time.

Amortized Inference Instead of treating inference as an inverse problem that has to be solved for each observation, VAE's train inference network (encoder) so the cost of inference is amortized over many observations [Kingma and Welling, 2013]. Once the encoder is trained, inference for any observation can be performed quickly with a single pass through the inference network. Encoder-free methods like ours may still be useful in the case when we have a trained generative model (decoder) but no trained encoder and want to perform inference for only a few samples or, more likely, for when we want to improve the solution produced by the encoder at test time.

Undirected Models Besides directed models, there is a wide literature for fast inference in undirected models [Baqué et al., 2016] [Singh et al., 2010]. Note that inference in undirected models like Deep Restricted Boltzmann Machines [Salakhutdinov and Hinton, 2009] can already be parallelized: non-consecutive layers can be updated in parallel in red-black fashion. In fact, the same degree of parallelization can be achieved in a directed model using our technique. While there is also a wide literature on bounding the log-partition function of an *undirected* model [Wainwright et al., 2005], we derive the FM bound by lower bounding the log-partition function of a *directed* model. The technique we use may be applicable to undirected models, but that is not explored in this paper.

Structure Learning The FM bound we derive is closely related to an interesting family of models called *forest mixture models*. These models may be applicable to the problem of structure learning, where the task is to infer the graphical structure of the underlying model from data [Chow and Liu, 1968]. However, in this paper we narrowly focus on the problem of inference in a *given* generative model, not on training a new one.

3 PRELIMINARIES

Vector-valued variables are written in bold. The component-wise product of two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ is denoted $u \odot v$. Unless stated otherwise, all expectations, including the variance $\mathrm{Var}[\cdot]$, standard deviation $\mathrm{Std}[\cdot]$, and conditional entropy $H(\boldsymbol{y}|\boldsymbol{x})$, are taken with respect to the variational distribution $q(\boldsymbol{y}|\boldsymbol{x})$, though we sometimes write this explicitly for emphasis.

An *exponential family* of distributions is a family of distributions of the form

$$p(x) = \exp\{g(x) + t(x) \cdot \eta - a(\eta)\} \tag{1}$$

Where g is the log-base measure, t are the sufficient statistics, η are the natural parameters, and a is the log-partition function. When η is a function of another random variable $\boldsymbol{y}$, e.g., $\eta = b + \boldsymbol{w} \cdot \boldsymbol{y}$, we will sometimes write $\eta = \eta(\boldsymbol{y})$ for emphasis.

We denote the Gaussian probability density function with mean μ and variance σ^2 as $\mathcal{N}(\mu, \sigma^2)$. When we write $\log p(x) \propto f(x)$, we mean $\log p(x) = f(x) + constant$.

3.1 FOREST MIXTURE MODELS

Consider a general directed model with a single layer of observed variables $\boldsymbol{x} \in \mathbb{R}^n$ and latent variables $\boldsymbol{y} \in \mathbb{R}^m$. The joint distribution $p(\boldsymbol{x}, \boldsymbol{y})$ takes the form

$$p(\boldsymbol{x}, \boldsymbol{y}) = \left[\prod_{j=1}^{m} p(y_j)\right] \left[\prod_{i=1}^{n} p(x_i|\boldsymbol{y})\right] \tag{2}$$

A directed model is a *forest model* if each x_i has exactly one parent in the model's directed dependency graph; they are so-named because the resulting graphical model is a forest with one tree per latent variable y_j. These models are particularly simple because the y_j's are conditionally independent given $\boldsymbol{x}$. Let $\boldsymbol{e}_i \in I^m$ be the one-hot vector indicating the parent of x_i, so $e_{ij} = 1$ if and only if y_j is the parent of x_i. Then we can write

$$p(x_i|\boldsymbol{y}) = \prod_{j=1}^{m} p(x_i|y_j)^{e_{ij}} \tag{3}$$

Suppose we want to fit a forest model to data, but we don't know which x_i's should be the children of which y_j's. One way to handle this uncertainty is to treat the $\boldsymbol{e}_i$'s as independent latent random variable that have to

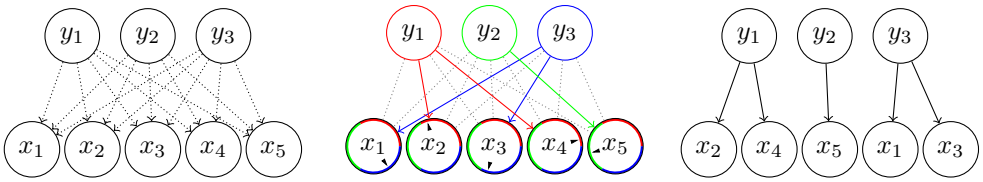


Figure 1: Visualization of sampling from a forest mixture model. (a) In a forest mixture model, the edges between $\boldsymbol{x}$ and $\boldsymbol{y}$ are unknown random variables. (b) To sample from the model, first the parent of each x_i is chosen independently at random according to $p(\boldsymbol{e}_i)$. In this visualization, each $p(e_i)$ is uniform over the latent variables. (c) After sampling a forest structure from $p(\boldsymbol{e})$, $\boldsymbol{x}$ and $\boldsymbol{y}$ are sampled according to the resulting forest model.

be inferred, just like $\boldsymbol{y}$. To do this, we must first define a prior $p(\boldsymbol{e}_i)$ for each i. Given such a prior, the joint distribution over $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{e} \equiv \{\boldsymbol{e}_i\}_{i=1}^n$ is

$$p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{e}) = \left[\prod_{i=1}^n p(\boldsymbol{e}_i)\right] \left[\prod_{j=1}^m p(y_j)\right] \left[\prod_{i=1}^n p(x_i|\boldsymbol{y}, \boldsymbol{e}_i)\right] \tag{4}$$

The resulting model is a *forest mixture model* (FMM): to sample from this model, we first draw a random forest structure by sampling from the prior $p(\boldsymbol{e})$; then, $\boldsymbol{x}$ and $\boldsymbol{y}$ are sampled from the selected forest model.

Though the y_j's are no longer conditionally independent given $\boldsymbol{x}$, they are independent given $\boldsymbol{x}$ and $\boldsymbol{e}$. Similarly, the $\boldsymbol{e}_i$'s are conditionally independent given $\boldsymbol{x}$ and $\boldsymbol{y}$. To see this, define $\hat{p}(x_i|y_j) \equiv p(x_i|y_j, e_{ij} = 1)$. Then the joint distribution can be written

$$p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{e}) = \left[\prod_{i=1}^n p(\boldsymbol{e}_i)\right] \left[\prod_{j=1}^m p(y_j)\right] \prod_{i=1}^n \prod_{j=1}^m \hat{p}(x_i|y_j)^{e_{ij}} \tag{5}$$

In the next section, we will use the mean-field variational ELBO for this model, which for a given variational distribution $q(\boldsymbol{y}, \boldsymbol{e}|\boldsymbol{x})$ is

$$\begin{aligned}
\log p(\boldsymbol{x}) &\geq \mathbb{E}[\log p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{e})] - D_{KL}(q(\boldsymbol{y}, \boldsymbol{e}|\boldsymbol{x}) \| p(\boldsymbol{y}, \boldsymbol{e})) \\
&= \sum_{i=1}^n \sum_{j=1}^m \mathbb{E}[e_{ij}] \mathbb{E}[\log \hat{p}(x_i|y_j)] \\
&\quad - \sum_{j=1}^m D_{KL}(\, q(y_j|\boldsymbol{x}) \| \, p(y_j)) \\
&\quad - \sum_{i=1}^n D_{KL}(\, q(\boldsymbol{e}_i|\boldsymbol{x}) \| \, p(\boldsymbol{e}_i))
\end{aligned} \tag{6}$$

4 THE FOREST MIXTURE BOUND

For simplicity, we only consider shallow models in this section. The extension to deep models is straightforward (see Appendix C).

A single-layer *deep exponential family* (DEF) model is a directed model with a single layer of observed variables $\boldsymbol{x} \in \mathbb{R}^n$ and hidden variables $\boldsymbol{y} \in \mathbb{R}^m$, where the conditional distribution is in an exponential family. The joint distribution $p(\boldsymbol{x}, \boldsymbol{y})$ takes the form

$$p(\boldsymbol{x}, \boldsymbol{y}) = \left[\prod_{j=1}^m p(y_j)\right] \left[\prod_{i=1}^n p(x_i|\boldsymbol{y})\right] \tag{7}$$

$$p(x_i|\boldsymbol{y}) = \exp\left\{g(x_i) + t(x_i)\eta_i(\boldsymbol{y}) - a(\eta_i(\boldsymbol{y}))\right\} \tag{8}$$

Suppose we are given an observation $\boldsymbol{x}$ and want to approximately infer the posterior $p(\boldsymbol{y}|\boldsymbol{x})$ by maximizing the variational ELBO, and suppose the y_j's are conditionally independent given $\boldsymbol{x}$, so $p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x}) \prod_{j=1}^m p(y_j|\boldsymbol{x})$. Then the mean-field variational ELBO is

$$\log p(\boldsymbol{x}) \geq \max_{q(\boldsymbol{y}|\boldsymbol{x})} \mathbb{E}[\log p(\boldsymbol{x}, \boldsymbol{y})] + H(\boldsymbol{y}|\boldsymbol{x}) \tag{9}$$

$$\equiv \max_{q(\boldsymbol{y}|\boldsymbol{x})} \sum_{j=1}^m \mathbb{E}\left[\log p(y_j|\boldsymbol{x})\right] + H(y_j|\boldsymbol{x}) \tag{10}$$

$$= \sum_{j=1}^m \max_{q(y_j|\boldsymbol{x})} \mathbb{E}\left[\log p(y_j|\boldsymbol{x})\right] + H(y_j|\boldsymbol{x}) \tag{11}$$

In the second line, $\log p(\boldsymbol{x})$ is constant with respect to $q(\boldsymbol{y}|\boldsymbol{x})$ and can be removed without changing the optimization problem. In this case, the ELBO separates into a sum of terms, each of which involves only a single y_j. This allows us to optimize the ELBO by updating each $q(y_j|\boldsymbol{x})$ independently and in parallel.

In a general DEF, the y_j's are not conditionally independent and the objective does not separate. However, without much manipulation, much of the ELBO does separate: for a single-layer DEF, the ELBO can be written

$$\log p(\boldsymbol{x}) \geq \mathbb{E}[\log p(\boldsymbol{x}, \boldsymbol{y})] + H(\boldsymbol{y}|\boldsymbol{x}) \tag{12}$$

$$= \sum_{i=1}^{n} \mathbb{E}[\log p(x_i|\boldsymbol{y})] + \sum_{j=1}^{m} \mathbb{E}[\log p(y_j)] + H(y_j|\boldsymbol{x}) \tag{13}$$

So only the $\mathbb{E}[\log p(x_i|\boldsymbol{y})]$ terms aren't separable. However, if η_i is an affine function of $\boldsymbol{y}$, so $\eta_i \equiv b_i + \boldsymbol{w}_i \cdot \boldsymbol{y}$ for some $b_i \in \mathbb{R}$ and $\boldsymbol{w}_i \in \mathbb{R}^m$, then each $\mathbb{E}[\log p(x_i|\boldsymbol{y})]$ term can be expanded

$$\mathbb{E}[\log p(x_i|\boldsymbol{y})] = g(x_i) + t(x_i)\mathbb{E}[\eta_i] - \mathbb{E}[a(\eta_i)] \tag{14}$$

$$= g(x_i) + t(x_i)\left(b_i + \boldsymbol{w}_i \cdot \mathbb{E}[\boldsymbol{y}]\right) - \mathbb{E}[a(b_i + \boldsymbol{w}_i \cdot \boldsymbol{y})] \tag{15}$$

From this we can see the only term left preventing the entire ELBO from separating is $\mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x})}[-a(\eta_i(\boldsymbol{y}))]$, a high-dimensional expectation of the non-linear log-partition function. The one thing we know about the log-partition function in exponential families is that it's convex. This suggests we use Jensen's inequality to bound $\mathbb{E}[-a(\eta_i)]$. Note that using Jensen's to bring the expectation over q inside a gives an inequality in the wrong direction because $-a(\eta_i)$ is concave; to get a lower bound, we need to pull an expectation out from the inside of a. The derivation of the ELBO gives a hint on how to do this: recall

$$\log p(x) = \log \int p(x, y)dy \tag{16}$$

$$= \log \int \frac{q(y|x)}{q(y|x)} p(x, y)dy \tag{17}$$

$$= \log \mathbb{E}_{q(y|x)}\left[\frac{p(x, y)}{q(y|x)}\right] \tag{18}$$

$$\geq \mathbb{E}_{q(y|x)}\left[\log \frac{p(x, y)}{q(y|x)}\right] \tag{19}$$

In the same way, we will introduce a variational or auxiliary distribution inside the concave function $-a(\eta)$, then use Jensen's to pull it out. For each i, introduce an auxiliary discrete distribution over m categories $\boldsymbol{\varepsilon}_i \in \Delta^{m-1}$, so

$$\sum_{j=1}^{m} \varepsilon_{ij} = 1 \qquad \varepsilon_{ij} \geq 0 \; \forall j \in [m] \tag{20}$$

Injecting this inside the log-partition function gives

$$\mathbb{E}[-a(b_i + \boldsymbol{w}_i \cdot \boldsymbol{y})] = \mathbb{E}\left[-a\left(b_i + \sum_{j=1}^{m} \varepsilon_{ij} \frac{w_{ij} y_j}{\varepsilon_{ij}}\right)\right] \tag{21}$$

To use Jensen's inequality, we first need to bring b_i inside the sum, which we can do using $b_i = \sum_{j=1}^{m} \varepsilon_{ij} b_i$. This partitions the bias b_i into m parts according to $\boldsymbol{\varepsilon}_i$. However, to get a sufficiently tight bound, we'll need to consider more general splittings: introduce another set of auxiliary parameters $\hat{\boldsymbol{b}}_i \in \mathbb{R}^m$ with the constraint $b_i = \sum_{j=1}^{m} \varepsilon_{ij} \hat{b}_{ij}$. Then

$$\mathbb{E}[-a(b_i + \boldsymbol{w}_i \cdot \boldsymbol{y})] = \mathbb{E}\left[-a\left(\sum_{j=1}^{m} \varepsilon_{ij}\left(\hat{b}_{ij} + \frac{w_{ij} y_j}{\varepsilon_{ij}}\right)\right)\right]$$

$$\geq \sum_{j=1}^{m} \varepsilon_{ij} \mathbb{E}\left[-a\left(\hat{b}_{ij} + \frac{w_{ij} y_j}{\varepsilon_{ij}}\right)\right] \tag{22}$$

Bounding this term for each i separates the entire ELBO into a sum of terms, each of which involves only a single y_j. Plugging this in directly to get a final bound on log-likelihood results in an unwieldy expression, so first we will introduce new notation to simplify the bound.

4.1 CONNECTION WITH FMM

To demonstrate the relation of the above bound and forest mixture models, let us define

$$\hat{\eta}_{ij} \equiv \hat{b}_{ij} + \frac{w_{ij} y_j}{\varepsilon_{ij}} \tag{23}$$

$$\hat{p}(x_i|y_j) \equiv \exp\left\{g(x_i) + t(x_i)\hat{\eta}_{ij} - a(\hat{\eta}_{ij})\right\} \tag{24}$$

Then $\eta_i = \sum_{j=1}^{m} \varepsilon_{ij} \hat{\eta}_{ij}$ and the bound can be rewritten as follows:

$$\mathbb{E}\left[-a\left(\eta_i\right)\right] \geq \sum_{j=1}^{m} \varepsilon_{ij} \mathbb{E}\left[-a\left(\hat{\eta}_{ij}\right)\right] \tag{25}$$

This expression can be used to impose bounds on each $\mathbb{E}[\log p(x_i|\boldsymbol{y})]$:

$$\mathbb{E}[\log p(x_i|\boldsymbol{y})] = g(x_i) + t(x_i)\mathbb{E}[\eta_i] - \mathbb{E}[a(\eta_i)] \tag{26}$$

$$\geq g(x_i) + t(x_i)\mathbb{E}[\eta_i] - \sum_{j=1}^{m} \varepsilon_{ij} \mathbb{E}[a(\hat{\eta}_{ij})] \tag{27}$$

$$= \sum_{j=1}^{m} \varepsilon_{ij}\left(g(x_i) + t(x_i)\mathbb{E}[\hat{\eta}_{ij}] - \mathbb{E}[a(\hat{\eta}_{ij})]\right) \tag{28}$$

$$= \sum_{j=1}^{m} \varepsilon_{ij} \mathbb{E}[\log \hat{p}(x_i|y_j)] \tag{29}$$

input : An observation $\boldsymbol{x} \in \mathbb{R}^n$ and model parameters $W \in \mathbb{R}^{n \times m}$, $\boldsymbol{b} \in \mathbb{R}^n$, $\sigma_y^2 \in \mathbb{R}$ and $\sigma_x^2 \in \mathbb{R}$.
output: The mean-field variational distribution $q(\boldsymbol{y}|\boldsymbol{x}) \equiv \prod_{j=1}^m q(y_j|\boldsymbol{x})$
initialize $(\mu_0)_j$ and $(\sigma_0)_j^2$ for each $j \in [m]$
for $t = 0$ **to** $T-1$ **do**
  for $i = 1$ **to** n **do**
    for $j = 1$ **to** m **do**
      $(\varepsilon_t)_{ij} = \frac{|w_{ij}|(\sigma_t)_j}{\sum_{j'=1}^m |w_{ij'}|(\sigma_t)_{j'}^2}$
      $(\hat{b}_t)_{ij} = (b_i + \sum_{j=1}^m w_{ij}(\mu_t)_j) - \frac{w_{ij}(\mu_t)_j}{(\varepsilon_t)_{ij}}$
    end
  end
  for $j = 1$ **to** m **do**
    $(\mu_{t+1})_j \equiv \frac{\sum_{i=1}^m w_{ij}(x_i - (\hat{b}_t)_{ij})}{\frac{\sigma_x^2}{\sigma_y^2} + \sum_{j=1}^m \frac{w_{ij}^2}{(\varepsilon_t)_{ij}}}$
    $(\sigma_{t+1})_j^2 \equiv \frac{1}{\frac{1}{\sigma_y^2} + \frac{1}{\sigma_x^2}\sum_{j=1}^m \frac{w_{ij}^2}{(\varepsilon_t)_{ij}}}$
  end
end
return $q(y_j|\boldsymbol{x}) = \mathcal{N}((\mu_T)_j, (\sigma_T)_j^2)$ for $j \in [m]$

Algorithm 1: The FM algorithm in the Gaussian case.

Finally, plugging the above expression into the ELBO gives

$$\begin{aligned}\log p(x) &\geq \mathbb{E}[\log p(\boldsymbol{x}, \boldsymbol{y})] + H(\boldsymbol{y}|\boldsymbol{x}) \\ &\geq \sum_{i=1}^n \sum_{j=1}^m \varepsilon_{ij} \mathbb{E}[\log \hat{p}(x_i|y_j)] \\ &\quad - \sum_{j=1}^m D_{KL}(q(\boldsymbol{y}|\boldsymbol{x}) \| p(\boldsymbol{y})) \end{aligned} \tag{30}$$

Comparing (30) with (6) confirms that this bound is *identical* to the ELBO of a forest mixture model with the same $\hat{p}(x_i, y_j)$ and $q(y_j|\boldsymbol{x})$, with $q(e_{ij} = 1|\boldsymbol{x}) = \varepsilon_{ij}$ (so that $\mathbb{E}[e_{ij}] = \varepsilon_{ij}$) and $p(\boldsymbol{e}_i) = q(\boldsymbol{e}_i|\boldsymbol{x})$ (so that the second KL term of the FMM ELBO is zero and disappears entirely). For this reason, we call this bound the *forest mixture bound* (FM bound). Note this bounds the DEF ELBO by the ELBO of each FMM in a large family of FMM's parameterized by $\boldsymbol{\varepsilon} \equiv \{\boldsymbol{\varepsilon}_i\}_{i=1}^n$ and $\hat{\boldsymbol{b}} \equiv \{\hat{\boldsymbol{b}}_i\}_{i=1}^n$.

5 ALGORITHM

To optimize the FM bound, we propose an alternating maximization algorithm: in the first step, update all $q(y_j|\boldsymbol{x})$ in parallel while holding all ε_{ij} and $\hat{b}_{ij}$ fixed; in the second step, update all ε_{ij} and $\hat{b}_{ij}$ in parallel while holding all $q(y_j|\boldsymbol{x})$ fixed. In this section, we will derive the optimal updates for $q(y_j|\boldsymbol{x})$, ε_{ij}, and $\hat{b}_{ij}$ in the case when each x_i and y_j are Gaussian with known variance:

$$p(y_j) = \mathcal{N}(0, \sigma_y^2) \qquad p(x_i|\boldsymbol{y}) = \mathcal{N}(\eta_i(\boldsymbol{y}), \sigma_x^2) \tag{31}$$

We will derive the updates for the auxiliary parameters first since this will help simplify the update for the variational distribution later.

5.1 AUXILIARY PARAMETER UPDATES

Maximizing the FM bound over $\boldsymbol{\varepsilon}$ and $\hat{\boldsymbol{b}}$ is equivalent to maximizing $\mathcal{L}_i \equiv \sum_{j=1}^m \varepsilon_{ij} \mathbb{E}[-a(\hat{\eta}_{ij})]$ over $\boldsymbol{\varepsilon}_i$ and $\hat{\boldsymbol{b}}_i$ for each i, since these are the only terms in the FM bound that depend on $\boldsymbol{\varepsilon}$ and $\hat{\boldsymbol{b}}$. In the Gaussian case, $-a(\hat{\eta}_{ij}) = -\frac{1}{2\sigma_x^2}\hat{\eta}_{ij}^2$ and

$$\mathcal{L}_i = \sum_{j=1}^m \varepsilon_{ij} \mathbb{E}\left[-\frac{1}{2\sigma_x^2}\hat{\eta}_{ij}^2\right] \tag{32}$$

$$= -\frac{1}{2\sigma_x^2}\sum_{j=1}^m \varepsilon_{ij}\left(\text{Var}\left[\hat{\eta}_{ij}\right] + \mathbb{E}\left[\hat{\eta}_{ij}\right]^2\right) \tag{33}$$

$$= -\frac{1}{2\sigma_x^2}\sum_{j=1}^m \frac{w_{ij}^2 \text{Var}[y_j]}{\varepsilon_{ij}} + \varepsilon_{ij}\left(\hat{b}_{ij} + \frac{w_{ij}\mathbb{E}[y_j]}{\varepsilon_{ij}}\right)^2 \tag{34}$$

Theorem 1 *Holding $q(y_j|\boldsymbol{x})$ constant, the choice of $\hat{\boldsymbol{b}}_i$ and $\boldsymbol{\varepsilon}_i$ that maximizes $\mathcal{L}_i$ is $\hat{\boldsymbol{b}}_i = \hat{\boldsymbol{b}}_i^*$ and $\boldsymbol{\varepsilon}_i = \boldsymbol{\varepsilon}_i^*$, where*

$$\hat{b}_{ij}^* = \mathbb{E}[\eta_i] - \frac{w_{ij}\mathbb{E}[y_j]}{\varepsilon_{ij}^*} \quad \varepsilon_{ij}^* = \frac{|w_{ij}|\textit{Std}[y_j]}{\sum_{j'=1}^m |w_{ij'}|\textit{Std}[y_{j'}]} \tag{35}$$

For a proof, see Appendix A. Note that these computations can be parallelized across i and j.

5.2 VARIATIONAL UPDATES

Holding the auxiliary parameters fixed, each variational distribution $q(y_j|\boldsymbol{x})$ can be updated in parallel:

Theorem 2 *For a fixed $\boldsymbol{\varepsilon}$ and $\hat{\boldsymbol{b}}$, the choice for the next variational distribution $q_{t+1}(y_j|\boldsymbol{x})$ that maximizes the FM bound is $q_{t+1}(y_j|\boldsymbol{x}) = \mathcal{N}((\mu_{t+1}^*)_j, (\sigma_{t+1}^*)_j^2)$, where*

$$(\mu_{t+1}^*)_j \equiv \frac{(\boldsymbol{x} - \mathbb{E}_{q_t}[\boldsymbol{\eta}]) \cdot \boldsymbol{w}_j + \mathbb{E}_{q_t}[y_j]\sum_{i=1}^n \frac{w_{ij}^2}{\varepsilon_{ij}}}{\frac{\sigma_x^2}{\sigma_y^2} + \sum_{i=1}^n \frac{w_{ij}^2}{\varepsilon_{ij}}} \tag{36}$$

$$(\sigma_{t+1}^*)_j^2 \equiv \frac{1}{\frac{1}{\sigma_y^2} + \frac{1}{\sigma_x^2}\sum_{i=1}^n \frac{w_{ij}^2}{\varepsilon_{ij}}} \tag{37}$$

For a proof, see Appendix B.

6 DISCUSSION

Tightness We derived the FM bound by using Jensen's inequality to lower bound the ELBO. For a given variational distribution q, the gap between the two bounds is

$$\text{GAP} \equiv \sum_{i=1}^{n} \mathbb{E}[-a(\eta_i)] - \sum_{i=1}^{n}\sum_{j=1}^{m} \varepsilon_{ij}\mathbb{E}[-a(\hat{\eta}_{ij})] \quad (38)$$

In the Gaussian case, for an optimal choice of auxiliary parameters (see Appendix A),

$$\sum_{j=1}^{m} \varepsilon_{ij}\mathbb{E}[-a(\hat{\eta}_{ij})] = -\frac{1}{2\sigma_x^2}\|\boldsymbol{w}_i \odot \text{Std}[\boldsymbol{y}]\|_1^2 - \frac{1}{2\sigma_x^2}\mathbb{E}[\eta_i]^2 \quad (39)$$

$$\mathbb{E}[-a(\eta_i)] = -\frac{1}{2\sigma_x^2}\|\boldsymbol{w}_i \odot \text{Std}[\boldsymbol{y}]\|_2^2 - \frac{1}{2\sigma_x^2}\mathbb{E}[\eta_i]^2 \quad (40)$$

$$\text{GAP} = \frac{1}{2\sigma_x^2}\sum_{i=1}^{n} \|\boldsymbol{w}_i \odot \text{Std}[\boldsymbol{y}]\|_1^2 - \|\boldsymbol{w}_i \odot \text{Std}[\boldsymbol{y}]\|_2^2 \quad (41)$$

Since $\sum_{i=1}^{n}\|\boldsymbol{w}_i\|_1^2 \geq \sum_{i=1}^{n}\|\boldsymbol{w}_i\|_2^2$, the FM bound imposes a stronger regularization on the variance of the variational distribution compared to the variational ELBO. For this reason, the variational distribution q that maximizes the FM bound generally has a smaller variance compared to the variational distribution that maximizes the ELBO.

The FM bound tightly bounds the ELBO when p is a forest model, so that w_{ij} has exactly one non-zero element in the component $j(i)$ corresponding to the parent of x_i. In this case,

$$\|\boldsymbol{w}_i \odot \text{Std}[\boldsymbol{y}]\|_1^2 = w_{ij(i)}^2 \text{Var}[y_{j(i)}] = \|\boldsymbol{w}_i \odot \text{Std}[\boldsymbol{y}]\|_2^2 \quad (42)$$

The bound is also tight when $\text{Var}[\boldsymbol{y}] = 0$, but in this case both the ELBO and the FM bound yield $-\infty$ because of the conditional entropy term $H(\boldsymbol{y}|\boldsymbol{x})$.

Speed of Convergence Let's examine the role of $\boldsymbol{\varepsilon}$ in the update for $q(y_j|\boldsymbol{x})$. If $\sum_{j=1}^{m}\frac{w_{ij}^2}{\varepsilon_{ij}}$ is large, then $\mathbb{E}_{q_{t+1}} \approx \mathbb{E}_{q_t}[y_j]$, and so the FM algorithm makes a small update for y_j. If $\sum_{j=1}^{m}\frac{w_{ij}^2}{\varepsilon_{ij}}$ is small, then $\mathbb{E}_{q_{t+1}}[y_j]$ makes a large step in the direction of the residual $\boldsymbol{x} - \mathbb{E}[\boldsymbol{\eta}]$. In fact, if for some j, $\varepsilon_{ij} = 1$ for all i where w_{ij} is non-zero, then the FM algorithm updates $q(y_j|\boldsymbol{x})$ exactly as CAVI would. In this sense, $\boldsymbol{\varepsilon}$ acts like an attention parameter that selects which $q(y_j|\boldsymbol{x})$ to change and by how much.

If p is a forest model, then the FM algorithm chooses $\boldsymbol{\varepsilon}_i$ to be the one-hot vector indicating the parent of x_i. In this case, the FM algorithm makes coordinate updates for all j in parallel and converges in one iteration. If p is forest-like, i.e., $|\boldsymbol{w}_j| \cdot |\boldsymbol{w}_{j'}|$ is small for $j \neq j'$, then $\boldsymbol{\varepsilon}_i$ is close to one-hot and the FM algorithm makes damped, nearly-CAVI updates in parallel. In this sense, the speed at which the FM algorithm converges depends on how inherently forest-like the model p is.

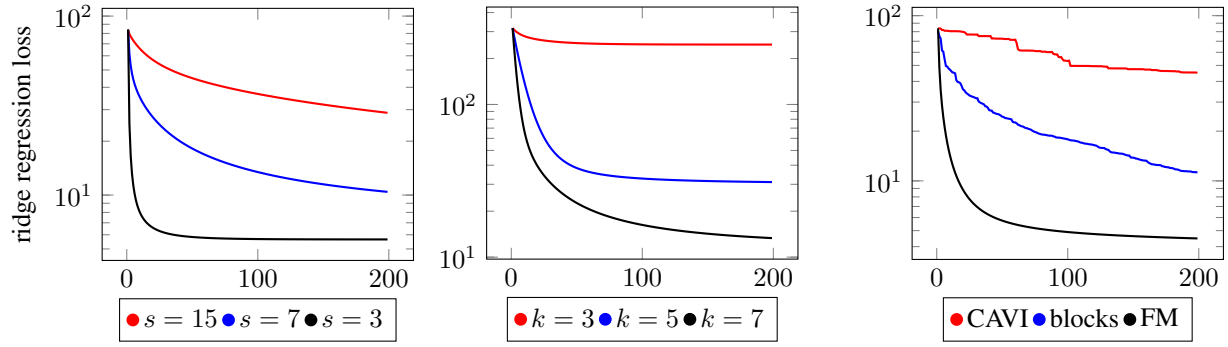
7 EXPERIMENTS

Recall that we derived the FM bound by lower bounding the ELBO. Algorithms that optimize the ELBO like CAVI will generally provide a superior lower bound on log-likelihood compared to the FM algorithm. For a more fair comparison, we can instead measure how quickly these algorithms converge to the optimal mean. In the Gaussian case, optimizing the mean of the mean-field variational distribution is equivalent to minimizing a ridge regression objective:

$$\frac{1}{2\sigma_x^2}\sum_{i=1}^{n}\left(x_i - (b_i + \boldsymbol{w}_i \cdot \mathbb{E}[\boldsymbol{y}])\right)^2 + \frac{1}{2\sigma_y^2}\sum_{j=1}^{m}\mathbb{E}[y_j]^2 \quad (43)$$

To evaluate each algorithm on the ridge regression problem, we must first choose a $\boldsymbol{x}$, $\boldsymbol{b}$, and a set of w_{ij}. All the algorithms we consider in this section are guaranteed to converge to the optimal solution, so we are only interested in comparing how quickly each algorithm converges to that optimal solution. This is measured by recording the objective value achieved by the mean of the variational distribution $\mathbb{E}_{q_t}[\boldsymbol{y}]$ in the ridge regression problem across 200 iterations.

In the first experiment, we choose $\boldsymbol{x}$ to be a vectorized sample from the MNIST dataset, with pixel values scaled to lie in the interval $[-1, 1]$; we choose $\boldsymbol{b}$ to be the average of 1000 randomly chosen MNIST samples; and we construct a synthetic w_{ij} as follows: given an integer window side length s, we construct all possible square $s \times s$ windows of pixels. For windows that overlap the border of the 28×28 MNIST image region, we clip the window so that it lies entirely inside the image region, resulting in a rectangular window. For each window, we add a latent variable y_j to the model and a corresponding $\boldsymbol{w}_j$, where $w_{ij} = 1$ if pixel i lies in window j, and $w_{ij} = 0$ otherwise. The resulting model is more forest-like for smaller choices of s: if $s = 1$, the windows are



(a) FM algorithm on synthetic windows of different sizes

(b) FM algorithm on CNN kernels of different sizes

(c) CAVI, block, and FM comparison

Figure 2: The ridge regression objective over 200 iterations.

disjoint and the graphical model is exactly a forest. Figure 2a demonstrates the rate of convergence of the FM algorithm for various choices of s. As we expect, the FM algorithm converges faster for more forest-like models, i.e., smaller s. Note that the objective value achieved by the optimal solution to the ridge regression problem changes as w_{ij} changes.

The second experiment is similar to the first, except it uses $\boldsymbol{x}$ from the CIFAR-10 dataset, $\boldsymbol{b} = 0$, and instead of uniform windows, uses the first layer kernels from a convolutional neural net trained several times changing only the width of the first layer kernels. Figure 2b demonstrates the FM algorithm converges faster for more forest-like models even using real-world data.

Our last experiment compares the convergence of the FM algorithm with CAVI and block coordinate ascent. Here we choose $\boldsymbol{x}$ and $\boldsymbol{b}$ the same as in the first experiment, but we choose w_{ij} differently to make blocking the latent variables easy: first we partition the 28×28 MNIST image region into 16 regions, each of size 7×7. Then, we construct all possible 7×7 windows (as in the first experiment with $s = 7$), then clip them to fit in the first region. This is repeated for each region. If we block the latent variables according to which region the corresponding windows were clipped to, then the blocks will be conditionally independent, since windows clipped to different regions must be disjoint. Blocking in this way guarantees that the block coordinate algorithm will converge to the optimal solution. Figure 2c compares the rate of convergence for CAVI, block coordinate ascent, and the FM algorithm. The figure shows our block-free method can outperform the block coordinate method, even when the blocking is quite good.

8 CONCLUSION

In this paper we derived a forest mixture bound on the log-likelihood of deep exponential families. This bound gets around the "explaining away effect" by using a set of auxiliary parameters to separate the problem of inference for each latent variable in the same layer, allowing us to make parallel updates. We then made a deep dive into the simple case where all variables are Gaussian: we derived the exact variable updates, then tested the algorithm on both synthetic and real-world data. Our promising results show that fast, parallel inference in deep exponential families is possible without the use of blocks.

A AUXILIARY PARAMETER UPDATES

Proof of **Theorem 1**: First, we will find the optimal choice of $\hat{\boldsymbol{b}}_i$ for any given $\boldsymbol{\varepsilon}_i$. Since $\hat{\boldsymbol{b}}_i$ is constrained by $\sum_{j=1}^m \varepsilon_{ij} \hat{b}_{ij} = b_i$, let's first parameterize $\hat{\boldsymbol{b}}_i$ by a set of unconstrained parameters: let $\boldsymbol{\gamma}_i \in \mathbb{R}^m$ and write

$$\hat{b}_{ij} = b_i - \gamma_{ij} + \boldsymbol{\varepsilon}_i \cdot \boldsymbol{\gamma}_i \tag{44}$$

So for any choice of $\boldsymbol{\gamma}_i$, the constraint $b_i = \sum_{j=1}^m \varepsilon_{ij} \hat{b}_{ij}$ is satisfied. Now we can differentiate the bound with respect to γ_{ij}, set to zero and solve. We will need the following partial derivatives:

$$\frac{\partial \hat{b}_{ij}}{\partial \gamma_{ij}} = -1 + \varepsilon_{ij} \qquad \frac{\partial \hat{b}_{ij'}}{\partial \gamma_{ij}} = \varepsilon_{ij} \; \forall j' \neq j \tag{45}$$

Now setting the partial derivative of $\mathcal{L}_i$ with respect to γ_{ij} to zero,

$$0 = \frac{\partial}{\partial \gamma_{ij}} \mathcal{L}_i = -\frac{1}{\sigma_x^2} \sum_{j'=1}^{m} \varepsilon_{ij'} \mathbb{E}[\hat{\eta}_{ij'}] \frac{\partial \hat{b}_{ij'}}{\partial \gamma_{ij}} \tag{46}$$

$$= \frac{\varepsilon_{ij}}{\sigma_x^2} \left(\mathbb{E}[\hat{\eta}_{ij}] - \sum_{j'=1}^{m} \varepsilon_{ij'} \mathbb{E}[\hat{\eta}_{ij'}] \right) \tag{47}$$

The derivative is zero for all j in particular when the choice of $\hat{b}_{ij}$ makes $\mathbb{E}[\hat{\eta}_{ij}]$ constant across j. We can verify this is satisfied by the choice $\gamma_{ij} = \frac{w_{ij}}{\varepsilon_{ij}} \mathbb{E}[y_j]$, which makes $\hat{b}_{ij} = \hat{b}^*_{ij}$:

$$\mathbb{E}[\hat{\eta}_{ij}] = \mathbb{E}\left[\hat{b}_{ij} + \frac{w_{ij}}{\varepsilon_{ij}} y_j \right] \tag{48}$$

$$= \mathbb{E}\left[\mathbb{E}[\eta_i] - \frac{w_{ij}}{\varepsilon_{ij}} \mathbb{E}[y_j] + \frac{w_{ij}}{\varepsilon_{ij}} y_j \right] \tag{49}$$

$$= \mathbb{E}[\eta_i] \tag{50}$$

Plugging this choice into $\mathcal{L}_i$ yields

$$\mathcal{L}_i = -\frac{1}{2\sigma_x^2} \sum_{j=1}^{m} \left(\frac{w_{ij}^2 \text{Var}[y_j]}{\varepsilon_{ij}} + \varepsilon_{ij} \mathbb{E}[\eta_i]^2 \right) \tag{51}$$

$$= -\frac{1}{2\sigma_x^2} \left(\sum_{j=1}^{m} \frac{w_{ij}^2 \text{Var}[y_j]}{\varepsilon_{ij}} \right) - \frac{1}{2\sigma_x^2} \mathbb{E}[\eta_i]^2 \tag{52}$$

Now let's try to find the optimal choice of ε_{ij}. Since ε_{ij} is constrained by $\varepsilon_i \in \Delta^{m-1}$, we'll also parameterize ε_{ij} by a set of unconstrained parameters $\boldsymbol{\tau}_i \in \mathbb{R}^m$:

$$\varepsilon_{ij} = \exp\{\tau_{ij}\} / \sum_{j'=1}^{m} \exp\{\tau_{ij'}\} \tag{53}$$

We will need the following partial derivatives:

$$\frac{\partial \varepsilon_{ij}}{\partial \tau_{ij}} = \varepsilon_{ij}(1 - \varepsilon_{ij}) \quad \frac{\partial \varepsilon_{ij'}}{\partial \tau_{ij}} = -\varepsilon_{ij} \varepsilon_{ij'} \; \forall j' \neq j \tag{54}$$

Now setting the partial derivative of $\mathcal{L}_i$ with respect to τ_{ij} to zero,

$$0 = \frac{\partial}{\partial \tau_{ij}} \mathcal{L}_i = \frac{1}{2\sigma_x^2} \sum_{j'=1}^{m} \frac{w_{ij'}^2 \text{Var}[y_{j'}]}{\varepsilon_{ij'}^2} \frac{\partial \varepsilon_{ij'}}{\partial \tau_{ij}} \tag{55}$$

$$= \frac{1}{2\sigma_x^2} \sum_{j'=1}^{m} \text{Var}[\hat{\eta}_{ij'}] \frac{\partial \varepsilon_{ij'}}{\partial \tau_{ij}} \tag{56}$$

$$= \frac{\varepsilon_{ij}}{2\sigma_x^2} \left(\text{Var}[\hat{\eta}_{ij}] - \sum_{j'=1}^{m} \varepsilon_{ij'} \text{Var}[\hat{\eta}_{ij'}] \right) \tag{57}$$

The derivative is zero for all j in particular when the choice of ε_{ij} makes $\text{Var}[\hat{\eta}_{ij}]$ constant across j. We can verify this is satisfied by the choice $\tau_{ij} = \log |w_{ij} \text{Std}[y_j]|$, which makes $\varepsilon_{ij} = \varepsilon^*_{ij}$:

$$\text{Var}[\hat{\eta}_{ij}] = \frac{w_{ij}^2 \text{Var}[y_j]}{\varepsilon_{ij}^2} \tag{58}$$

$$= \frac{w_{ij}^2 \text{Var}[y_j]}{w_{ij}^2 \text{Var}[y_j] / \left(\sum_{j'=1}^{m} |w_{ij'}| \text{Std}[y_{j'}] \right)^2} \tag{59}$$

$$= \|\boldsymbol{w}_i \odot \text{Std}[\boldsymbol{y}]\|_1^2 \tag{60}$$

Plugging this choice into $\mathcal{L}_i$ yields

$$\mathcal{L}_i = -\frac{1}{2\sigma_x^2} \sum_{j=1}^{m} |w_{ij}| \text{Std}[y_j] \left(\sum_{j'=1}^{m} |w_{ij}| \text{Std}[y_j] \right) - \frac{1}{2\sigma_x^2} \mathbb{E}[\eta_i]^2 \tag{61}$$

$$= -\frac{1}{2\sigma_x^2} \left(\sum_{j=1}^{m} |w_{ij}| \text{Std}[y_j] \right)^2 - \frac{1}{2\sigma_x^2} \mathbb{E}[\eta_i]^2 \tag{62}$$

$$= -\frac{1}{2\sigma_x^2} \|\boldsymbol{w}_i \odot \text{Std}[\boldsymbol{y}]\|_1^2 - \frac{1}{2\sigma_x^2} \mathbb{E}[\eta_i]^2 \tag{63}$$

B VARIATIONAL UPDATES

Proof of **Theorem 2**: First, note that for any DEF, the optimal update equation is as follows:

$$\log q_{t+1}(y_j | \boldsymbol{x}) \propto \log p(y_j) + \sum_{i=1}^{n} \varepsilon_{ij} \log \hat{p}_t(x_i | y_j) \tag{64}$$

In the Gaussian case, we have

$$\log p(y_j) \propto -\frac{1}{2\sigma_y^2} y_j^2 \tag{65}$$

$$\log \hat{p}(x_i|y_j) \propto -\frac{1}{2\sigma_x^2}(x_i - \hat{\eta}_{ij})^2 \tag{66}$$

$$\propto -\frac{1}{2\sigma_x^2}\left(x_i - \left(\hat{b}_{ij} + \frac{w_{ij}}{\varepsilon_{ij}} y_j\right)\right)^2 \tag{67}$$

$$\propto \frac{1}{\sigma_x^2}\frac{(x_i - \hat{b}_{ij})w_{ij}}{\varepsilon_{ij}} y_j - \frac{1}{2\sigma_x^2}\frac{w_{ij}^2}{\varepsilon_{ij}^2} y_j^2 \tag{68}$$

Plugging this in yields

$$\log q(y_j|\boldsymbol{x}) \propto \frac{1}{\sigma_x^2}\left((\boldsymbol{x} - \hat{\boldsymbol{b}}_j)\cdot \boldsymbol{w}_j\right) y_j - \frac{1}{2} y_j^2 \left(\frac{1}{\sigma_y^2} + \frac{1}{\sigma_x^2}\sum_{i=1}^n \frac{w_{ij}^2}{\varepsilon_{ij}}\right) \tag{69}$$

$$\propto -\frac{1}{\sigma_x^2}\left((\boldsymbol{x} - \hat{\boldsymbol{b}}_j)\cdot \boldsymbol{w}_j\right) y_j - \frac{1}{2(\sigma_{t+1}^*)_j^2} y_j^2 \tag{70}$$

$$\propto -\frac{1}{2(\sigma_{t+1}^*)_j^2}\left(y_j - \frac{\frac{1}{\sigma_x^2}(\boldsymbol{x} - \hat{\boldsymbol{b}}_j)\cdot \boldsymbol{w}_j}{\frac{1}{\sigma_y^2} + \frac{1}{\sigma_x^2}\sum_{i=1}^n \frac{w_{ij}^2}{\varepsilon_{ij}}}\right)^2 \tag{71}$$

$$\propto -\frac{1}{2(\sigma_{t+1}^*)_j^2}\left(y_j - \frac{(\boldsymbol{x} - \hat{\boldsymbol{b}}_j)\cdot \boldsymbol{w}_j}{\frac{\sigma_x^2}{\sigma_y^2} + \sum_{i=1}^n \frac{w_{ij}^2}{\varepsilon_{ij}}}\right)^2 \tag{72}$$

After substituting $\hat{b}_{ij} = \mathbb{E}[\eta_i] - \frac{w_{ij}}{\varepsilon_{ij}}\mathbb{E}_{q_t}[y_j]$ and rearranging, we get $\log q_{t+1}(y_j|\boldsymbol{x}) \propto \mathcal{N}((\mu_{t+1}^*)_j, (\sigma_{t+1}^*)_j^2)$.

C EXTENSION TO DEEP MODELS

A DEF model with observed variables $\boldsymbol{y}^{(0)} \in \mathbb{R}^{m_0}$ and L layers of latent variables $\{\boldsymbol{y}^{(\ell)}\}_{\ell=1}^L$ with $\boldsymbol{y}^{(\ell)} \in \mathbb{R}^{m_\ell}$ has joint distribution

$$p(\{\boldsymbol{y}^{(\ell)}\}_{\ell=0}^L) = \left[\prod_{\ell=0}^{L-1}\prod_{i=1}^{m_\ell} p(y_i^{(\ell)}|\boldsymbol{y}^{(\ell+1)})\right]\left[\prod_{i=1}^{m_L} p(y_i^{(L)})\right] \tag{73}$$

$$p(y_i^{(\ell)}|\boldsymbol{y}^{(\ell+1)}) = \exp\left\{g(y_i^{(\ell)}) + t(y_i^{(\ell)})\eta_i^{(\ell)} - a(\eta_i^{(\ell)})\right\} \tag{74}$$

$$\eta_i^{(\ell)} \equiv b_i^{(\ell)} + \boldsymbol{w}_i^{(\ell)}\cdot \boldsymbol{y}^{(\ell+1)} \tag{75}$$

The ELBO for this model is

$$\log p(y^{(0)}) \geq \sum_{i=1}^{m_0} \mathbb{E}[\log p(y_i^{(0)}|\boldsymbol{y}^{(1)})] \tag{76}$$

$$+ \sum_{\ell=1}^{L-1}\sum_{i=1}^{m_\ell} \mathbb{E}[\log p(y_i^{(\ell)}|\boldsymbol{y}^{(\ell+1)})] + \underset{q}{H}(y_i^{(\ell)}|\boldsymbol{y}^{(0)}) \tag{77}$$

$$+ \sum_{i=1}^{m_L} p(y_i^{(L)}) + \underset{q}{H}(y_i^{(L)}|\boldsymbol{y}^{(0)}) \tag{78}$$

For each $\ell \in \{0, \ldots, L-1\}$, introduce the auxiliary parameters $\{\boldsymbol{\varepsilon}_i^{(\ell)}\}_{i=1}^{m_\ell}$ and $\{\hat{\boldsymbol{b}}_i^{(\ell)}\}_{i=1}^{m_\ell}$, with $\boldsymbol{\varepsilon}_i^{(\ell)} \in \Delta^{m_{\ell+1}-1}$ and $\hat{\boldsymbol{b}}_i^{(\ell)} \in \mathbb{R}^{m_{\ell+1}}$ constrained by $b_i^{(\ell)} = \sum_{j=1}^{m_{\ell+1}} \varepsilon_{ij}^{(\ell)} \hat{b}_{ij}^{(\ell)}$. For all $\ell \in \{0, \ldots, L-1\}$, $i \in [m_\ell]$, and $j \in [m_{\ell+1}]$, define

$$\hat{\eta}_{ij}^{(\ell)} \equiv \hat{b}_{ij}^{(\ell)} + \frac{w_{ij}^{(\ell)}}{\varepsilon_{ij}^{(\ell)}} y_j^{(\ell+1)} \tag{79}$$

$$\hat{p}(y_i^{(\ell)}|y_j^{(\ell+1)}) \equiv \exp\{g(y_i^{(\ell)}) + t(y_i^{(\ell)})\hat{\eta}_{ij}^{(\ell)} - a(\hat{\eta}_{ij}^{(\ell)})\} \tag{80}$$

Then by (25),

$$\mathbb{E}[\log p(y_i^{(\ell)}|\boldsymbol{y}^{(\ell+1)})] \geq \sum_{j=1}^{m_{\ell+1}} \varepsilon_{ij}^{(\ell)} \mathbb{E}[\log \hat{p}(y_i^{(\ell)}|y_j^{(\ell+1)})] \tag{81}$$

Plugging this into the ELBO yields

$$\log p(\boldsymbol{y}^{(0)}) \geq \sum_{i=1}^{m_0}\sum_{j=1}^{m_1} \varepsilon_{ij}^{(\ell)} \mathbb{E}[\log \hat{p}(y_i^{(0)}|y_j^{(1)})] \tag{82}$$

$$+ \sum_{\ell=1}^{L-1}\sum_{i=1}^{m_\ell}\sum_{j=1}^{m_{\ell+1}} \varepsilon_{ij}^{(\ell)} \mathbb{E}[\log \hat{p}(y_i^{(\ell)}|y_j^{(\ell+1)})] + \underset{q}{H}(y_i^{(\ell)}|\boldsymbol{y}^{(0)}) \tag{83}$$

$$+ \sum_{i=1}^{m_L} p(y_i^{(L)}) + \underset{q}{H}(y_i^{(L)}|\boldsymbol{y}^{(0)}) \tag{84}$$

This objective separates as a sum of terms, each of which involves no more than one latent variable in the same layer. This allows any group of variables forming an independent set in the model graph to be updated in parallel, the same as for undirected models.

References

[Baqué et al., 2016] Baqué, P., Bagautdinov, T., Fleuret, F., and Fua, P. (2016). Principled parallel mean-field inference for discrete random fields. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5848–5857.

[Blei et al., 2017] Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

[Chow and Liu, 1968] Chow, C. and Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467.

[Johnson et al., 2013] Johnson, M., Saunderson, J., and Willsky, A. (2013). Analyzing hogwild parallel Gaussian Gibbs sampling. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 2715–2723. Curran Associates, Inc.

[Kingma and Welling, 2013] Kingma, D. P. and Welling, M. (2013). Auto-Encoding Variational Bayes. *ArXiv e-prints*.

[Ranganath et al., 2015] Ranganath, R., Tang, L., Charlin, L., and Blei, D. (2015). Deep exponential families. In Lebanon, G. and Vishwanathan, S. V. N., editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 762–771, San Diego, California, USA. PMLR.

[Recht et al., 2011] Recht, B., Re, C., Wright, S., and Niu, F. (2011). Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 24*, pages 693–701. Curran Associates, Inc.

[Sa et al., 2016] Sa, C. D., Re, C., and Olukotun, K. (2016). Ensuring rapid mixing and low bias for asynchronous Gibbs sampling. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1567–1576, New York, New York, USA. PMLR.

[Salakhutdinov and Hinton, 2009] Salakhutdinov, R. and Hinton, G. (2009). Deep Boltzmann machines. In van Dyk, D. and Welling, M., editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 448–455, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA. PMLR.

[Singh et al., 2010] Singh, S., Subramanya, A., Pereira, F., and McCallum, A. (2010). Distributed MAP inference for undirected graphical models. In *Neural Information Processing Systems (NIPS) Workshop on Learning on Cores, Clusters, and Clouds (LCCC)*.

[Singh et al., 2017] Singh, S. S., Lindsten, F., and Moulines, E. (2017). Blocking strategies and stability of particle Gibbs samplers. *Biometrika*, 104(4):953–969.

[Sontag and Jaakkola, 2009] Sontag, D. and Jaakkola, T. (2009). Tree block coordinate descent for MAP in graphical models. In van Dyk, D. and Welling, M., editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 544–551, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA. PMLR.

[Terenin et al., 2015] Terenin, A., Simpson, D., and Draper, D. (2015). Asynchronous Gibbs sampling. *ArXiv e-prints*.

[Wainwright et al., 2005] Wainwright, M. J., Jaakkola, T. S., and Willsky, A. S. (2005). A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51(7):2313–2335.

[Wang and Banerjee, 2014] Wang, H. and Banerjee, A. (2014). Randomized block coordinate descent for online and stochastic optimization. *CoRR*, abs/1407.0107.

[Zhao et al., 2014] Zhao, T., Yu, M., Wang, Y., Arora, R., and Liu, H. (2014). Accelerated mini-batch randomized block coordinate descent method. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 3329–3337. Curran Associates, Inc.