Stochastic Spectral Descent for Discrete Graphical Models

David Carlson, Ya-Ping Hsieh, Edo Collins, Lawrence Carin, *Fellow, IEEE,* Volkan Cevher, *Senior member, IEEE*

***Abstract*—Interest in deep probabilistic graphical models has increased in recent years, due to their state-of-the-art performance on many machine learning applications. Such models are typically trained with the stochastic gradient method, which can take a significant number of iterations to converge. Since the computational cost of gradient estimation is prohibitive even for modestly-sized models, training becomes slow and practically-usable models are kept small. In this paper we propose a new, largely tuning-free algorithm to address this problem. Our approach derives novel majorization bounds based on the Schatten-∞ norm. Intriguingly, the minimizers of these bounds can be interpreted as gradient methods in a non-Euclidean space. We thus propose using a stochastic gradient method in non-Euclidean space. We both provide simple conditions under which our algorithm is guaranteed to converge, and demonstrate empirically that our algorithm leads to dramatically faster training and improved predictive ability compared to stochastic gradient descent for both directed and undirected graphical models.**

I. Introduction

Graphical models have become increasingly popular as a probabilistic approach to learning, allowing control over model complexity with modular extensions into "deep" models. Resulting models have already produced state-of-the-art performance on various classification tasks [1]. For instance, Markov Random Fields (MRFs) have been successfully applied to a wide number of data types. Examples include binary images with the Restricted Boltzmann Machine (RBM) [2] and count modeling with the Replicated Softmax [3]. Deep extensions of MRFs include the Deep RBM [4]. Directed graphical models, also known as Bayesian Networks or Belief Nets (BN), have also found increasing popularity.

Unfortunately, training elaborate deep models is notoriously hard. Since the optimization objective is typically non-convex, even asserting local optimality is difficult. In spite of often having differentiable objective functions, computation of the gradient scales poorly with the dimensionality of the model parameters, rendering exact gradient computation intractable for even modest-sized models. For instance, the recent rise of interest in BNs is due to the tractability of approximate methods, including variational methods [5] and recognition models [6, 7]. However, the computational bottleneck still remains the gradient estimation, where Markov Chain Monte Carlo (MCMC) methods, including Contrastive Divergence (CD) methods [2], are used in both MRFs and BNs.

As a result, learning schemes typically proceed by using classical stochastic gradient descent (SGD), which is guaranteed to converge to a stationary point, or with methods that attempt to locally adapt to the Euclidean geometry, including ADAgrad [8] and RMSprop [9]. These algorithms may suffer from diminishing returns in training performance, where minor improvements require orders of magnitude more training iterations. Combined with the high cost of gradient estimation, this poses a hindrance to the adoption of large-scale probabilistic models for many practical applications.

In this paper we motivate a novel algorithm that operates on a *non-Euclidean* geometry. Central to our algorithm is a new class of global majorization bounds for objective functions in probabilistic graphical models (or, more generally, energy based models), which, from the majorization-minimization perspective, suggests searching for the steepest descent direction with respect to the Schatten-∞ norm. We provide numerical evidence to demonstrate major performance improvements over previous methods, suggesting the fact that non-Euclidean geometry is preferred over Euclidean geometry in probabilistic graphical models.

The organization of this paper is as follows. In Section II, we first show how to adapt to the global geometry of the objective function in a non-Euclidean space, derive novel majorization bounds that use the Schatten-∞ norm (alternatively known as the spectral or matrix-2 norm), and show the conditions under which the algorithm is guaranteed to converge. In Section III we show that viewing BNs as a Boltzmann energy distribution allows a joint framework for analyzing both discrete MRFs and BNs. Using this framework, in Section IV we propose the Stochastic Spectral Descent algorithm (SSD), which employs relatively inexpensive nonlinear operations (as compared to the gradient estimation) on the gradient to minimize the majorization bound. Using the convergence theory in Section II-C, we propose a method that adapts the minibatch size to approximate convergence conditions. Empirically, we show in Section V that SSD not only provides up to an order of magnitude speed-up compared to other approaches, but also leads to state-of-the-art performance for similarly-sized models due to improved optimization.

The main contributions of the paper include the rigorous

D. Carlson is with the Department of Statistics and Grossman Center for the Statistics of Mind, Columbia University, New York, NY.
E-mail: david.edwin.carlson@gmail.com

L. Carin is with the Department of Electrical and Computer Engineering, Duke University, Durham, NC.
E-mail: lcarin@duke.edu

E. Collins, Y.P. Hsieh and V. Cevher are with the Laboratory for Information and Inference Systems (LIONS), École Polytechnique Fédérale de Lausanne (EPFL), Switzerland.
E-mail: {edo.collins, ya-ping.hsieh, volkan.cevher}@epfl.ch

analysis of SSD as generalized gradient descent, the extension of the algorithm to directed models, as well as replicated-softmax models, and finally a series of experiments that demonstrate the algorithm works well in practice, leading to state-of-the-art performance for directed models that rivals performance of undirected models.

A. Notation and Preliminaries

Bold lower-case letters represent vectors, and bold upper-case letters represent matrices. $\langle\cdot,\cdot\rangle$ denotes an inner product, and $\boldsymbol{x}\odot\boldsymbol{y}$ denotes element-wise (Hadamard) multiplication. $\mathbf{W}_{m,\cdot}$ denotes the m^{th} row of a matrix. The ℓ_p norm for a vector $\boldsymbol{x}$ is defined $||\boldsymbol{x}||_p = (\sum_n |x_n|^p)^{1/p}$. Letting $\boldsymbol{\lambda} = (\lambda_1,\ldots,\lambda_K)^T$ be the vector of singular values of a matrix $\mathbf{X}\in\mathbb{R}^{M\times N}$, where $K=\min(M,N)$, then the Schatten p-norm is defined as $||\mathbf{X}||_{S_p} = (\sum_{n=1}^K |\lambda_n|^p)^{1/p}$ and $||\mathbf{X}||_{S_\infty} = \max\{|\lambda_1|,\ldots,|\lambda_K|\}$. The dual norm is written as $||\cdot||_*$. The Frobenius norm is $||\cdot||_F$.

Bounds written as only a subset of the parameters consider all unwritten parameters to be held constant. The sigmoid function $\sigma(\cdot)$ is given by $\sigma(x)=\frac{\exp(x)}{1+\exp(x)}$. The softmax function softmax$(\cdot)$ is defined for $\boldsymbol{x}\in\mathbb{R}^N$ as $\text{softmax}(\boldsymbol{x}) = \frac{1}{\sum_{n=1}^N \exp(x_n)}(\exp(x_1),\ldots,\exp(x_N))^T$.

II. Non-Euclidean Geometry of Minimization Problems

A. The Generalized Gradient Descent

Consider the minimization of a function $F(\boldsymbol{x})$ with Lipschitz gradient in the Euclidean norm:

$$||\nabla F(\boldsymbol{x}) - \nabla F(\boldsymbol{y})||_2 \le L_2||\boldsymbol{x}-\boldsymbol{y}||_2$$

where $L_2>0$ is the Lipschitz constant. It is well-known that such function admits a global majorization bound

$$F(\boldsymbol{y}) \le F(\boldsymbol{x}) + \langle\nabla F(\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle + \frac{L_2}{2}||\boldsymbol{y}-\boldsymbol{x}||_2^2. \quad (1)$$

The usual gradient descent aims at minimizing the above majorization bound, which results in the iteration

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L_2}\nabla F(\boldsymbol{x}_k).$$

It is well known that gradient descent generates a sequence of points $\{x_k\}$'s with $\nabla F(x_k)\to 0$ [10]. In the case where $F(\cdot)$ is also convex, one can show convergence to the global minimum.

Now, suppose that instead of the Euclidean norm, $\nabla F(\cdot)$ is Lipschitz with respect to a general norm:

$$||\nabla F(\boldsymbol{x}) - \nabla F(\boldsymbol{y})||_* \le L||\boldsymbol{x}-\boldsymbol{y}||.$$

Then the majorization bound (1) is still valid, with the Euclidean norm replaced by a general norm $||\cdot||$ and L_2 by L (see appendix of [11] for a proof). Based on this perspective, consider the iteration rule based on minimizing the right-hand side of (1) at each stage:

$$\boldsymbol{x}_{k+1} \in \arg\min_{\boldsymbol{y}} F(\boldsymbol{x}_k) + \langle\nabla F(\boldsymbol{x}_k), \boldsymbol{y}-\boldsymbol{x}_k\rangle + \frac{L}{2}||\boldsymbol{y}-\boldsymbol{x}_k||^2. \quad (2)$$

Define the $\#$-operator [12] as

$$\boldsymbol{s}^\# \in \arg\max_{\boldsymbol{x}} \left\{\langle\boldsymbol{s},\boldsymbol{x}\rangle - \frac{1}{2}||\boldsymbol{x}||^2\right\}. \quad (3)$$

Then one can show (see [11] or [13]) that the resulting iteration is given by

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\left[\nabla F(\boldsymbol{x}_k)\right]^\#. \quad (4)$$

In the following we call the iteration (4) *generalized gradient descent* (GGD). The term “gradient descent” (GD) is reserved for Euclidean norm. Notice that neither (2) or (3) need the minimizers to be unique. In those cases, the iteration (4) is to be understood as picking an arbitrary element in $[\nabla F(\boldsymbol{x}_k)]^\#$.

Though resembling the mirror descent in its proximal iteration with Bregman distance form [14], (3) is **not** an instance of mirror descent, but instead is closer to the classical gradient descent in spirit. One can see that our iteration (4) generates a *monotonically decreasing* sequence in function values, a characteristic shared by classical gradient descent but not by mirror descent. Another way of seeing the difference is from the perspective of monotone operator theory. First, by a result in the monotone operator theory (Lemma 1 of [15]), the set of minimizers of the mirror descent iteration must either be unique or empty. However, the minimizer of (3) is not necessarily unique, thereby ruling out the possibility of (3) corresponding to proximal iteration with **any** Bregman distance regularizer. Consequently, (3) is not an instance of mirror descent in the classical sense.

Generalizing proximal methods to Banach spaces is, for the time being, a highly nontrivial work. Specifically, the iteration (3) corresponds to a proximal-type algorithm in the ℓ_∞ geometry. Although there exist some initial works of proximal algorithm in the Banach spaces (see, e.g., [16]), they usually only apply to ℓ_p spaces with $1<p<\infty$, and ℓ_∞ is excluded due to its ill-behaved geometry (e.g., it is not uniformly convex). From this perspective, we prefer not to link (3) with any proximal-type algorithms (including mirror descent), except for classical gradient descent.

B. Motivations for Non-Euclidean Geometry

There is no reason *a priori* that GGD is better than the GD. To compare the two on the same footing, we use the following convergence rate that is proven in [11]:

$$F(x_k) - F(x^*) \le \text{const.} \times \frac{LR^2}{k} \quad (5)$$

where $R = ||x_0 - x^*||$ is the radius from the initial point to the optimal point, measured with respect to the norm $||\cdot||$. Notice that there is a tradeoff between the involved parameters: from (1), using a larger norm (such as 2-norm) may result in a better Lipschitz constant, while the corresponding iterate (4) leads to a worse radius dependency. Remarkably, an example in [11] shows that, for certain functions, the product $L_\infty R_\infty^2$ can be much smaller than $L_2R_2^2$, where L_∞ and R_∞ denote the Lipschitz constant and radius measured with respect to ℓ_∞ norm, and similar for L_2 and R_2. In a sense, a function with the above property is said to exhibit "favorable geometry" in Schatten-∞ norm.

Inspired by the above arguments, in this paper we aim to explore the "favorable geometry" of loss functions appearing in the deep discrete graphical models. However, two obstacles present themselves along the way:

- How can we find a majorization bound (2) for loss functions in deep discrete graphical models?
- Computing the gradient of loss functions is in practice a computationally prohibitive task.

In [13], the first problem is solved for Restricted Boltzmann Machines (RBM) as follows: the authors showed that, although the loss function is non-convex, it is possible to treat the "data term" (see subsequent sections) and the partition function term separately to obtain a global majorization bound that is naturally expressed in Schatten-∞ norm. Furthermore, not only does the bound (2) continue to hold, it is also empirically observed that doing GGD with respect to Schatten-∞ norm outperforms all state-of-the-art learning algorithms for RBM, which all lie Euclidean norms. This brings out the important message:

> *The loss function of an RBM favors the global geometry induced by the Schatten-∞ norm rather than the Euclidean norm.*

In this paper, we will extend this observation to several important deep discrete graphical models and also show empirically that the Schatten-∞ norm is indeed a superior choice over Euclidean norm for bounding the loss functions in the analyzed graphical models. In our analysis, L_F and L_{S^∞} are the same for the matrix parameters in these models[1], leading to a comparison on the radius R_F versus R_{S^∞}. We note that $R^2_{S^\infty} \leq R^2_F \leq \text{rank}(\mathbf{X}_0 - \mathbf{X}^*)R^2_{S^\infty}$. The optimization radius for the Schatten-∞ norm tends to scale better with the dimensionality of the model, which is demonstrated empirically in the experiments.

The common solution to the second problem is to consider only a stochastic estimate of the gradient, which is much cheaper than obtaining the exact gradient. The price to pay, however, is that now the algorithms become sensitive to the inexactness (or the so-called *noise level*) of gradient estimate. In order to address this problem, in the next subsection we derive general conditions under which the convergence is ensured. In practice, however, in certain cases the derived conditions may be too conservative and may cause the algorithm to converge slowly. As a result, we will also consider heuristic settings of stepsizes in our experiments.

C. The Generalized Stochastic Gradient Descent

In this subsection we derive the basic conditions for our algorithms to converge. We present the analysis in full generality; to conclude the convergence of our algorithms, it amounts to substituting the corresponding inner products, constants and norms.

Consider the minimization of a general (non-convex) function

$$\min_{\boldsymbol{x}} F(\boldsymbol{x})$$

with a global majorization bound

$$F(\boldsymbol{y}) \leq F(\boldsymbol{x}) + \langle \nabla F(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + C||\boldsymbol{y} - \boldsymbol{x}||^2 \tag{6}$$

where $C > 0$ is a constant. We assume that $\min_{\boldsymbol{x}} F(x) > -\infty$. Instead of exact gradient, we only have access to a noisy first order oracle information:

$$G(\boldsymbol{x}) = \nabla F(\boldsymbol{x}) + w(\boldsymbol{x})$$

where $w(\boldsymbol{x})$ is random and satisfies $\mathbb{E}w(\boldsymbol{x}) = 0$. We propose the following Generalized Stochastic Gradient Descent (Generalized SGD) iteration:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - h_k G(\boldsymbol{x}_k)^\# \tag{7}$$

where the h_k's are stepsizes. Notice that each $\boldsymbol{x}_k$ is a random variable depending on realizations of $\{G(\boldsymbol{x}_i)\}_{i=0}^{k-1}$ or, equivalently, $\{w(\boldsymbol{x}_i)\}_{i=0}^{k-1}$. Denote $F_k = F(\boldsymbol{x}_k)$, $\nabla F_k = \nabla F(\boldsymbol{x}_k)$, $G_k = G(\boldsymbol{x}_k)$ and $w_k = w(\boldsymbol{x}_k)$.

In the following, a key assumption we shall make is that, for each k, there exists $t_k > 0$ such that $||w_k||_* \leq t_k ||\nabla F_k||_*$. We note that this assumption is rather unconventional, since in most classical works on stochastic optimization the noise is usually assumed to be independent and identically distributed (iid). However, in classical problems, the noise is usually due to the environment and must be incorporated into the problem formulation. In our case, the noise arises due to the gradient estimate, which we can control. Under our assumption, we are allowed to keep constant stepsize across iterations and still obtain convergence. This is in sharp contrast to the iid noise case, where one must decrease the stepsize along a carefully chosen sequence, so as to guarantee convergence (see, e.g., [18]).

We now show that the iteration (7) produces a sequence $\{x_k\}$'s whose gradients converge to 0, hence in a sense we can reach to a stationary point in expectation, if h_k's are chosen properly according to the noise level.

Theorem II.1. *Suppose that there exists a ρ such that $\frac{(1-2t_k)^2}{4C(1-t_k)^2} \geq \rho > 0$ for all k, and suppose that the stepsizes h_k's satisfy*

$$0 < h_k \leq \frac{1}{\sqrt{C}} \left[\sqrt{\frac{(1-2t_k)^2}{4C(1-t_k)^2} - \rho} + \frac{1-2t_k}{2\sqrt{C}(1-t_k)} \right]. \tag{8}$$

Then $||\mathbb{E}\nabla F_k||^2_* \to 0$ as $k \to \infty$. Moreover, let ϵ be a given precision: $||\nabla F(\boldsymbol{x})||_* < \epsilon$. Then the required number of iteration is no greater than $\mathcal{O}(\frac{1}{\epsilon^2})$.

If it happens that $F(\cdot)$ is also convex or our starting point lies in a convex region of $F(\cdot)$, then it is also possible to derive the convergence rate in function value. Denote

$$q(h, C, t) = h(1 - Ch) - \frac{t}{1-t}h.$$

Theorem II.2. *Let $F(\cdot)$ be convex and satisfying (6). Suppose that there exists a ρ such that $q(h_k, C, t_k)(1-t_k)^2 \geq \rho > 0$*

[1] These quantities are dependent on the L_∞ and L_2 for the log-sum-exp function. In [17], we show that L_∞ is no worse than $\Omega(\log N)$ compared to L_2, further motivating the use of L_{S^∞}

for all k. Then

$$\mathbb{E}F_k - F^* \leq \frac{R^2}{\frac{1}{C} + 2k\rho} = \mathcal{O}\left(\frac{1}{k}\right) \tag{9}$$

where F^* is the optimal function value and

$$R \triangleq \max_{\boldsymbol{x}:F(\boldsymbol{x})\leq F(\boldsymbol{x}_0)} \min_{\boldsymbol{x}^* \in \mathcal{X}^*} ||\boldsymbol{x}^* - \boldsymbol{x}||,$$

$\mathcal{X}^*$ the set of minimizers.

The proofs can be found in Appendix A.

An important observation from our theorems is that the stepsize can be fixed to a constant. For example, let us fix t_k's to $\frac{1}{3}$. In view of (8), it suffices to set $\rho = \frac{1}{16C}$ and $h_k = \frac{1}{4C}$. Similar conclusions hold for *Theorem II.2*.

Notice that, although we derive our theorems based on the deterministic relation $||w_k||_* \leq t_k||\nabla F_k||_*$, our theorems can be readily generalized to the case where this relation holds with high probability. Combining with the observation in the last paragraph, we see that, in order to guarantee a high probability convergence, it suffices to set $t_k = \frac{||\nabla F_k||_*}{||w_k||_*}$ to a large constant. This can be achieved through setting a constant SNR in the sampling procedures, which we approximate in our experiments.

We note that there are two ways that the SNR ratio is controlled for these problems: the number of samples used in a Monte Carlo Integration step and the size of the minibatch. We primarily control the SNR by adapting the minibatch size. Previously, [19] proposed adapting the minibatch size to control the SNR of the gradient and gave convergence analysis in the GD case. Our proposed method is very similar to [19], but we used different variance estimators (see Section IV-A3) and adapt this step for non-Euclidean norms.

III. Global Majorization Bounds

Given the general algorithmic template described above, we now show its adaptation to the broad class of energy-based models, which includes the RBM, SBN, and their replicated-softmax variants, which will be studied more closely in the next section. Assume that a model has both visible observations (units) $\boldsymbol{v} \in \mathcal{V}$ and hidden units $\boldsymbol{h} \in \mathcal{H}$ where both $\mathcal{V}$ and $\mathcal{H}$ are finite sets, such as the binary vector $\{0,1\}^M$. The joint probability distribution for $\{\boldsymbol{v}, \boldsymbol{h}\}$ is parameterized by $\boldsymbol{\theta}$, and the marginal likelihood on the observations is $p_{\boldsymbol{\theta}}(\boldsymbol{v}) = \sum_{\boldsymbol{h}\in\mathcal{H}} p_{\boldsymbol{\theta}}(\boldsymbol{v}, \boldsymbol{h})$. The maximum likelihood (ML) estimator for observations $\{\boldsymbol{v}\}_{n=1,N}$ is $\boldsymbol{\theta}^{ML} = \arg\max_{\boldsymbol{\theta}} \prod_n p_{\boldsymbol{\theta}}(\boldsymbol{v}_n)$. This probability distribution may be represented in terms of a Boltzmann or Gibbs distribution with an energy function $-E_{\boldsymbol{\theta}}(\boldsymbol{v}, \boldsymbol{h})$ that is uniquely defined up to a constant by the model, with

$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{\exp(-E_{\boldsymbol{\theta}}(\boldsymbol{v}, \boldsymbol{h}))}{Z(\boldsymbol{\theta})}.$$

$Z(\boldsymbol{\theta})$, called the partition function, forces the sum of the probability for all possible states $\{\boldsymbol{v}, \boldsymbol{h}\}$ to equal 1. The objective function can be written as the sum of the data term $f(\boldsymbol{\theta})$ and the log-partition function $\log Z(\boldsymbol{\theta})$,

$$\begin{aligned} \boldsymbol{\theta}^{ML} &= \arg\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) + \log Z(\boldsymbol{\theta}) \\ f(\boldsymbol{\theta}) &= -\tfrac{1}{N}\textstyle\sum_n \log \sum_{\boldsymbol{h}} \exp(-E_{\boldsymbol{\theta}}(\boldsymbol{v}_n, \boldsymbol{h})) \\ \log Z(\boldsymbol{\theta}) &= \log \textstyle\sum_{\boldsymbol{v}} \sum_{\boldsymbol{h}} \exp(-E_{\boldsymbol{\theta}}(\boldsymbol{v}, \boldsymbol{h})). \end{aligned} \tag{10}$$

Although the models we discuss in Section IV are non-convex, it is possible to derive a global upper bound on $F(\boldsymbol{\theta})$ by combining upper bounds on the data term $f(\boldsymbol{\theta})$ and the log-partition function $\log Z(\boldsymbol{\theta})$. We first note that for energy functions of class $\mathcal{C}^1$ with respect to parameters $\boldsymbol{\theta}$ have a bound on the data term.

Theorem III.1. *The difference between $f(\boldsymbol{\theta})$ and $f(\boldsymbol{\phi})$ for parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ is bound by*

$$\begin{aligned} f(\boldsymbol{\phi}) - f(\boldsymbol{\theta}) &\leq \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}), \boldsymbol{\phi} - \boldsymbol{\theta} \rangle \\ &- \min_{n,\boldsymbol{h}} (E_{\boldsymbol{\theta}}(\boldsymbol{v}_n, \boldsymbol{h}) - E_{\boldsymbol{\phi}}(\boldsymbol{v}_n, \boldsymbol{h}) \\ &+ \langle \nabla_{\boldsymbol{\theta}} E(\boldsymbol{v}_n, \boldsymbol{h}), \boldsymbol{\phi} - \boldsymbol{\theta} \rangle). \end{aligned}$$

See Appendix B for a proof.

We note that for convex negative energy function, such as the RBM, then Jensen's inequality simplifies this to

$$f(\boldsymbol{\phi}) \leq f(\boldsymbol{\theta}) + \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}), \boldsymbol{\phi} - \boldsymbol{\theta} \rangle. \tag{11}$$

The log partition function has a similar bound.

Theorem III.2. *The difference in the log partition function evaluated at parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ is bound by*

$$\begin{aligned} \log Z(\boldsymbol{\phi}) - \log Z(\boldsymbol{\theta}) &\leq \langle \nabla_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta}), \boldsymbol{\phi} - \boldsymbol{\theta} \rangle \\ &+ \max_{\boldsymbol{v},\boldsymbol{h}} (E_{\boldsymbol{\theta}}(\boldsymbol{v}, \boldsymbol{h}) - E_{\boldsymbol{\phi}}(\boldsymbol{v}, \boldsymbol{h}) \\ &+ \langle \nabla_{\boldsymbol{\theta}} E(\boldsymbol{v}, \boldsymbol{h}), \boldsymbol{\phi} - \boldsymbol{\theta} \rangle) \qquad (12) \\ &+ \frac{1}{2} \max_{\boldsymbol{v},\boldsymbol{h}} (E_{\boldsymbol{\theta}}(\boldsymbol{v}, \boldsymbol{h}) - E_{\boldsymbol{\phi}}(\boldsymbol{v}, \boldsymbol{h}))^2. \end{aligned}$$

See Appendix C for a proof.

To apply these theorems, we focus on two broad special cases. First, in many generative BN models, the partition function is known analytically as a constant, so only the data term changes. In this case, an upper bound can be found by utilizing only Theorem III.1. The second special case is when the energy function is linear, such as in Ising Models, Binary RBMs, or in the RBM part of the Deep Belief Net. In this case, Theorem III.2 reduces to

$$\begin{aligned} f(\boldsymbol{\phi}) &\leq f(\boldsymbol{\theta}) + \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}), \boldsymbol{\phi} - \boldsymbol{\theta} \rangle \\ &+ \frac{1}{2} \max_{\boldsymbol{v},\boldsymbol{h}} (\langle \nabla_{\boldsymbol{\theta}}(-E(\boldsymbol{v}, \boldsymbol{h})), \boldsymbol{\phi} - \boldsymbol{\theta} \rangle)^2. \end{aligned} \tag{13}$$

The first max statement in (12) drops out because the first order approximation in a linear function is exact. The second max statement is simplified to only depend on the gradient that determines the difference between the energy functions exactly.

Given Theorems III.1 and III.2, the global bound on $F(\boldsymbol{\theta})$ is the combination of the upper bound on (11) and (12) (or (11) and (13) if appropriate). Note that $\nabla F(\boldsymbol{\theta}) = \nabla f(\boldsymbol{\theta}) + \nabla(\log Z(\boldsymbol{\theta}))$, so the bound is dependent on the gradient with respect to $F(\boldsymbol{\theta})$.

If instead of using the ML estimator, a penalized ML or maximum *a posteriori* scheme is used, the global lower bounds have an additional term due to the penalization on the parameters.

A. Variational Methods

Instead of directly using the model likelihood to estimate model parameters, variational methods provide a lower bound to the model likelihood, replacing the true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{h}|\boldsymbol{v})$ over the hidden units with a simple, tractable form $q(\boldsymbol{h})$. This approximation is commonly performed in Sigmoid Belief Nets [20, 6, 21]. The Evidence Lower Bound Objective (ELBO) uses this variational posterior to give a lower bound on the model likelihood. The ELBO is then maximized instead of the model likelihood,

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\boldsymbol{v}) &\geq \mathcal{L} \\ \mathcal{L} &= \mathbb{E}_{q(\boldsymbol{h})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{h},\boldsymbol{v}) - \log q(\boldsymbol{h})] \\ \mathcal{L} &= -g(\boldsymbol{\theta},q) - \log Z(\boldsymbol{\theta}) - \mathbb{E}_{q(\boldsymbol{h})}[\log q(\boldsymbol{h})] \\ g(\boldsymbol{\theta},q) &= -\log \sum_{\boldsymbol{h}} \exp(-E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h}) + \log q(\boldsymbol{h})).\end{aligned}$$

The log partition function has the same form as $g(\boldsymbol{\theta},q)$, i.e., the log of the sum of exponentials. In the parallel to (11), the difference between $g(\boldsymbol{\theta},q)$ and $g(\boldsymbol{\phi},q)$ has the bound

$$\begin{aligned}g(\boldsymbol{\phi},q) &\leq g(\boldsymbol{\theta},q) + \langle \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta},q), \boldsymbol{\phi}-\boldsymbol{\theta}\rangle \\ &- \min_{n,\boldsymbol{h}}(-E_{\boldsymbol{\phi}}(\boldsymbol{v}_n,\boldsymbol{h}) + E_{\boldsymbol{\theta}}(\boldsymbol{v}_n,\boldsymbol{h}) \\ &+ \langle \nabla_{\boldsymbol{\theta}}(-E(\boldsymbol{v}_n,\boldsymbol{h})), \boldsymbol{\phi}-\boldsymbol{\theta}\rangle). \qquad (14)\end{aligned}$$

The proof of (14) is identical in form to the proof of Theorem III.1 in Appendix B. Using (14), the bounding techniques hold for both the likelihood and the ELBO.

IV. Model Definitions, Majorization Bounds, and Descent Schemes

Here we show the application of Theorems III.1 and III.2 to specific types of models, and we show that similar directed and undirected graphical models have similar bounds on their parameters. The norms used in this section and their associated #-operators can be found in Table I for reference.

A. Binary Models: Sigmoid Belief Nets and Binary Restricted Boltzmann Machines

Both the Sigmoid Belief Net (SBN) [22] and the Binary RBM [2] consist of a two layer model with visible units $\boldsymbol{v} \in \{0,1\}^M$ and hidden units $\boldsymbol{h} \in \{0,1\}^J$ with parameters $\boldsymbol{\theta} = \{\boldsymbol{c},\mathbf{W},\boldsymbol{b}\}$, where $\boldsymbol{c} \in \mathbb{R}^M$, $\boldsymbol{b} \in \mathbb{R}^J$, and $\mathbf{W} \in \mathbb{R}^{M\times J}$. Both models have the relationship that

$$p_{\boldsymbol{\theta}}(\boldsymbol{v}|\boldsymbol{h}) = \prod_{m=1}^{M} p_{\boldsymbol{\theta}}(v_m|\boldsymbol{h}) = \prod_{m=1}^{M} \text{Bern}(v_m;\sigma([\boldsymbol{c}+\mathbf{W}\boldsymbol{h}]_m).$$

However, because the SBN is a directed graphical model and the RBM is an undirected graphical model, the relationship between the hidden units and the visible units is different. Specifically, the SBN has a form such that the hidden nodes are simple to draw *a priori* and the RBM has posterior units that are simple to draw *a posteriori*. This is summarized by

$$\begin{aligned}\text{SBN:} \quad p_{\boldsymbol{\theta}}(\boldsymbol{h}) &= \textstyle\prod_{j=1}^{J} \text{Bern}(h_j;\sigma(b_j)) \\ \text{RBM:} \quad p_{\boldsymbol{\theta}}(\boldsymbol{h}|\boldsymbol{v}) &= \textstyle\prod_{j=1}^{J} \text{Bern}(h_j;\sigma([\boldsymbol{b}+\mathbf{W}^T\boldsymbol{v}]_j)).\end{aligned}$$

These relationship leads to the following energy functions,

$$\begin{aligned}\text{SBN:} -E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h}) &= \boldsymbol{v}^T\boldsymbol{c} + \boldsymbol{v}^T\mathbf{W}\boldsymbol{h} + \boldsymbol{h}^T\boldsymbol{b} \\ &- \textstyle\sum_{m=1}^{M} \log(1+\exp([\boldsymbol{c}+\mathbf{W}\boldsymbol{h}]_m)) \\ &- \textstyle\sum_{j=1}^{J} \log(1+\exp(b_j)) \\ \text{RBM:} -E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h}) &= \boldsymbol{v}^T\boldsymbol{c} + \boldsymbol{v}^T\mathbf{W}\boldsymbol{h} + \boldsymbol{h}^T\boldsymbol{b}.\end{aligned}$$

Although the SBN has a more complicated energy function, the partition function is a constant at 1. In contrast, the RBM log partition function is intractable to calculate for realistic problem sizes, and is estimated through Annealed Importance Sampling (AIS) [23].

We first focus specifically on the global bounds for $\mathbf{W}$ when perturbing it by an amount $\mathbf{U}$. We previously proposed a bound for the RBM using the Schatten-∞ in [13], which was derived by viewing the RBM objective as a difference-of-convex-functions problem. This bound was

$$F(\mathbf{W}+\mathbf{U}) \leq F(\mathbf{W}) + \text{tr}(\nabla_{\mathbf{W}}F(\mathbf{W})\mathbf{U}^T) + \tfrac{MJ}{2}||\mathbf{U}||^2_{S^\infty} \quad (15)$$

Here we apply our framework to the SBN problem. Since the SBN has a constant, analytic partition function, we only need to apply Theorem III.1. In the SBN this reduces to bounding the first order approximation over $\mathbf{W}$ on $\boldsymbol{v}^T\mathbf{W}\boldsymbol{h} - \sum_{m=1}^{M}\log(1+\exp([\boldsymbol{c}+\mathbf{W}\boldsymbol{h}]_m))$. Analyzing these functions gives a bound on the Schatten-∞ norm, and we provide a derivation of this in Appendix E. In the SBN, $\mathbf{W}$ has a global bound:

$$F(\mathbf{W}+\mathbf{U}) \leq F(\mathbf{W}) + \text{tr}(\nabla_{\mathbf{W}}F(\mathbf{W})\mathbf{U}^T) + \frac{J}{8}||\mathbf{U}||^2_{S^\infty}. \quad (16)$$

Intriguingly, under our results both these models have the same Lipschitz constants in the Schatten-∞ norm as in the Frobenius norm. Typically the Frobenius norm would actually have a better constant, but under our theory we cannot prove this. The minimizer of this majorization function is not in the direction of the gradient. Rather, the #-operator that minimizes this bound is given by taking the SVD of the gradient, $\mathbf{A}\text{diag}(\boldsymbol{\lambda})\mathbf{B}^T = \nabla_{\mathbf{W}}F(\mathbf{W})$, and setting $\mathbf{U} = s||\boldsymbol{\lambda}||_1\mathbf{A}\mathbf{B}^T$, with s set at $\frac{-4}{J}$ for the SBN and $\frac{-1}{MJ}$ for the RBM.

We bound the vector parameters $\boldsymbol{c}$ and $\boldsymbol{b}$ on the ℓ_2 norm. This leads to standard gradient updates, and these majorization functions are both minimized by a stepsize of -4. Details can be found in Appendix E.

1) SBN Gradient Estimates: Gradient estimation in the SBN is computationally intensive. The gradient on $\mathbf{W}$ is $-\nabla_{\mathbf{W}}F(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{p(\boldsymbol{h}|\boldsymbol{v}_n)}[(\boldsymbol{v}_n - \sigma(\boldsymbol{c}+\mathbf{W}\boldsymbol{h}))\boldsymbol{h}^T]$. Following [22], each of these expectations can be estimated by Monte Carlo integration. Generating samples from $p(\boldsymbol{h}|\boldsymbol{v})$ is not analytic, and instead a Gibbs sampler is used to sample from $p(h_j|h_i, i \neq j, \boldsymbol{v})$. This estimation procedure costs $\mathcal{O}(N_{batch}CMJ^2)$, where N_{batch} is the number of data samples used in a mini-batch and C is the number of Gibbs sweeps used in the estimation procedure. The cost with the standard variational approximation is the same with C representing the number of block coordinate passes. This gradient estimation is the main computational bottleneck in the algorithm.

This procedure is similar to the Contrastive Divergence (CD) procedure for the RBM [2]. In CD, a Gibbs sampler is repeatedly used to draw approximate samples from the full model of $p(\boldsymbol{v}, \boldsymbol{h})$, and then these approximate samples are used to estimate the model gradients. For CD, the computational cost is $\mathcal{O}(N_{batch}CMJ)$, which scales better than the estimation procedure for the SBN. We note there is an increasing amount of work on sampling related binary variables [24], as well as deterministic methods approximating functions of random fields [25]. If possible, adapting these methods to the SBN case could make the computational cost the same for the gradient estimation in the SBN and the RBM.

Instead of using posterior sampling, variational methods [20, 5] are often used, putting a simple, tractable form on $q(\boldsymbol{h}) = \prod_j \text{Bern}(h_j; \pi_j)$ and minimizing $\boldsymbol{\pi} = \arg\min_{\pi} \text{KL}(q(\boldsymbol{h})||p(\boldsymbol{h}|\boldsymbol{v}))$ in order to get simple estimation procedures. In this case, a simpler MCI scheme is used to estimate the gradient by drawing samples from $q(\boldsymbol{h})$, as in [26]. This can be used with mini-batches to estimate parameters in the Stochastic Variational Inference scheme [27].

2) Computational Cost of the #-Operator: The #-operator requires an SVD, which is typically regarded an expensive operation. However, the computational cost of gradient estimation for the RBM ($\mathcal{O}(N_{batch}CMJ)$) and the SBN ($\mathcal{O}(N_{batch}CMJ^2)$) is *more* expensive, rendering the SVD as a small overhead on an iteration. The computational cost of the SVD is $\mathcal{O}(MJ\min(M, J, N_{batch}))$. Since relatively large numbers of samples ($C \simeq 25$ [23]) are needed for the RBM, a typical batch size makes the SVD relatively cheap. For the SBN, the relative cost scales at $\mathcal{O}(CN_{batch}J/\min(M, J, N_{batch})$. For large networks and batch sizes, the gradient estimation cost scales to make the SVD a relatively cheaper operation.

3) Adapting Minibatch Size to Gradient SNR: An approach for convergence given in Section II-C is to use a constant step size while maintaining a minimum SNR given on the gradient estimate. However, it is not known *a priori* what the magnitude of the gradient will be, nor what the estimation error will be. One strategy to approximate this requirement is to estimate the current SNR by using bootstrap methods over the data points used in the gradient estimation, using sufficient statistics saved during the Monte Carlo Integration or the Variational Posterior step. This bootstrap step introduces trivial overhead, since the computations required are the same as in Section IV-A2.

While the theory uses the dual norm (nuclear norm), we propose here to estimate the Frobenius norm error for two reasons: (i) the nuclear norm calculation requires non-trivial computational resources, and repeating it for a bootstrap leads to non-negligible overhead, and (ii) the Frobenius norm error is expected to scale with $\mathcal{O}(\frac{1}{\sqrt{N_b}})$ [19]. This scaling relationship makes it easy to estimate the number of samples necessary for the desired SNR (i.e. to decrease SNR by half, the batch size must be increased four times).

We note that there is a second source of stochastic noise, from Monte Carlo integration (MCI). Surprisingly, we found empirically that changing the number of MCI samples in the SBN had a much smaller effect on the noise estimate than the minibatch size, and that adapting the minibatch size had a much larger effect for given computational resources. We note as well that the error due to MCI also scales $\mathcal{O}(\frac{1}{\sqrt{CN_b}})$, so increasing the minibatch size also decreases the MCI error. In our experiments, we only adapt the minibatch size. After the batch size is reached, the number of samples must be increased, but this limit is not feasibly obtained in these problems.

4) SBN Function Evaluation: Explicit function evaluations for the SBN is limited to models with very small treewidth [20]. Instead of calculating the model likelihood directly, one approach is to lower bound the model likelihood with variational methods [20, 21]. Recently, [21] proposed to use the Harmonic Mean Estimator to estimate the model likelihood in the SBN. However, this estimator is known to dramatically overestimate performance, and the variance on the estimator may be infinite [28].

The Annealed Importance Sampler (AIS) [29] is an alternative to the Harmonic Mean Estimator based on the Gibbs samplers and simulated annealing. Asymptotically, the estimator is unbiased and the variance goes to 0. This has recently been applied to evaluate RBM and the Deep Belief Network [23], as well as topic models [3]. While the AIS estimator is asymptotically unbiased, for finite numbers of samples the estimates are biased positively, which can cause performance to be overestimated. Recently, the Reverse AIS Estimator (RAISE) [30] was proposed to address this problem in Markov Random Fields, and is asymptotically unbiased and biased negatively for finite sample sizes. Running both AIS and RAISE for finite sample sizes can give an accurate performance range on the model.

For the SBN, we have implemented AIS to perform accurate model evaluations, and give algorithmic details in Appendix D. As well, we extended the RAISE method to the SBN to give a lower bound on the model performance. These methods are costly, so line search methods are inappropriate in this problem.

5) Deep Versions: There has been significant interest in deeper versions of the SBN and the RBM. For the RBM, Salakhutdinov [4] gave a framework for learning and evaluating a multiple hidden layer RBM. The SBN has historically had deeper versions [22, 20], and is the inspiration for the Deep Belief Net [31]. In a three-layer model, they have visible nodes $\boldsymbol{v} \in \{0,1\}^M$, $\boldsymbol{h}^{(1)} \in \{0,1\}^{J^{(1)}}$, and $\boldsymbol{h} \in \{0,1\}^{J^{(2)}}$, with parameters $\{\boldsymbol{c}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}, \mathbf{W}^{(1)}, \mathbf{W}^{(2)}\}$. The energy function for the DRBM is

$$\begin{aligned} -E_{\boldsymbol{\theta}}(\boldsymbol{v}, \boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}) &= \boldsymbol{v}^T\boldsymbol{c} + \boldsymbol{v}^T\mathbf{W}^{(1)}\boldsymbol{h}^{(1)} + (\boldsymbol{h}^{(1)})^T\boldsymbol{b}^{(1)} \\ &+ \boldsymbol{h}^{(1)}\mathbf{W}^{(2)}\boldsymbol{h}^{(2)} + (\boldsymbol{h}^{(2)})^T\boldsymbol{b}^{(2)}, \end{aligned}$$

and the generative model for the DSBN is

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{h}^{(2)}) &= \textstyle\prod_{j=1}^{J^{(2)}} \text{Bern}(h_j^{(2)}; \sigma(b_j^{(2)})) \\ p_{\boldsymbol{\theta}}(\boldsymbol{h}^{(1)}|\boldsymbol{h}^{(2)}) &= \textstyle\prod_{j=1}^{J^{(1)}} \text{Bern}(h_j^{(1)}; \sigma([\boldsymbol{b}^{(1)} + \mathbf{W}^{(2)}\boldsymbol{h}^{(2)}]_j) \\ p_{\boldsymbol{\theta}}(\boldsymbol{v}|\boldsymbol{h}^{(2)}) &= \textstyle\prod_{m=1}^{M} \text{Bern}(v_m; \sigma([\boldsymbol{c} + \mathbf{W}^{(1)}\boldsymbol{h}^{(1)}]_m). \end{aligned}$$

TABLE I
LIST OF #-OPERATORS WHEN USING THE FOLLOWING NORMS IN (3)

Norm	$\mathbf{X}^{\#}$ or $\boldsymbol{x}^{\#}$
$\|\|\cdot\|\|_2^2$ or $\|\|\cdot\|\|_F^2$	$\boldsymbol{x}$ or $\mathbf{X}$
$\|\|\cdot\|\|_{S^\infty}^2$	$\|\|\boldsymbol{s}\|\|_1\mathbf{U}\mathbf{V}$, $\mathbf{U}\text{diag}(\boldsymbol{s})\mathbf{V}^T = \mathbf{X}$ (SVD)
$\|\|\cdot\|\|_\infty^2$	$\|\|\boldsymbol{x}\|\|_1 \times \text{sign}(\boldsymbol{x})$
$\max_{m=1,\ldots,M} \|\|\mathbf{U}_{m,\cdot}\|\|_1^2$	$\mathbf{A}\sum_{m'} \|\|\mathbf{X}_{m',\cdot}\|\|_\infty$ $A_{mj} = \text{sign}(X_{mj})1_{\|X_{mj}\|=\|\|X_{m\cdot}\|\|_\infty}$

TABLE II
LIPSCHITZ-GRADIENT CONSTANTS AND THEIR IMPLIED STEP SIZES, WITH RESPECT TO THE PARAMETER MATRIX $\mathbf{W} \in \mathbb{R}^{M \times J}$. CONSTANTS M, J, AND $\bar{D}$ ARE DEFINED IN SECTION IV.

	RBM		**SBN**		**RSBN**		**RS-RBM**	
	Lip.	step	Lip.	step	Lip.	step	Lip.	step
S_∞	MJ	$\frac{1}{MJ}$	$\frac{J}{4}$	$\frac{4}{J}$	$\frac{\bar{D}J}{1}$	$\frac{1}{\bar{D}J}$	$\frac{\bar{D}^2J}{1}$	$\frac{1}{\bar{D}^2J}$
$\ell_1 R$	–	–	–	–	$\frac{\bar{D}}{1}$	$\frac{1}{\bar{D}}$	$\frac{\bar{D}^2}{1}$	$\frac{1}{\bar{D}^2}$

Both the DSBN and the DRBM have Schatten-∞ bounds on the parameters $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$. The minimizers of the majorization functions for these parameters are the same as (15) and (16) for $\mathbf{W}^{(1)}$, and for $\mathbf{W}^{(2)}$ the stepsize is $-\frac{4}{J^{(2)}}$ for the DSBN and $-\frac{1}{J^{(1)}J^{(2)}}$ for the DRBM. These patterns continue for deeper models.

B. Topic Modeling: Replicated Softmax Models

For topic modeling problems, document n is represented by a vector of counts $\boldsymbol{v}_n \in \mathbb{Z}^M$, where each unit v_{nm} represents the number of times the unique dictionary word m appears in document n. The total number of words in the document is $D_n = ||\boldsymbol{v}_n||_1$. This bag-of-words assumption is common in probabilistic topic modeling problems [32].

An undirected topic model called the Replicated Softmax-RBM (RS-RBM) was proposed in [3], and a directed topic model, which we will refer to as the Replicated Softmax Belief Network (RSBN) was proposed in [26]. Both use this vector representation of the observations and binary hidden nodes $\boldsymbol{b} \in \{0,1\}^J$. Both models are parameterized by $\boldsymbol{\theta} = \{\boldsymbol{c}, \mathbf{W}, \boldsymbol{b}\}$, and both have a multinomial distribution given the hidden units, with

$$\boldsymbol{v}_n|\boldsymbol{h}_n = \text{Multi}(D_n; \text{softmax}(\boldsymbol{c} + \mathbf{W}\boldsymbol{h})).$$

Like in the binary models, the difference in the undirected and directed models is the relationship to the hidden units,

$$\begin{array}{lll} \text{RSBN:} & p_{\boldsymbol{\theta}}(\boldsymbol{h}) = & \prod_{j=1}^J \text{Bern}(h_j; \sigma(b_j)) \\ \text{RS-RBM:} & p_{\boldsymbol{\theta}}(\boldsymbol{h}|\boldsymbol{v}) = & \prod_{j=1}^J \text{Bern}(h_j; \sigma([D\boldsymbol{b} + \mathbf{W}^T\boldsymbol{v}]_j). \end{array}$$

These relationships give the following energy functions,

$$\begin{aligned} \text{RSBN:} -E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h}) &= \boldsymbol{v}^T\boldsymbol{c} + \boldsymbol{v}^T\mathbf{W}\boldsymbol{h} + \boldsymbol{h}^T\boldsymbol{b} \\ &- D\log\textstyle\sum_{m=1}^M \exp([\boldsymbol{c} + \mathbf{W}\boldsymbol{h}]_m)) \\ &- \textstyle\sum_{j=1}^J \log(1 + \exp(b_j)) \\ \text{RS-RBM:} -E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h}) &= \boldsymbol{v}^T\boldsymbol{c} + \boldsymbol{v}^T\mathbf{W}\boldsymbol{h} + D\boldsymbol{h}^T\boldsymbol{b}. \end{aligned}$$

We first examine the relationship for $\mathbf{W}$ versus a perturbed version $\mathbf{W} + \mathbf{U}$. For the RS-RBM, the energy function is linear and convex, so we use (11) and (13). The bound depends on the final term in (13) over the gradient $-\nabla_{\mathbf{W}}E(\boldsymbol{v},\boldsymbol{h}) = \boldsymbol{v}^T\boldsymbol{h}^T$, which is simplified to

$$\frac{1}{2}\max_{\substack{\boldsymbol{v}\in\mathbb{Z}^M ||\boldsymbol{v}||_1 \leq D \\ \boldsymbol{h}\in\{0,1\}^J}} \text{tr}(\boldsymbol{v}\boldsymbol{h}^T\mathbf{U}^T)^2 \leq \frac{D^2}{2}||\mathbf{U}||_{S^\infty}^2.$$

Letting $\bar{D^2}$ represent the average squared number of words in a document, the global bound is

$$F(\mathbf{W}+\mathbf{U}) \leq F(\mathbf{W}) + \text{tr}(\nabla_{\mathbf{W}}F(\mathbf{W})\mathbf{U}^T) + \frac{\bar{D^2}J}{2}||\mathbf{U}||_{S^\infty}^2 \quad (17)$$

This leads to the same update steps as in the SBN and RBM with a different stepsize. A further derivation of (17) is in Appendix F. We note that this could be alternatively bounded by

$$\begin{aligned} F(\mathbf{W}+\mathbf{U}) &\leq F(\mathbf{W}) + \text{tr}(\nabla_{\mathbf{W}}F(\mathbf{W})\mathbf{U}^T) \\ &+ \frac{\bar{D^2}}{2}\max_{m=1,\ldots,M}||\mathbf{U}_{m,\cdot}||_1^2. \end{aligned} \quad (18)$$

The relationship is bounded by the maximum ℓ_1-norm of the perturbation in a row. Letting $\mathbf{A} = \nabla_{\mathbf{W}}F(\mathbf{W})$, the minimizer of (18) is given by

$$\tilde{\mathbf{U}}_{mj} = \begin{cases} \alpha\ \text{sign}(A_{mj})\sum_{m'}||A_{m',\cdot}||_\infty, & |A_{mj}| = ||\mathbf{A}_{m\cdot}||_\infty \\ 0, & \text{otherwise} \end{cases}$$

with $\alpha = \frac{1}{\bar{D^2}}$. The result for the RSBN depends on the first order approximation of the concave negative energy function. Despite a different derivation, the RSBN has the same bounds as the RS-RBM expressed in (18) and (17), with $\bar{D^2}$ replaced by $\bar{D}$. The derivation of (17) for the RSBN is in Appendix G.

We note that similar to the binary models in Section IV-A, our bound for these problems is tighter on the Schatten-∞ norm compared to the Frobenius norm. This motivates using methods that work either in the Schatten-∞ norm or the max ℓ_1 row norm. There is a computational trade-off between the two geometries because the max ℓ_1-row update is parallelizable and the nonlinear operations on the gradient are calculable in linear time.

Both the RS-RBM and RSBN are bound on the ℓ_∞ norm for $\boldsymbol{c}$ and the ℓ_2 norm on $\boldsymbol{b}$. The optimal step for the ℓ_∞ norm uses only the ℓ_1 norm and the sign of the gradient, and bound derivation are provided in Appendix G and F.

1) Gradient Estimation, Variational Methods, and Function Evaluations: For the RS-RBM, the gradient estimation and function evaluation proceed as in [3], which estimates gradients with a contrastive divergence procedure and the gradient with an AIS sampler.

For the RSBN, the gradient is given in [26], and we estimate this via Monte Carlo Integration with a Gibbs sampler over the hidden units to get approximate samples. To match the performance evaluation in the RS-RBM, we develop an AIS sampler for the RSBN, which is very close in procedure to the RS-RBM AIS sampler in [3], as well as the AIS sampler for the SBN detailed in Appendix D. For the RS-RBM, we use the approach of [3]. The variational methods of [20, 26] are applied here as well, which evaluate a lower bound on the model likelihood.

The same procedure discussed in Section IV-A3 can be used to approximate the SNR and choose an appropriate minibatch size.

2) Computational Complexity of the #-Operator: We note here that the computational costs of the gradient estimations follow Section IV-A2, where the RS-RBM has the same computational scaling as the RBM and the RSBN has the same computational cost as the SBN. Because of this, the SVD required in the #-operator when the spectral norm is used leads to small overhead. The #-operator corresponding to the bound in (18) is $\mathcal{O}(MJ)$ and causes trivial overhead for the size of the minibatch and the number of Gibbs samples used in the experiments.

3) Deeper Versions: A deeper version of the RS-RBM, called the Over-Replicated Softmax-RBM, was introduced in [33]. A stochastic autoencoder deeper version of the RSBN was shown in [6, 26]. Instead of these models, a deep RSBN and a deep RS-RBM could also be defined by using the same relationship for hidden layers $\boldsymbol{h}^{(1)}$ and $\boldsymbol{h}^{(2)}$ shown in Section IV-A5. In these models, $\mathbf{W}^{(1)}$ has the bound given in (18) or (17), while the deeper layers would have the same global bounds and steps as the DSBN and the DRBM for $\mathbf{W}^{(\ell)}$ for $\ell > 1$.

C. Deep Belief Nets

Deep Belief Nets (DBN) [31] are deep graphical models that have both undirected and directed edges. For a 3-layer DBN, the first hidden layer $\boldsymbol{h}^{(1)} \in \{0,1\}^{J^{(1)}}$ and the second hidden layer $\boldsymbol{h}^{(2)} \in \{0,1\}^{J^{(2)}}$ are jointly drawn from an undirected RBM model with parameters $\{\boldsymbol{b}^{(1)}, \mathbf{W}^{(2)}, \boldsymbol{b}^{(2)}\}$, and the visible units $\boldsymbol{v} \in \{0,1\}^M$ are drawn from the directed model with $v_m|\boldsymbol{h}^{(1)} \sim \text{Bern}(v_m; \sigma([\boldsymbol{c} + \mathbf{W}^{(1)}\boldsymbol{h}^{(1)}]_m))$. The energy function for this model is written as

$$\begin{aligned} -E_{\boldsymbol{\theta}}(\boldsymbol{h}, \boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}) &= \boldsymbol{v}^T\boldsymbol{c} + \boldsymbol{v}^T\mathbf{W}^{(1)}\boldsymbol{h}^{(1)} + (\boldsymbol{h}^{(1)})^T\boldsymbol{b}^{(1)} \\ &- \sum_{m=1}^{M} \log(1 + \exp([\boldsymbol{c} + \mathbf{W}^{(1)}\boldsymbol{h}^{(1)}]_m)) \\ &+ (\boldsymbol{h}^{(1)})^T\mathbf{W}^{(2)}\boldsymbol{h}^{(2)} + (\boldsymbol{h}^{(2)})^T\boldsymbol{b}^{(2)}. \end{aligned}$$

The exact log partition function is intractable for realistic size problems, but there are joint AIS and variational lower bounding techniques to approximate the model likelihood [23].

The theories that we have developed allow us to create global bounds on the DBN, and the DBN follows the bounds developed for the SBN and RBM cases. Specifically, the bound on $\mathbf{W}^{(1)}$ is dependent on the Schatten-∞ norm with $\frac{J^{(1)}}{8}||\mathbf{U}^{(1)}||^2_{S^\infty}$, and the bound on $\mathbf{W}^{(2)}$ is dependent on the Schatten-∞ norm with $\frac{J^{(1)}J^{(2)}}{2}||\mathbf{U}^{(2)}||^2_{S^\infty}$. An L-layer DBN models will follow the SBN updates on $\mathbf{W}^{\ell}$ for $\ell = 1, \ldots, L-1$ and the RBM for $\mathbf{W}^{(L)}$.

V. Experiments

We compare the stochastic algorithms using the proposed geometries to Stochastic Gradient Descent (SGD), as well as RMSprop [9] and ADAgrad [8]. Both ADAgrad and RMSprop are pertinent comparisons, because they both are designed to utilize the local geometry of the learning problem [34]. These general-purpose descent algorithms dynamically learn the geometry from the historical gradients, and perform an element-wise re-weighting of the gradient. These procedures are widely used in training neural networks, and are subject to continued investigation [35, 34]. These algorithms use the parameters in Table III when appropriate (ADAgrad and RMSprop have defined step-size schemes in the algorithm).

In contrast, our algorithms use the geometry of the objective function, and do not need the historical gradient information. It therefore does not have a burn-in period where it learns the appropriate settings. We denote the algorithm utilizing the maximum ℓ_1 row as L1R. Following [13], we call the stochastic algorithm using the Schatten-∞ norm Stochastic Spectral Descent. As well, the comparative performance of Stochastic Spectral Descent actually *improves* with increased dimensionality due to the optimization radius (see Section II-B). This result is similar to [11], which bases algorithms on the ℓ_∞ norm to scale to larger problem sizes.

We perform two distinct versions of the Stochastic Spectral Descent algorithm: (i) uses a constant minibatch size with a heuristically decreasing step-size constant (denoted as SSD) with the values in Table III; and (ii) uses the same step size for all iterations, but adapts the size of the minibatch to approximate a constant SNR in the experiments (here, chosen to be 1/3), which we denote as SSD-adapt. This procedure estimates the current SNR on $\mathbf{W}$ via the method in Section IV-A3. If the SNR on the current iteration is too low, then we draw the additional datapoints necessary to achieve the desired SNR. The next minibatch size is chosen to approximately give the desired SNR. A minimum minibatch size of 10 is used. The SSD-adapt scheme approximates the conditions necessary for convergence, although it does not strictly satisfy them due to the use of the Frobenius instead of the dual norm for computational reasons.

A. Implementation Details

The codes developed here are implemented in MATLAB R2015a. However, because the bulk of the computational cost is due to large, vectorized linear algebra operations, the relative time taken by the SVD should be fairly accurate. For the RBM based models, the computational time is similar to the RBM toolbox[2], and a version written in C++ with Eigen[3] gave marginal speed-ups. For the directed graphical models, because the computational scaling of the gradient

[2] https://github.com/skaae/rbm_toolbox

[3] http://eigen.tuxfamily.org/

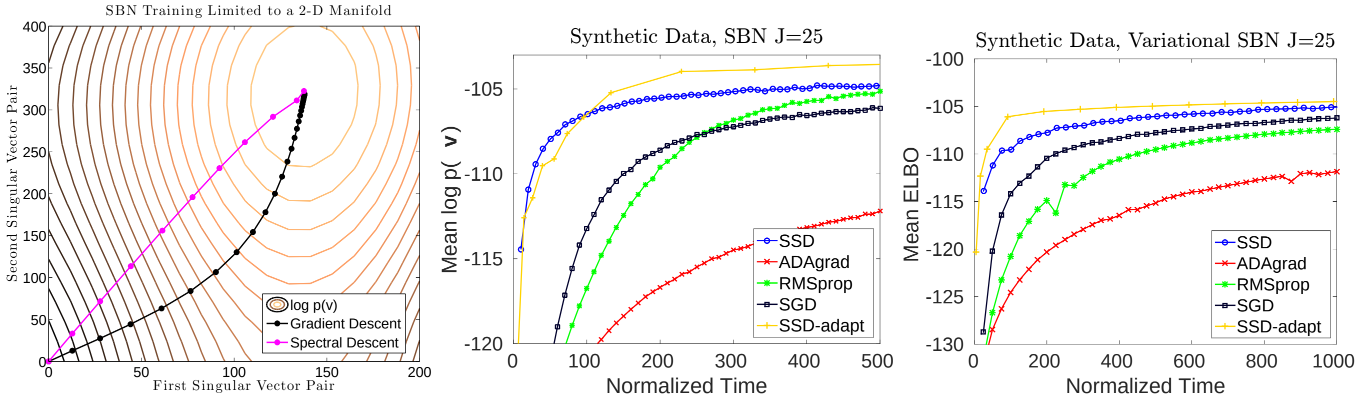

Fig. 1. Synthetic Dataset for the SBN model (Left) The objective surface as a function of the first singular vector pair and the second singular vector pair of the gradient at the initial point. The training curves for gradient descent and spectral descent projected into this space are shown to demonstrate the improved search direction of spectral descent. (Middle) Learning curves for synthetic data with MAP estimation. (Right) Learning curves with synthetic data for Variational approximations

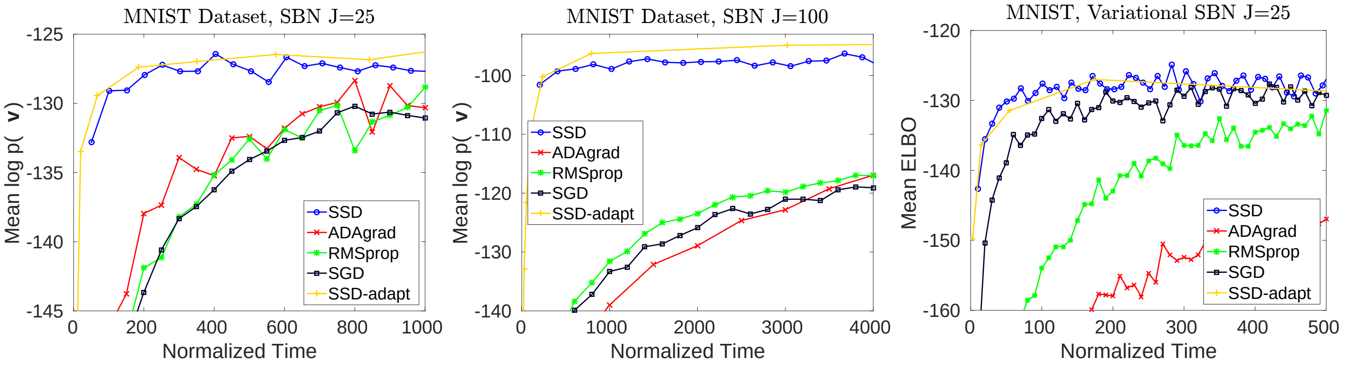

Fig. 2. Learning curves for the MNIST dataset on the SBN model. Using derived step-sizes for SSD, and tuned step-sizes for other algorithms (Left) MAP estimation curves for J=25 (Middle) MAP estimation problem for J=100 (Right) Variational estimation problem for J=25

TABLE III
EXPERIMENTAL SETTINGS

	SBN	RSBN	RS-RBM
(R)AIS Samples	500	100	100
Gibbs Sweeps	7	7	5
W Base Step-Size	$8/J$	$1/(DJ)$	$1/(J)$
Stepsize Decay	$t^{-.5}$	$t^{-.5}$	$t^{-.5}$
ℓ_2 Norm Penalty	10^{-4}	10^{-4}	10^{-4}
Minibatch Size	100	100	100

is significantly worse than the SVD operation, the overhead is trivial; in these experiments it was less than 2 percent per iteration. To normalize the presentation of the results, all timing information is shown in normalized time, where 1 unit corresponds to the about of time a SGD iteration takes. The experiments were run on a 4-core Xeon processor at 1.8 GHz with 64 GB of RAM, and repeated on several similar machines.

B. Sigmoid Belief Nets

The SBN has recently experienced an increased focus on training [26, 21]. Most recent approaches have focused on using variational approximations to learn larger models, and the SGD approach [22] is typically used to learn the objective function of small models. Here, we demonstrate that we are able to effectively learn the *true* objective function up to hundreds of hidden nodes. As well, we use the result in Section III-A and apply our techniques to the variational approximation, and also demonstrate improved fitting performance in the variational models.

To demonstrate the effect of the spectral descent step, we first use a small, synthetic dataset. First, in Figure 1, we use a small $(M = 25,\ J = 10)$ network so that explicit calculations can be used. At the initial point, we take the gradient and limit steps to the first singular vector pair and the second singular vector pair for visualization purposes. The steps from gradient descent and spectral descent are projected into this space, and we run updates for the two algorithms using the same step sizes. Gradient descent first optimizes primarily the first singular vector pair, and then turns to optimize the second singular vector pair. The spectral descent steps accurately capture an effective search direction. We continue with synthetic data, where we show the SBN for a $M = 250, J = 25$ network, with a MAP learning curves in Figure 1(middle) and VB learning curves in Figure 1 (right). These results use 8/J step sizes for the SSD and SSD-adapt

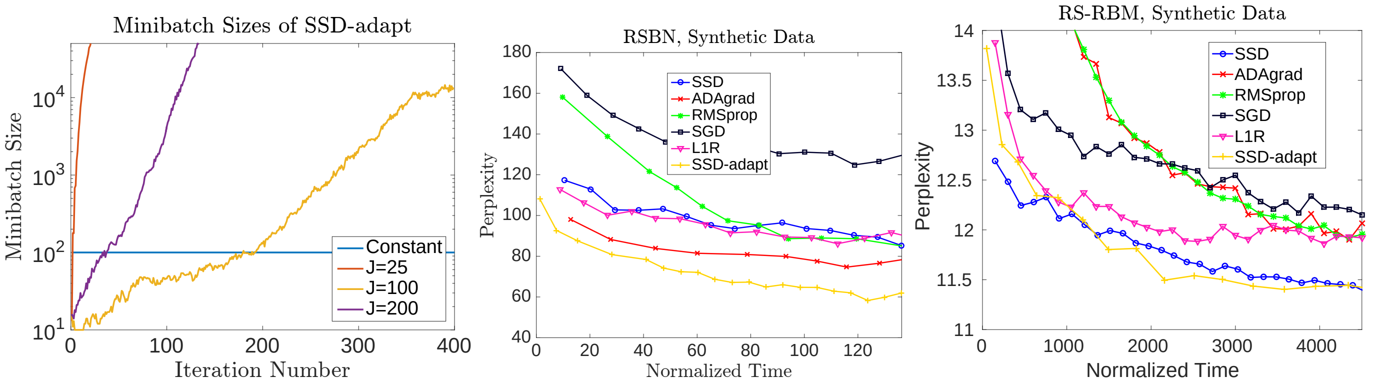

Fig. 3. (Left) The size of mini-batch used versus iteration for SSD-adapt for SBN training (Middle) Results on RSBN training reporting the perplexity metric with $J = 10$ with exact gradients and log-likelihood. (Right) Results on RS-RBM for a synthetic dataset of size $J = 25$

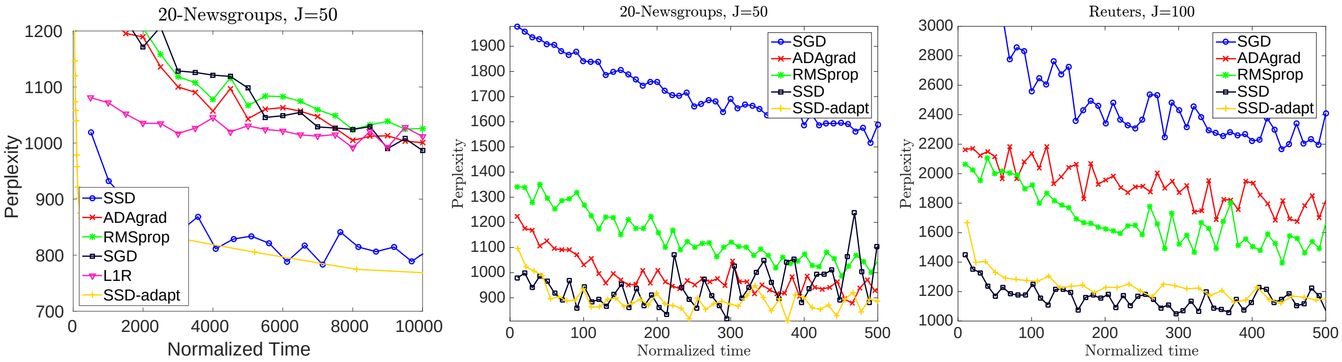

Fig. 4. (Left) Learning curves for 20 Newsgroups with the RS-RBM. (Middle) Learning curves for 20 newsgroups with the RSBN (Right) Learning curves for Reuters with the RSBN

TABLE IV
COMPARISON OF RESULTS ON MNIST. RESULTS FROM PROPOSED METHOD MARKED WITH *. BRACKETS GIVE AN UPPER AND LOWER BOUND ON PERFORMANCE. SINGLE NUMBERS ARE LOWER BOUNDS OR UNBIASED ESTIMATES.

Method	Dimension	Test log-prob.
RBM (CD3) [23]	25	-143.20
SBN (Online VB) [21]	25	-138.34
SBN (VB, SSD) *	25	-125.70
SBN (SSD) *	25	[-128.12, -127.45]
SBN (SSD-adapt) *	25	[-129.25, -127.23]
RBM (CD25) [13]	100	-97.11
SBN (VB, SSD) *	100	109.76
SBN (SSD) *	100	[-104.5, -102.0]
SBN (SSD-adapt) *	100	[-101.6, -99.1]
SBN (online VB) [21]	200	-118.12
SBN (VB) [21]	200	-116.96
SBN (NVIL) [26]	200	-113.1
SBN (VB, SSD) *	200	-109.11
SBN (SSD)*	200	[-101.0, -98.5]
SBN (SSD-adapt) *	200	[-100.4, -98.0]
RBM (SGD) [23]	500	-86.22
RBM (SSD) [13]	500	-85.65
DBN [23]	500 – 2000	-86.22
DBM [4]	500 – 1000	-84.62

algorithms, but the step sizes for all other algorithms were *tuned* by sweeping over step-sizes. This tuning proceeded by starting at the same stepsize as SSD, and increasing the stepsize over $\mathbf{W}$ until performance no longer improved. Only the best curves for the tuned algorithms (SGD, ADAgrad, RMSprop) are shown. Even for this small network, SSD shows improved performance over the competing algorithms, and SSD-adapt shows modest improvements over SSD.

We next used the MNIST dataset that has been stochastically binarized as in [23]. First, we show the learning curves for a $J = 25$ network in Figure 2 (left) for the MAP estimate and the variational estimate (middle). Even in these small models, SSD and SSD-adapt improve performance over the *tuned* competing algorithms. As well as improving learning speed, SSD-adapt is giving state-of-the-art performance over the best reported model, which is detailed in Table IV. Compared to the heuristic SSD method, SSD-adapt shows improvement in the ML estimation case, and near identical performance in the variational case.

Of primary concern is the training of *larger* networks. For the J=100 case, we show the learning curves for the MAP estimation problem in Figure 2(right). Here, after thousands of iterations, no competing algorithm is able to learn the SBN as well as SSD can in *100* iterations, representing more than an order of magnitude improvement. SSD-adapt improves over SSD as well. In Table IV, we show that this model is about

2 nats away from the RBM model of a corresponding size. While this algorithm is not beating the corresponding RBM, this is a much closer margin than previously demonstrated.

We note that the SSD-adapt algorithm allows optimization to an improved point compared to SSD, or any other compared algorithm. One reason for this is the improved optimization at later times, where SSD-adapt takes many fewer steps with a much larger batchsize. We demonstrate the batchsizes in Figure 3 (left). At early times, the batchsize is smaller than the heuristic size of 100 that we use, but at late times can be orders of magnitude larger, which drastically reduces the number of total iterations. Empirically, the growth in the batchsize appears to grow exponentially at a near constant rate.

With the SSD or the SSD-adapt algorithm, it is possible to learn even larger networks. Learning curves for these algorithms are not shown because the competing algorithms are thoroughly uncompetitive. However, we give the comparison for larger networks in Table IV, which shows over 10 nat improvement in the model likelihood with 200 hidden nodes.

C. Topic Modeling

To demonstrate performance on the RSBN and the RS-RBM models, we compared the algorithms using geometry (L1R, SSD, SSD-adapt) with SGD and RMSprop. We first test our algorithms on a dataset of synthetically generated data. These data were created by substantiating a RSBN or RS-RBM with weights sampled from a normal distribution. Unit biases were set to zero. Samples were generated by setting the hidden layer to a random binary vector and then sampling an assignment for the visible layer, either by a simple forward pass in the case of the RSBN, or with Gibbs sampling for the RS-RBM. Using initial small random weights, we then proceeded to test the convergence, in terms of perplexity, of the various algorithms on problems of different scale. Perplexity is a commonly used metric in topic-modeling [32], with $perplexity(\boldsymbol{\theta}) = \exp(-(N\bar{D})^{-1}\sum_n \log p(\boldsymbol{v}|\boldsymbol{\theta}))$. There is a deterministic mapping between the objective function (log-likelihood) and the perplexity estimate, and a decrease in perplexity corresponds to an increase in log-likelihood. The RSBN and RS-RBM used the stepsizes in Table III for SSD, SSD-adapt, and L1R, which is more optimistic than the Lipschitz step. The stepsizes for RMSprop, ADAgrad, and SGD were tuned as in the previous section.

For these synthetic datasets, we first show the performance for the RSBN with exact gradients in Figure 3(middle) with J=10 and M=100. In these case, L1R, SSD, and ADAgrad show similar performance, with greatly improved performance over SGD and RMSprop. SSD-adapt shows further improvements over the competing algorithms. For the RS-RBM, we set the CD order to 3 and used 100 samples in AIS per document as in [3]. We show a synthetic dataset with 25 hidden nodes with RS-RBM in Figure 3(right). In the RS-RBM, SSD and SSD-adapt are very similar, showing improved performance over competing algorithms.

We compare results on the well-known 20-Newsgroups dataset[4]. We used the preprocessing of [3], which reduced the vocabulary to M=2,000 most frequent words in the corpus, split the corpus into a training set (11,284 documents) and evaluation set (7,502 documents). For the RSBN, the resulting convergence plot is shown in Figure 4 (middle) for J=100. SSD is the clear winner, and both L1R and RMSprop significantly improve over SGD. The RS-RBM training result is shown in Figure 4 (left) for J=50. For the RS-RBM, SSD and SSD-adapt show similar performance, with dominant performance over competing algorithms. While SSD and SSD-adapt give essentially the same performance, we mention that SSD-adapt is significantly smoother as a result of taking many fewer iterations. As well, using the same AIS settings as [3], the estimated mean test perplexity is 841.53, which is considerably lower than the reported perplexity for SGD training of 953 [3]. Unlike the RBM problem [36], different local modes in the RS-RBM appear to have different performance levels. Our SGD code achieved a level of 945 after 50,000 iterations.

We applied the RSBN to the Reuters Corpus. This was split into a 794,414 training documents and 10,000 testing document. The size of the corpus allowed an online optimization scheme where mini-batches are observed in an online fashion. The data were preprocessed to contain $M = 10,000$ unique words. To show that the algorithms scale in an online setting, an RSBN with $J = 100$ hidden units was used. Figure 4 (right) shows the results of training this experiment. SSD shows the best performance, and converges well before the other algorithms. After 10,000 iterations, the hold-out perplexity is 1712 from the SSD algorithm, versus 1841 for RMSprop and 2190 for L1R. SGD was uncompetitive for this network size.

VI. Discussion

In this paper we have introduced a novel framework for analyzing discrete graphical models with hidden, unobserved nodes. This is used to develop novel majorization-minimization schemes for several different models, all of which lead to bounds on the Schatten-∞ norm. The form of this bound inspires the use of the SSD algorithm, which for larger models gives orders-of-magnitude improvements in learning efficiency over other standard stochastic gradient techniques. This optimization technique is supported with convergence analysis of the nonlinear stochastic algorithms.

As well as using heuristic schemes based upon decreasing step-sizes, we developed a version of the SSD algorithm, denoted SSD-adapt, which used a constant step size and adapts the minibatch size to approximate the conditions necessary for convergence. This adaptive algorithm either surpasses or matches the effective of the SSD algorithm with heuristic stepsizes in our experiments.

Via the increase in learning efficiency from the SSD algorithm, we not only get increased learning efficiency but also generate state-of-the-art performance for networks of the same size. While the performance of the SBN does not beat the RBM here, the performance differences are greatly decreased. As well, there is a great deal of work improving the gradient estimation in RBMs [25, 37]. Our SBN experiments used a naive Gibbs sampler, but the adaptation of techniques from the RBM or binary sampling schemes [24] could lead to

[4] http://qwone.com/˜jason/20Newsgroups/

great improvements in model optimization and performance. The SSD algorithm combined with future work on gradient estimation may make the SBN modeling performance improve or tie the RBM for large graphical models.

Further, we have demonstrated similar results with topic models, and have shown that it yields state-of-the-art learning efficiency in the topic modeling problems considered here. This demonstrates the broad applicability of the theorems presented in this paper. As well, the analysis has revealed similar properties between the directed and undirected graphical models, as well as similar model performance.

Acknowledgements

The research reported here was funded in part by ARO, DARPA, DOE, NGA and ONR, and in part by the European Commission under grants MIRG-268398 and ERC Future Proof, by the Swiss Science Foundation under grants SNF 200021-146750, SNF CRSII2-147633, and the NCCR Marvel. We thank the reviewers for their helpful comments.

Appendix

A. *Proof of Theorem II.1 and II.2*

We will need the following property of $\#$-operator, whose proof can be found in [11] or [13]:

$$\langle \boldsymbol{s}, \boldsymbol{s}^{\#} \rangle = ||\boldsymbol{s}||_*^2 = ||\boldsymbol{s}^{\#}||^2. \tag{19}$$

By substituting the iteration (7) into the majorization bound and using the property (19), we have

$$\begin{aligned} F_{k+1} &\leq F_k - h_k\langle \nabla F_k, G_k^{\#}\rangle + Ch_k^2||G_k^{\#}||^2 \\ &= F_k - h_k\langle G_k, G_k^{\#}\rangle + Ch_k^2||G_k^{\#}||^2 + h_k\langle w_k, G_k^{\#}\rangle \\ &= F_k - h_k(1-Ch_k)||G_k||_*^2 + h_k\langle w_k, G_k^{\#}\rangle \\ &\leq F_k - h_k(1-Ch_k)||G_k||_*^2 + h_k||w_k||_*||G_k||_* \end{aligned}$$

where the last line follows by Hölder's inequality. By assumption, $||w_k||_* \leq t_k||\nabla F_k||_*$ and therefore $||G_k||_* \geq \frac{1-t_k}{t_k}||w_k||_*$. Using this inequality, we can further simplify the above to

$$F_k - F_{k+1} \geq q(h_k, C, t_k)||G_k||_*^2$$

where

$$q(h, C, t) = h(1-Ch) - \frac{th}{1-t}.$$

Taking expectation (with respect to $w_0, w_1, ..., w_k$) on both sides therefore gives

$$\begin{aligned} \mathbb{E}\left[F_k - F_{k+1}\right] &\geq q(h_k, C, t_k)\mathbb{E}||G_k||_*^2 \\ &\geq q(h_k, C, t_k)||\mathbb{E}\nabla F_k||_*^2 \end{aligned} \tag{20}$$

where we have used Jensen's inequality and $\mathbb{E}w_k = 0$. Now, the condition (8) of the stepsizes ensures that

$$q(h_k, C, t_k) \geq \rho > 0$$

for all k. Therefore, summing up (20) from 0 to $k-1$ gives

$$\mathbb{E}\left[F_0 - F_k\right] \geq \rho\sum_{i=0}^{k-1}||\mathbb{E}\nabla F_i||_*^2$$

Let F^* denote the minimum value of the objective function. Then $F_k \geq F^*$ for all k, which implies

$$\rho\sum_{i=0}^{k-1}||\mathbb{E}\nabla F_i||_*^2 \leq \mathbb{E}\left[F_0 - F_k\right] \leq F_0 - F^* < \infty. \tag{21}$$

Since this relation holds for all k, letting $k \to \infty$ we see that the sum on the left-hand side converges to a finite value, thus implying $\lim_{k\to\infty}||\mathbb{E}\nabla F_k||_* \to 0$. This proves the first assertion of *Theorem II.1.*

To prove the second half, let $\epsilon > 0$ be a given precision. Let $\mathcal{F}_k = \min_{0\leq i\leq k-1}||\mathbb{E}\nabla F_i||_*$. In view of (21),

$$\mathcal{F}_k^2 \leq \frac{1}{k}\sum_{i=0}^{k-1}||\mathbb{E}\nabla F_i||_*^2 \leq \frac{1}{k\rho}(F_0 - F*). \tag{22}$$

To get an upper bound on k to ensure $||\mathbb{E}\nabla F_k||_* < \epsilon$, it suffices, by (22), to solve

$$\sqrt{\frac{1}{k\rho}(F_0 - F*)} < \epsilon$$

which is of rate $k = \mathcal{O}(\frac{1}{\epsilon^2})$. This completes the proof of *Theorem II.1.*

To prove *Theorem II.2*, note that since $F(\cdot)$ is convex, we have, for all k, the following deterministic relation:

$$\begin{aligned} F_k - F^* &\leq \langle \nabla F_k, x_k - x^*\rangle \\ &= \langle G_k, x_k - x^*\rangle - \langle w_k, x_k - x^*\rangle \\ &\leq ||G_k||_*||x_k - x^*|| + ||w_k||_*||x_k - x^*|| \\ &\leq \frac{1}{1-t_k}R||G_k||_* \end{aligned} \tag{23}$$

where we have used $||G_k||_* \geq \frac{1-t_k}{t_k}||w_k||_*$ again. Taking expectation of (23) and substituting into (20), we get

$$\mathbb{E}\left[F_k - F_{k+1}\right] \geq \frac{q(h_k, C, t_k)(1-t_k)^2}{R^2}\left(\mathbb{E}\left[F_k - F^*\right]\right)^2.$$

Denote $\zeta_k = \mathbb{E}\left[F_k - F^*\right]$. Then

$$\begin{aligned} \frac{1}{\zeta_{k+1}} - \frac{1}{\zeta_k} &= \frac{\zeta_k - \zeta_{k+1}}{\zeta_k\zeta_{k+1}} \\ &\geq \frac{q(h_k, C, t_k)(1-t_k)^2}{R^2}\frac{\zeta_k^2}{\zeta_k^2} \\ &= \frac{q(h_k, C, t_k)(1-t_k)^2}{R^2}. \end{aligned} \tag{24}$$

Summing up (24) from 0 to $k-1$ gives

$$\frac{1}{\zeta_k} - \frac{1}{\zeta_0} \geq \frac{1}{R^2}\sum_{i=0}^{k-1}q(h_i, C, t_i)(1-t_i)^2. \tag{25}$$

Since $\nabla F(x^*) = 0$, $F_0 \leq F^* + C||x_0 - x^*||^2 \leq F^* + CR^2$. Combining this with (25) yields

$$\frac{1}{\zeta_k} \geq \frac{1}{R^2}\left[\frac{1}{C} + \sum_{i=0}^{k-1}q(h_i, C, t_i)(1-t_i)^2\right], \tag{26}$$

which completes the proof of *Theorem II.2*.

B. Proof of Theorem III.1

To prove the bound in Theorem III.1, we first consider the data term with a single observation $\boldsymbol{v}$

$$f(\boldsymbol{\theta}) = -\log\sum_{\boldsymbol{h}}\exp(-E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h})), \tag{27}$$

and define the log-sum-exp function as

$$g(\boldsymbol{x}) \quad = \quad \log\sum_{i}\exp(x_i). \tag{28}$$

Define the vector $\boldsymbol{x} \in \mathbb{R}^{|\mathcal{H}|}$ with each entry $x_i = -E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h}_i)$ and $\boldsymbol{y} \in \mathbb{R}^{|\mathcal{H}|}$ with each entry $y_i = -E_{\boldsymbol{\phi}}(\boldsymbol{v},\boldsymbol{h}_i)$, then note by convexity of $g(\cdot)$ that

$$\begin{aligned} g(\boldsymbol{y}) \geq &\quad g(\boldsymbol{x}) + \langle\nabla_{\boldsymbol{x}} g(\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle, \quad \text{or} \\ f(\boldsymbol{\phi}) \leq &\quad f(\boldsymbol{\theta}) - \langle\nabla_{\boldsymbol{x}} g(\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle. \end{aligned} \tag{29}$$

Using the relationships

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) &= -\sum_{i}^{|\mathcal{H}|}\left[\nabla_{\boldsymbol{x}} g(\boldsymbol{x})\right]_i \nabla_{\boldsymbol{\theta}}\left(-E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h}_i)\right), \\ y_i - x_i &= \langle\nabla_{\boldsymbol{\theta}}(-E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h}_i)), \boldsymbol{\phi}-\boldsymbol{\theta})\rangle + r_i, \\ r_i &= -E_{\boldsymbol{\phi}}(\boldsymbol{v},\boldsymbol{h}_i) + E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h}_i) \\ &\quad -\langle\nabla_{\boldsymbol{\theta}}(-E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h}_i)), \boldsymbol{\phi}-\boldsymbol{\theta})\rangle, \end{aligned}$$

where r_i denotes the error from the first order approximation, then

$$\begin{aligned} &\langle\nabla_{\boldsymbol{x}} g(\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle \\ = &\sum_{i}\langle\left[\nabla_{\boldsymbol{x}} g(\boldsymbol{x})\right]_i\left(\langle\nabla_{\boldsymbol{\theta}}(-E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h}_i)), \boldsymbol{\phi}-\boldsymbol{\theta})\rangle + r_i\right) \\ = & -\langle\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}), \boldsymbol{\phi}-\boldsymbol{\theta}\rangle + \sum_{i}^{|\mathcal{H}|}\left[\nabla_{\boldsymbol{x}} g(\boldsymbol{x})\right]_i r_i. \end{aligned} \tag{30}$$

Combining (29) and (30) gives

$$f(\boldsymbol{\phi}) \leq f(\boldsymbol{\theta}) + \langle\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}), \boldsymbol{\phi}-\boldsymbol{\theta}\rangle - \sum_{i}^{|\mathcal{H}|}\left[\nabla_{\boldsymbol{x}} g(\boldsymbol{x})\right]_i r_i. \tag{31}$$

Since $\nabla_{\boldsymbol{x}} g(\boldsymbol{x})$ takes values in a simplex, this is bounded by

$$-\sum_{i}^{|\mathcal{H}|}\left[\nabla_{\boldsymbol{x}} g(\boldsymbol{x})\right]_i r_i \leq \max_{||\boldsymbol{c}||=1}(-\boldsymbol{c}^T\boldsymbol{r}) = -\min_i r_i,$$

giving

$$f(\boldsymbol{\phi}) \leq f(\boldsymbol{\theta}) + \langle\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}), \boldsymbol{\phi}-\boldsymbol{\theta}\rangle - \min_i r_i, \tag{32}$$

which proves the result for a single observation $\boldsymbol{v}$. Union bound is used to extend to multiple observations.

C. Proof of Theorem III.2

To prove Theorem III.2, we analyze the form of

$$\log Z(\boldsymbol{\theta}) = \log\sum_{\boldsymbol{v}}\sum_{\boldsymbol{h}}\exp(-E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h})).$$

First, define the vector $\boldsymbol{x} \in \mathbb{R}^{|\mathcal{H}\times\mathcal{V}|}$ with each entry $x_i = -E_{\boldsymbol{\theta}}(\boldsymbol{v}_i,\boldsymbol{h}_i)$ and $\boldsymbol{y} \in \mathbb{R}^{|\mathcal{H}\times\mathcal{V}|}$ with each entry $y_i = -E_{\boldsymbol{\phi}}(\boldsymbol{v}_i,\boldsymbol{h}_i)$, then using the upper bound on the log-sum-exp function from [13] gives

$$\begin{aligned} g(\boldsymbol{y}) &\leq g(\boldsymbol{x}) + \langle\nabla_{\boldsymbol{x}} g(\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle + \frac{1}{2}||\boldsymbol{y}-\boldsymbol{x}||_\infty^2, \quad \text{or} \end{aligned}$$

$$\begin{aligned} \log Z(\boldsymbol{\phi}) \quad \leq \quad & \log Z(\boldsymbol{\theta}) \\ & +\langle\nabla_{\boldsymbol{x}} g(\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle + \frac{1}{2}||\boldsymbol{y}-\boldsymbol{x}||_\infty^2. \end{aligned} \tag{33}$$

Using the relationships from (29) and (30) with the different sign gives

$$\langle\nabla_{\boldsymbol{x}} g(\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle \leq \langle\nabla_{\boldsymbol{\theta}}\log Z(\boldsymbol{\theta}), \boldsymbol{\phi}-\boldsymbol{\theta}\rangle + \max_i r_i. \tag{34}$$

We can rewrite the infinity norm as

$$||\boldsymbol{y}-\boldsymbol{x}||_\infty^2 = \max_{\boldsymbol{v},\boldsymbol{h}} |-E_{\boldsymbol{\phi}}(\boldsymbol{v},\boldsymbol{h}) - -E_{\boldsymbol{\phi}}(\boldsymbol{v},\boldsymbol{h})|^2. \tag{35}$$

Plugging (34) and (35) into (33) gives the stated result in Theorem III.2. Equation (13) will follow simply because $r_i = 0 \ \ \forall i$.

D. AIS

In the SBN, the log partition function for the full model is explicit and constant, but the log-partition function for the marginal probability on the observations is not. This follows because

$$p_{\boldsymbol{\theta}}(\boldsymbol{v}) = \log\textstyle\sum_{\boldsymbol{h}}\exp(-E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h})) = Z_{\boldsymbol{\theta},\boldsymbol{v}}$$

is unfortunately intractable for large numbers of hidden nodes. However, the ratio between $\frac{Z_{\boldsymbol{\theta},\boldsymbol{v}}}{Z_{\boldsymbol{\phi},\boldsymbol{v}}}$ can be estimated by using the the Annealed Importance Sampler (AIS) [29, 3]. If we let $\mathbf{W}$ associated with $\boldsymbol{\phi}$ be a zero matrix, then the partition function $Z_{\boldsymbol{\phi},\boldsymbol{v}}$ is analytic. Define a set of temperatures $0 = \beta_0 < \beta_1 \cdots < \beta_{K-1} < \beta_K = 1$ with $p_k(\boldsymbol{v},\boldsymbol{h}) = \exp(-\beta_k E_{\boldsymbol{\phi}}(\boldsymbol{v},\boldsymbol{h}) - (1-\beta_k)E_{\boldsymbol{\phi}}(\boldsymbol{v},\boldsymbol{h}))$ and $T_k(\cdot|\cdot)$ denote a Gibbs sweep over $\boldsymbol{h}$ associated with $p_k(\boldsymbol{v},\boldsymbol{h})$, then the necessary conditions for AIS to give asymptotically unbiased estimates are satisfied [29]. The log probability of the observation can be calculated via AIS as in Algorithm 1. We note that this procedure is computationally expensive, but unlike the harmonic mean estimator [28, 21] generates accurate estimates. As well, this is trivially extended to the RAISE method of [30], which addresses the non-asymptotic bias of AIS to give conservative estimates. For the number of samples used, AIS and RAISE gave near-identical performance, supporting the accuracy of the estimator. A similar algorithm was used for the RSBN algorithm.

Algorithm 1 SBN Annealed Importance Sampler

```
Inputs: $\boldsymbol{v}$, $t_0, \ldots, t_K$, $\boldsymbol{\theta}$
Sample $\boldsymbol{h}_0 \sim p_0(\boldsymbol{h})$
$\log\omega = \log p_1(\boldsymbol{v},\boldsymbol{h}_0) - \log p_0(\boldsymbol{v},\boldsymbol{h}_0)$
for $k = 0, \ldots, K-1$ do
    Sample $\boldsymbol{h}_k \sim T_k(\boldsymbol{h}_k|\boldsymbol{h}_{k-1})$
    $\log\omega = \log\omega + \log p_{k+1}(\boldsymbol{v},\boldsymbol{h}_k) - \log p_k(\boldsymbol{v},\boldsymbol{h}_k)$
end for
$\log\hat{p}(\boldsymbol{v}) = \log Z_0 + \log\omega$
```

E. Proofs of Sigmoid Belief Net Results

First, we want to prove (16). Since the SBN log partition function is a constant, we only need to use Theorem III.1. Focusing on the parameter $\mathbf{W}$, the energy function is

$$\begin{aligned} -E_{\mathbf{W}}(\boldsymbol{v},\boldsymbol{h}) &= \mathit{const} + \boldsymbol{v}^T\mathbf{W}\boldsymbol{h} \\ &- \textstyle\sum_{m=1}^{M} \log(1+\exp([\boldsymbol{c}+\mathbf{W}\boldsymbol{h}]_m)). \end{aligned}$$

Because the log partition function is a constant, we need only the term

$$\begin{aligned} -\min_{n,\boldsymbol{h}}(-E_{\mathbf{W}+\mathbf{U}}(\boldsymbol{v}_n,\boldsymbol{h}) + E_{\mathbf{W}}(\boldsymbol{v}_n,\boldsymbol{h}) + \\ \langle \nabla_{\mathbf{W}}(E_{\mathbf{W}}(\boldsymbol{v}_n,\boldsymbol{h})),\mathbf{U}\rangle), \end{aligned}$$

which is equivalent in Taylor's theorem to

$$\begin{aligned} -\min_{n,\boldsymbol{h}}\int_0^1 &\langle \nabla_{\mathbf{W}}E_{\mathbf{W}}(\boldsymbol{v}_n,\boldsymbol{h}) - \nabla_{\mathbf{W}}E_{\mathbf{W}+t\mathbf{U}}(\boldsymbol{v}_n,\boldsymbol{h}),\mathbf{U}\rangle dt \\ &\leq \frac{-1}{2}\min_{n,\boldsymbol{h}}\min_{t\in[0,1]}\langle \frac{d}{dt}\nabla_{\mathbf{W}}E_{\mathbf{W}+t\mathbf{U}}(\boldsymbol{v}_n,\boldsymbol{h}),\mathbf{U}\rangle. \end{aligned}$$

Letting $\hat{\boldsymbol{v}}_t = \sigma(\boldsymbol{c}+(\mathbf{W}+t\mathbf{U})\boldsymbol{h})$, we have

$$\begin{aligned} \nabla_{\mathbf{W}}E_{\mathbf{W}+t\mathbf{U}}(\boldsymbol{v}_n,\boldsymbol{h}) &= (\boldsymbol{v}-\hat{\boldsymbol{v}}_t)\boldsymbol{h}^T, \\ \frac{d}{dt}\nabla_{\mathbf{W}}E_{\mathbf{W}+t\mathbf{U}}(\boldsymbol{v}_n,\boldsymbol{h}) &= (-\hat{\boldsymbol{v}}_t\odot(1-\hat{\boldsymbol{v}}_t)\odot\mathbf{U}\boldsymbol{h})\boldsymbol{h}^T, \end{aligned}$$

which gives

$$\begin{aligned} &\langle\frac{d}{dt}\nabla_{\mathbf{W}}E_{\mathbf{W}+t\mathbf{U}}(\boldsymbol{v}_n,\boldsymbol{h}),\mathbf{U}\rangle \\ \geq\ & \mathrm{tr}((-\hat{\boldsymbol{v}}_t\odot(1-\hat{\boldsymbol{v}}_t)\odot\mathbf{U}\boldsymbol{h})(\mathbf{U}\boldsymbol{h})^T) \\ =\ & \mathrm{tr}(\mathrm{diag}(-\hat{\boldsymbol{v}}_t\odot(1-\hat{\boldsymbol{v}}_t))\mathbf{U}\boldsymbol{h}\boldsymbol{h}^T\mathbf{U}^T) \\ \geq\ & \min_m[-\hat{\boldsymbol{v}}_t\odot(1-\hat{\boldsymbol{v}}_t)]_m\mathrm{tr}(\mathbf{U}\boldsymbol{h}\boldsymbol{h}^T\mathbf{U}^T) \\ \geq\ & -\frac{1}{4}||\mathbf{U}\boldsymbol{h}||_2^2, \end{aligned}$$

and

$$\begin{aligned} &\frac{-1}{2}\min_{n,\boldsymbol{h}}\min_{t\in[0,1]}\langle\frac{d}{dt}\nabla_{\mathbf{W}}E_{\mathbf{W}+t\mathbf{U}}(\boldsymbol{v}_n,\boldsymbol{h}),\mathbf{U}\rangle \qquad (36) \\ \leq\ & -\min_{n,\boldsymbol{h},t\in[0,1]}\left(\frac{-1}{4}||\mathbf{U}\boldsymbol{h}||_2^2\right) \\ \leq\ & \max_{n,\boldsymbol{h},t\in[0,1]}\left(\frac{1}{4}||\mathbf{U}\boldsymbol{h}||_2^2\right) \\ \leq\ & \frac{J}{4}||\mathbf{U}||_{S^\infty}^2. \qquad (37) \end{aligned}$$

The result in (37) is directly applied to get the Schatten-∞ bound for the SBN. The linear bounds on $\boldsymbol{b}$ and $\boldsymbol{c}$ are standard analysis from logistic regression.

F. Proof of Replicated Softmax-Restricted Boltzmann Machine Results

First, we prove (17). The energy function in the RS-RBM is

$$-E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h}) = \boldsymbol{v}^T\boldsymbol{c}+\boldsymbol{v}^T\mathbf{W}\boldsymbol{h}+D\boldsymbol{h}^T\boldsymbol{b}.$$

This is linear, so the only result needed is for (13). For $\mathbf{W}$, note that

$$-\nabla_{\mathbf{W}}E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h}) = \boldsymbol{v}\boldsymbol{h}^T$$

and for perturbation $\mathbf{U}$

$$\begin{aligned} \max_{\boldsymbol{v},\boldsymbol{h}}\langle\boldsymbol{v}\boldsymbol{h}^T,\mathbf{U}\rangle^2 &= \max_{\boldsymbol{v},\boldsymbol{h}}(\boldsymbol{v}^T\mathbf{U}\boldsymbol{h})^2 \\ &\leq \max_{\boldsymbol{v},\boldsymbol{h}}||\boldsymbol{v}||_2^2\ \ ||\boldsymbol{h}||_2^2\ \ ||\mathbf{U}||_{S^\infty}^2 \\ &\leq D^2J||\mathbf{U}||_S^\infty. \end{aligned} \tag{38}$$

The result in (38) is applied to give the result for $\mathbf{W}$. For $\boldsymbol{c}$ with a perturbation $\boldsymbol{a}$,

$$\begin{aligned} \max_{\boldsymbol{v},\boldsymbol{h}}\langle\boldsymbol{v},\boldsymbol{a}\rangle^2 &= \max_{\boldsymbol{v}\in\mathbb{Z}_+,||\boldsymbol{v}||_1=D}(\boldsymbol{v}^T\boldsymbol{a})^2 \\ &= D^2\max_m(a_m)^2 = D^2||\boldsymbol{a}||_\infty^2. \end{aligned} \tag{39}$$

The result on $\boldsymbol{c}$ is shown from (39). For $\boldsymbol{b}$ with perturbation $\boldsymbol{a}$, this is

$$\begin{aligned} \max_{\boldsymbol{v},\boldsymbol{h}}\langle D\boldsymbol{h},\boldsymbol{a}\rangle^2 &= D^2\max_{\boldsymbol{v}\in\mathbb{Z}_+,||\boldsymbol{v}||_1=D}(\boldsymbol{h}^T\boldsymbol{a})^2 \\ &\leq D^2M||\boldsymbol{a}||_2^2. \end{aligned} \tag{40}$$

G. Proofs of Replicated Softmax Belief Net Result

We want to prove the alternative to (17) for the RSBN. Focusing first on $\mathbf{W}$, the energy function for the RSBN is

$$\begin{aligned} -E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h}) &= \mathit{const}+\boldsymbol{v}^T\mathbf{W}\boldsymbol{h} \\ &\quad -D\log\sum_{m=1}^{M}\exp([\boldsymbol{c}+\mathbf{W}\boldsymbol{h}]_m)), \end{aligned}$$

with gradient

$$\begin{aligned} -\nabla_{\mathbf{W}}E_{\mathbf{W}+t\mathbf{U}}(\boldsymbol{h},\boldsymbol{v}) &= (\boldsymbol{v}-D\hat{\boldsymbol{v}}_t)\boldsymbol{h}^T \\ \hat{\boldsymbol{v}}_t &= \mathrm{softmax}(\boldsymbol{c}+(\mathbf{W}+t\mathbf{U})\boldsymbol{h}). \end{aligned}$$

As in Section E, the needed bound is on

$$-\min_{n,\boldsymbol{h},t\in[0,1]}\langle\frac{-d}{dt}\nabla_{\mathbf{W}}E_{\mathbf{W}+t\mathbf{U}}(\boldsymbol{h},\boldsymbol{v}),\mathbf{U}\rangle. \tag{41}$$

With this gradient given by

$$\begin{aligned} &\frac{-d}{dt}\nabla_{\mathbf{W}}E_{\mathbf{W}+t\mathbf{U}}(\boldsymbol{v},\boldsymbol{h}) \\ &= -D(\mathrm{diag}(\hat{\boldsymbol{v}})-\hat{\boldsymbol{v}}\hat{\boldsymbol{v}}^T)\mathbf{U}\boldsymbol{h}\boldsymbol{h}^T. \end{aligned} \tag{42}$$

The inner argument in (41) is simplified

$$\begin{aligned} &\langle\frac{-d}{dt}\nabla_{\mathbf{W}}E_{\mathbf{W}+t\mathbf{U}}(\boldsymbol{h},\boldsymbol{v}),\mathbf{U}\rangle \\ =\ & -D\cdot\mathrm{tr}((\mathrm{diag}(\hat{\boldsymbol{v}}_t)-\hat{\boldsymbol{v}}_t\hat{\boldsymbol{v}}_t^T)\mathbf{U}\boldsymbol{h}\boldsymbol{h}^T\mathbf{U}^T) \\ =\ & -D\cdot(\mathbf{U}\boldsymbol{h})^T((\mathrm{diag}(\hat{\boldsymbol{v}}_t)-\hat{\boldsymbol{v}}_t\hat{\boldsymbol{v}}_t^T)(\mathbf{U}\boldsymbol{h}) \\ \geq\ & -D\cdot||\mathbf{U}\boldsymbol{h}||_2^2\ ||\mathrm{diag}(\hat{\boldsymbol{v}}_t)-\hat{\boldsymbol{v}}_t\hat{\boldsymbol{v}}_t^T||_{S^\infty}. \end{aligned}$$

From Böhning [38], $||\mathrm{diag}(\hat{\boldsymbol{v}}_t)-\hat{\boldsymbol{v}}_t\hat{\boldsymbol{v}}_t^T||_{S^\infty}\in[0,\frac{1}{2}]$. Plugging into (41), we have

$$\begin{aligned} &-\min_{n,\boldsymbol{h},t\in[0,1]}\langle\frac{-d}{dt}\nabla_{\mathbf{W}}E_{\mathbf{W}+t\mathbf{U}}(\boldsymbol{h},\boldsymbol{v}),\mathbf{U}\rangle \\ \leq\ & \frac{D}{2}\max_{\boldsymbol{h}}||\mathbf{U}\boldsymbol{h}||_2^2 \qquad (43) \\ \leq\ & \frac{DJ}{2}||\mathbf{U}||_{S^\infty}^2, \qquad (44) \end{aligned}$$

which gives the result on $\mathbf{W}$. $\boldsymbol{b}$ is a standard result from logistic regression. $\boldsymbol{c}$ with a perturbation of $\boldsymbol{a}$ is given by noting

$$-\frac{d}{dt}\nabla_{\boldsymbol{c}} E_{\boldsymbol{c}+t\boldsymbol{a}}(\boldsymbol{v},\boldsymbol{h}) = D(\text{diag}(\hat{\boldsymbol{v}}) - \hat{\boldsymbol{v}}\hat{\boldsymbol{v}}^T)\boldsymbol{a},$$

and then

$$\begin{aligned} & -\min_{n,\boldsymbol{h},t\in[0,1]} \langle \frac{-d}{dt}\nabla_{\boldsymbol{c}} E_{\boldsymbol{c}+t\boldsymbol{a}}(\boldsymbol{v}_n,\boldsymbol{h}),\boldsymbol{a}\rangle \\ \leq \; & D \max_{n,\boldsymbol{h},t\in[0,1]} \boldsymbol{a}^T(\text{diag}(\hat{\boldsymbol{v}}_t) - \hat{\boldsymbol{v}}_t\hat{\boldsymbol{v}}_t^T)\boldsymbol{a}. \end{aligned}$$

This relates to the log-sum-exp ℓ_∞ bound, and using the proof of Theorem 1 from [13], the upper bound reduces to $D||\boldsymbol{a}||_\infty^2$, or, alternatively, using [38], $D/2||\boldsymbol{a}||_2^2$.

References

[1] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," *arXiv 1206.5533*, Jun. 2012.

[2] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, 2002.

[3] R. Salakhutdinov and G. Hinton, "Replicated Softmax : an Undirected Topic Model," *NIPS*, 2009.

[4] G. Salakhutdinov, R. Hinton, "Deep Boltzmann Machines," *AISTATS*, 2009.

[5] B. J. Frey and G. E. Hinton, "Variational learning in nonlinear gaussian belief networks." *Neural Computation*, 1999.

[6] K. Gregor, C. Blundell, A. Mnih, and D. Wierstra, "Deep AutoRegressive Networks," *ICML*, 2014.

[7] Z. Gan, C. Li, R. Henao, D. Carlson, and L. Carin, "Deep temporal sigmoid belief networks for sequence modeling," *NIPS*, 2015.

[8] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *JMLR*, 2010.

[9] T. Tieleman and Y. LeCun, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning*, 2012.

[10] Y. Nesterov, *Introductory lectures on convex optimization : a basic course*, ser. Applied optimization. Boston, Dordrecht, London: Kluwer Academic Publ., 2004.

[11] J. A. Kelner, Y. T. Lee, L. Orecchia, and A. Sidford, "An Almost-Linear-Time Algorithm for Approximate Max Flow in Undirected Graphs, and its Multicommodity Generalizations," *arXiv 1304.2338*, Apr. 2013.

[12] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems." *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.

[13] D. Carlson, V. Cevher, and L. Carin, "Stochastic Spectral Descent for Restricted Boltzmann Machines," *AISTATS*, 2015.

[14] A. Beck and M. Teboulle, "Mirror descent and nonlinear projected subgradient methods for convex optimization," *Oper. Res. Lett.*, vol. 31, no. 3, pp. 167–175, May 2003.

[15] J. Eckstein, "Nonlinear proximal point algorithms using bregman functions, with applications to convex programming," *Mathematics of Operations Research*, vol. 18, no. 1, pp. 202–226, 1993.

[16] T. Hohage and C. Homann, "A Generalization of the Chambolle-Pock Algorithm to Banach Spaces with Applications to Inverse Problems," *arXiv 1412.0126*, Nov. 2014.

[17] D. Carlson, E. Collins, Y.-P. Hsieh, L. Carin, and V. Cevher, "Preconditioned spectral descent for deep learning," *NIPS*, 2015.

[18] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM J. on Optimization*, vol. 19, no. 4, pp. 1574–1609, Jan. 2009.

[19] R. H. Byrd, G. M. Chin, J. Nocedal, and Y. Wu, "Sample Size Selection in Optimization Methods for Machine Learning," *Math. Programming*, 2012.

[20] L. K. Saul, T. Jaakkola, and M. I. Jordan, "Mean Field Theory for Sigmoid Belief Networks," *Journal of Artificial Intelligence Research*, 1996.

[21] Z. Gan, R. Henao, D. Carlson, and L. Carin, "Learning Deep Sigmoid Belief Networks with Data Augmentation," *AISTATS*, vol. 38, 2015.

[22] R. M. Neal, "Connectionist learning of belief networks," *Artificial Intelligence*, Jul. 1992.

[23] R. Salakhutdinov and I. Murray, "On the Quantitative Analysis of Deep Belief Networks," *ICML*, 2008.

[24] A. Pakman and L. Paninski, "Auxiliary-variable Exact Hamiltonian Monte Carlo Samplers for Binary Distributions," *NIPS*, pp. 1–9, 2013.

[25] M. Welling, "Herding dynamic weights for partially observed random field models," *UAI*, pp. 599–606, Jun. 2009.

[26] A. Mnih and K. Gregor, "Neural Variational Inference and Learning in Belief Networks," *ICML*, 2014.

[27] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic Variational Inference," *JMLR*, 2013.

[28] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno, "Evaluation Methods for Topic Models," in *ICML*, 2009.

[29] R. M. Neal, "Annealed importance sampling," *Statistics and Computing*, 2001.

[30] Y. Burda, R. B. Gross, and R. Salakhutdinov, "Accurate and Conservative Estimates of MRF Log-likelihood using Reverse Annealing," *AISATS*, 2015.

[31] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, 2006.

[32] D. Blei, A. Ng, and M. Jordan, "Latent {D}irichlet allocation," *JMLR*, 2003.

[33] N. Srivastava, R. Salakhutdinov, and G. Hinton, "Modeling Documents with Deep Boltzmann Machines," *UAI*, 2013.

[34] Y. N. Dauphin, H. de Vries, J. Chung, and Y. Bengio, "RMSProp and equilibrated adaptive learning rates for non-convex optimization," *arXiv:1502.04390*, Feb. 2015.

[35] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *arXiv 1212.5701*, p. 6, Dec. 2012.

[36] A. Yuille, "The convergence of contrastive divergences,"

NIPS, 2004.

[37] K. Cho, R. Tapani, and A. Ilin, “Enhanced Gradient and Adaptive Learning Rate for Training Restricted Boltzmann Machines,” *ICML*, 2011.

[38] D. Böhning, “Multinomial logistic regression algorithm,” *Annals of the Institute of Statistical Mathematics*, 1992.

PLACE
PHOTO
HERE

David Carlson received the B.S.E, M.S, and Ph.D. degrees in electrical and computer engineering from Duke University in Durham, NC in 2010, 2014, and 2015, respectively. Since 2015, he has been a postdoctoral researcher in the Department of Statistics at Columbia University, New York, NY. He is a member of the Eta Kappa Nu honor societies.

PLACE
PHOTO
HERE

Ya-Ping Hsieh received my B.S.E. degree in Electrical Engineering in 2010 and an M.S. degree in Communication Engineering in 2012, both from National Taiwan University. He was a research assistant at the Research Center for Information Technology Innovation, Academia Sinica, from 2013 to 2014. Since 2015, he has been a doctoral assistant at EPFL, advised by Prof. Volkan Cevher. He am broadly interested in any theory regarding data analysis, including statistical learning and convex/non-convex optimization. He also enjoys browsing several branches of mathematics, including the information theory and concentration of measure phenomenon.

PLACE
PHOTO
HERE

Edo Collins is currently a Ph.D. student at the Image and Visual Representation Lab at EPFL. He completed his bachelor studies in Computer Science at the Open University of Israel in 2009, and his master studies in Computational Linguistics at the university of Tübingen in 2014.

PLACE
PHOTO
HERE

Lawrence Carin (SM’96 – F’01) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, in 1985, 1986, and 1989, respectively. In 1989, he joined the Department of Electrical Engineering at Polytechnic University, Brooklyn, NY, as an Assistant Professor, where he became an Associate Professor in 1994. In 1995, he joined the Department of Electrical and Computer Engineering (ECE) at Duke University, where he is currently a Professor. He was the Chairman of the Duke ECE department from 2011 to 2014. He held the William H. Younger Distinguished Professorship from 2003 to 2013. He is a co-founder of Signal Innovations Group, Inc., a small business that was acquired by BAE Systems in 2014. Since 2014, he has been the Vice Provost of Research at Duke University. His research interests include machine learning and applied statistics. He has authored over 300 peer-reviewed papers, and is a member of Tau Beta Pi and Eta Kappa Nu honor societies.

PLACE
PHOTO
HERE

Volkan Cevher (SM10) received the B.S. (valedictorian) degree in electrical engineering in 1999 from Bilkent University in Ankara, Turkey, and he received the Ph.D. degree in Electrical and Computer Engineering in 2005 from the Georgia Institute of Technology in Atlanta. He held research scientist positions at the University of Maryland, College Park from 2006 to 2007 and at Rice University in Houston, Texas, from 2008 to 2009. Currently, he is an Assistant Professor at the Swiss Federal Institute of Technology Lausanne with a complimentary appointment at the Electrical and Computer Engineering Department at Rice University. His research interests include signal processing theory, machine learning, graphical models, and information theory. He received a Best Paper Award at SPARS in 2009 and an ERC StG in 2011.