Bias-Variance Trade-Off in Hierarchical Probabilistic Models Using Higher-Order Feature Interactions

Simon Luo*
The University of Sydney
Data61, CSIRO
simon.luo@data61.csiro.au

Mahito Sugiyama
National Institute of Informatics
JST, PRESTO
mahito@nii.ac.jp

Abstract

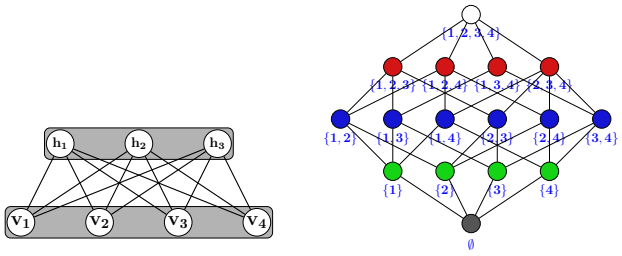
Hierarchical probabilistic models are able to use a large number of parameters to create a model with a high representation power. However, it is well known that increasing the number of parameters also increases the complexity of the model which leads to a bias-variance trade-off. Although it is a classical problem, the bias-variance trade-off between *hidden layers* and *higher-order interactions* have not been well studied. In our study, we propose an efficient inference algorithm for the log-linear formulation of the higher-order Boltzmann machine using a combination of Gibbs sampling and annealed importance sampling. We then perform a bias-variance decomposition to study the differences in *hidden layers* and *higher-order interactions*. Our results have shown that using *hidden layers* and *higher-order interactions* have a comparable error with a similar order of magnitude and using *higher-order interactions* produce less variance for smaller sample size.

Introduction

Hierarchical machine learning models can be used to identify higher-order feature interactions. They include a wide range of models used in machine learning such as graphical models and deep learning because they can be easily generalized for many different applications. Hierarchical models are widely used as they can use a large number of parameters to create a high representative power for modeling interactions between features. However, tuning towards the optimal model includes a classical machine learning problem known as the *bias-variance trade-off* (Friedman, Hastie, and Tibshirani 2001). Despite the prevalence of hierarchical models, the bias-variance trade-off for higher-order feature interactions have not been well studied.

In this paper, we study the differences in using *hidden layers* and *higher-order interactions* to achieve higher representation power in hierarchical models. In our study, we focus on the Boltzmann Machine (BM) (Ackley, Hinton, and Sejnowski 1987), one of the fundamental machine learning models. The family of BMs has been used in a wide range of machine learning models including graphical models and deep learning. The Restricted Boltzmann Machine (RBM) (Hinton 2012) (Figure 1a) and the Higher-Order Boltzmann Machine (HBM) (Sejnowski 1986; Min et al. 2014) (Figure 1b) are fundamental models for learning higher-order feature interactions. However, the two models have different methods for achieving a higher representation power. The RBM is represented by a bipartite graph consisting of two groups, the "visible layer" and the "hidden layer". The visible layer represents the direct observations of the data, while the hidden layer is used to identify latent features. The bias and the edge weights between the visible nodes and hidden nodes are tuned to model the interaction between the features. The HBM is represented using a finite partially ordered set (poset) using a Hasse diagram (Davey and Priestley 2002; Gierz et al. 2003). The poset is used to represent the outcome space of the HBM. A weight is placed on each of the nodes in the outcome space to model the higher-order feature interactions. From a theoretical perspective, both the RBM and HBM are capable models in modeling higher-order feature interactions (Le Roux and Bengio 2008).

In our study, we empirically perform a *bias-variance de-*



(a) (b)

Figure 1: Example of Boltzmann machine modeling high-feature interactions. (a) A Restricted Boltzmann machine with a configuration of 4 visible nodes and 3 hidden nodes. (b) The outcome space of the Higher-Order Boltzmann machine with 4 visible nodes. The green, blue, red and white nodes show first, second, third and fourth order interactions respectively. The bottom node (black) is used to normalize the Boltzmann machine.

*This work was accomplished when the first author was at National Institute of Informatics. The code for our implementation of the HBM is available at https://github.com/sjmluo/HBM

composition for both the RBM and HBM to study the total errors of each model from the trade-off between bias and variance. For our study, we use Contrastive Divergence (CD) as the inferencing technique for the RBM. For the HBM, we use the recent information geometric formulation of the HBM by Sugiyama *et al.* (Sugiyama, Nakahara, and Tsuda 2016; 2017) and use gradient descent to maximize the likelihood. The generalized Pythagorean theorem from information geometry enables us to decompose the total error represented as the Kullback–Leibler (KL) divergence into bias and variance terms. Our analysis uses a synthetic dataset with varying features, samples and model complexity. Our contribution includes: 1) A proposal to use a combination of Gibbs sampling and Annealed Importance Sampling (AIS) in inference to overcome the computational and numerical problems in training the HBM. 2) A study which compares the bias-variance trade-off in *hidden layers* and *higher-order interactions*.

Our results have shown that using *hidden layers* and *higher-order interactions* produce a similar error from the bias and *higher-order interactions* produce less variance for a smaller sample size. For larger datasets, the error from the bias is more dominant, therefore for sufficiently large datasets, *hidden layers* and *higher-order interactions* have a comparable error with a similar order of magnitude.

Formulation

This section presents the hierarchical probabilistic models to analyze the error in modeling higher-order feature interactions. We first introduce the generic Boltzmann machine along with the Restricted Boltzmann Machine (RBM). We then present the information geometry formulation of the log-linear model of hierarchical probability distribution, which includes the family of Boltzmann machines. Finally, we present the Higher-Order Boltzmann Machine (HBM) and our proposed inferencing algorithm.

Boltzmann Machine

A *Boltzmann Machine* (Ackley, Hinton, and Sejnowski 1987) is represented using an undirected graph $G = (V, E)$ with a vertex set $V = \{v_1, v_2, \ldots, v_n, h_1, h_2, \ldots, h_m\}$ and an edge set $E \subseteq \{\{x_i, x_j\} \mid x_i, x_j \in V\}$. Each vertex can be either "visible" or "hidden", where the visible node represents a direct observation from the dataset and the hidden node represents latent features detected in the model. The state of each vertex is represented by $\mathbf{x} = (x_1, x_2, \ldots, x_{n+m}) \in \{0,1\}^{n+m}$, which is a concatenation of $\mathbf{v} \in \{0,1\}^n$ and $\mathbf{h} \in \{0,1\}^m$. The generalized expression for the energy of the joint configuration $(\mathbf{v}, \mathbf{h})$ of the network is defined by:

$$\Phi(\mathbf{x}; \mathbf{b}, \mathbf{w}) = -\sum_{i=1}^{n+m} x_i b_i - \sum_{i,j=1}^{n+m} x_i x_j w_{i,j}, \qquad (1)$$

where the parameters $\mathbf{b} = (b_1, b_2, \ldots, b_{n+m})$ are the biases and $\mathbf{w} = (w_{1,2}, w_{1,3}, \ldots, w_{n+m-1,n+m})$ are the weights which are placed on the vertices and the edges respectively. If $\{i, j\} \notin E$ then $w_{i,j} = 0$ can be used to represent no edge connection between the two vertices. The probability of the current configuration of the entire Boltzmann Machine network G is given by:

$$p(\mathbf{x}; \mathbf{b}, \mathbf{w}) = \frac{\exp\left(-\Phi(\mathbf{x}; \mathbf{b}, \mathbf{w})\right)}{Z}, \qquad (2)$$

where Z is the partition function given by:

$$Z = \sum_{\mathbf{x} \in \{0,1\}^{n+m}} \exp\left(-\Phi(\mathbf{x}; \mathbf{b}, \mathbf{w})\right), \qquad (3)$$

which ensures $\sum_{\mathbf{x} \in \{0,1\}^{n+m}} p(\mathbf{x}; \mathbf{b}, \mathbf{w}) = 1$.

Restricted Boltzmann Machine

The Restricted Boltzmann Machine (RBM) is a BM in which the vertices form a bipartite graph. The two groups of nodes are known as the "visible layer" and the "hidden layer". The visible layer represents the direct observations on the dataset. An activation function is placed on the hidden nodes to model the interactions between features by the following equation:

$$p(h_j = 1 | \mathbf{v}) = \sigma\left(b_j + \sum_i v_i w_{ij}\right),$$

where σ is the sigmoid function. We apply the efficient contrastive divergence technique to adjust the weights and biases to maximize the product of probabilities of generating a given dataset by the BM (Hinton 2002; Tieleman 2008). The updates for the weights and the biases is given by:

$$\Delta \mathbf{W} = \epsilon\left(\mathbf{v}\mathbf{h}^T - \mathbf{v}'\mathbf{h}'^T\right), \qquad \Delta \mathbf{b} = \epsilon\left(\mathbf{x} - \mathbf{x}'\right).$$

Where $\mathbf{v}'$, $\mathbf{h}'$, $\mathbf{x}'$ represents the samples reconstructed from the model and ϵ represents the learning rate.

Additional hidden layers can be added to create the Deep Boltzmann Machine (DBM) to increase the representation power (Salakhutdinov and Hinton 2009; 2012). However, this is not analyzed in our study due to the exponential increase in complexity to compute the partition function in Equation (2).

Information Geometric Formulation of the Log-Linear Model

The information geometric log-linear probabilistic model has been introduced by Amari *et al.* (Amari 2001; Nakahara and Amari 2002; Nakahara, Amari, and Richmond 2006). Further advances in the log-linear formulation by Sugiyama *et al.* (Sugiyama, Nakahara, and Tsuda 2016) has enabled to analytically compute the Fisher information of parameters in hierarchical models. This formulation of the hierarchical model uses a partial order structure to represent the possible model outcomes.

Here we introduce the log-linear formulation introduced in Sugiyama *et al.* (Sugiyama, Nakahara, and Tsuda 2016). Let $(S, \leq)$ be a *partially ordered set* (*poset*) (Gierz et al. 2003), where a *partial order* $\leq$ is the relation between elements in a set S. The *poset* must satisfy the following three properties for $x, y, z \in S$: (1) $x \leq x$ (reflexivity),

(2) $x \leq y, y \leq x \Rightarrow x = y$ (anti-symmetry), and (3) $x \leq y, y \leq z \Rightarrow x \leq z$ (transitivity). We assume that the set S is finite, where $\perp \in S$ and $\perp \leq x, \forall x \in S$. To be concise, we use S^+ to denote $S \setminus \{\perp\}$.

The *zeta function* $\zeta : S \times S \rightarrow \{0, 1\}$ and the *Möbius function* $\mu : S \times S \rightarrow \mathbb{Z}$ are two functions used to construct the partial order structure. The *zeta function* is defined as:

$$\zeta(s, x) = \begin{cases} 1 & \text{if } s \leq x, \\ 0 & \text{otherwise.} \end{cases}$$

The *Möbius function* μ is defined to be the convolution inverse of the *zeta function*, i.e:

$$\mu(x, y) = \begin{cases} 1 & \text{if } x = y, \\ -\sum_{x \leq s < y} \mu(x, s) & \text{if } x < y, \\ 0 & \text{otherwise.} \end{cases}$$

The *log-linear model* on S provides a mapping of the discrete probability distribution to the structured outcome space $(S, \leq)$. Let the probability distribution P denote a probability distribution that assigns a probability $p(x)$ for each $x \in S$ while satisfying $\sum_{x \in S} p(x) = 1$. Each probability $p(x)$ for $x \in S$ is defined as:

$$\begin{aligned} \log p(x) &= \sum_{s \in S} \zeta(s, x)\, \theta(s) = \sum_{s \leq x} \theta(s), \\ \theta(x) &= \sum_{s \in S} \mu(s, x) \log p(s). \end{aligned} \tag{4}$$

$$\begin{aligned} \eta(x) &= \sum_{s \in S} \zeta(x, s)\, p(s), \\ p(x) &= \sum_{s \in S} \mu(x, s)\, \eta(s). \end{aligned} \tag{5}$$

Sugiyama *et al.* (Sugiyama, Nakahara, and Tsuda 2017) has shown that the set of distributions $\mathcal{S} = \{P \mid 0 < p(x) < 1 \text{ and } \sum p(x) = 1\}$ always become a *dually flat Riemannian manifold*. This makes the two functions θ and η a dual coordinate system on $\mathcal{S}$ which is connected through the Legendre transformation.

Higher-Order Boltzmann Machine

Higher order interactions in a Boltzmann machine are capable of modeling higher-order feature interactions. However, they are very rarely used in practice due to the high computational cost for inferencing and learning. The log-linear formulation of the Boltzmann machine provides an elegant representation of the outcome space. This formulation allows any parameters to be included or removed from S^+. For a given Boltzmann machine $S(B) = 2^V$ with $V = \{1, 2, \ldots, n\}$, the energy function of the kth order Boltzmann machine is defined as:

$$\begin{aligned} \Phi(\mathbf{x}; \mathbf{b}, \mathbf{w}) = &- \sum_{i_1 \in V} b_{i_1} x_{i_1} - \sum_{i_1, i_2 \in V} w_{i_1 i_2} x_{i_1} x_{i_2} \\ &- \sum_{i_1, i_2, i_3 \in V} w_{i_1 i_2 i_3} x_{i_1} x_{i_2} x_{i_3} \\ &- \cdots - \sum_{i_1, i_2, \ldots, i_k \in V} w_{i_1, i_2, \ldots, i_k} x_{i_1} x_{i_2} \ldots x_{i_k}, \end{aligned}$$

Sugiyama *et al.* (Sugiyama, Nakahara, and Tsuda 2016; 2017) have shown that the log-linear model can be used to represent the family of Boltzmann Machines. A *submanifold* of $\mathcal{S}$ can be used to represent the set of Gibbs distribution of the BM B given by $\mathcal{S}(B) = \{P \in \mathcal{S} \mid \theta(x) = 0, \forall x \notin B\}$. The Gibbs distribution in Equation (2) directly corresponds to the log-linear model in Equation (4) by:

$$\begin{aligned} \log p(x) &= \sum_{s \in B} \zeta(s, x)\, \theta(s) - \psi(\theta), \\ \psi(\theta) &= -\theta(\perp) = \log Z, \end{aligned} \tag{6}$$

where magnitude of $\theta(x)$ corresponds to the model parameters which model the order of interactions in the Boltzmann machine $B = \{x \in S^+ \mid |x| = 1 \text{ or } x \in E\}$, that is; $\theta(x) = b_i$ if $|x| = 1$ and $\theta(x) = w_{x_{ij}}$ if $|x| = 2$. The log-linear formulation of the Boltzmann machine shown in Equation (6) can be extended to be a kth order Boltzmann machine by $B = \{x \in S^+ \mid |x| \leq k\}$.

Inferencing Algorithm The log-linear formulation of the Boltzmann machine can be trained by minimizing the KL (Kullback-Leibler) divergence to approximate a given empirical distribution $\hat{P}$:

$$\min_{P_B \in \mathcal{S}(B)} D_{\mathrm{KL}}\left(\hat{P}, P_B\right) = \min_{P_B \in \mathcal{S}(B)} \sum_{P_B \in \mathcal{S}(B)} \hat{p}(s) \log \frac{\hat{p}(s)}{p_B(s)}. \tag{7}$$

This is equivalent to maximizing the log-likelihood $L(P_B) = N \sum_{s \in S} \hat{p}(s) \log p_B(s)$. The gradient is obtained as,

$$\begin{aligned} \frac{\partial}{\partial \theta_B(x)} D_{\mathrm{KL}}\left(\hat{P}, P_B\right) &= \frac{\partial}{\partial \theta_B(x)} \sum_{s \in S} \hat{p}(s) \log p_B(s) \\ &= \frac{\partial}{\partial \theta_B(x)} \sum_{s \in S} \left(\hat{p}(s) \sum_{\perp < u \leq s} \theta_B(u) \right) \\ &\quad - \frac{\partial}{\partial \theta_B(x)} \psi(\theta_B) \sum_{s \in S} \hat{p}(s) \\ &= \hat{\eta}(x) - \eta_B(x). \end{aligned}$$

However, $\eta_B(x)$ is computationally expensive to compute because it requires to compute all values of P_B. We propose to use a combination of Gibbs sampling and Annealed Important Sampling (AIS) (Neal 2001; Salakhutdinov 2008) to approximate the distribution of P_B.

Gibbs sampling for η_B Gibbs sampling (Geman and Geman 1984) is a Markov Chain Monte Carlo (MCMC) algorithm which approximates a multivariate probability distribution. It fixes all the other model parameters and updates each of the model parameters one-by-one until convergence:

$$P(\mathbf{x}_i = 1 | \mathbf{x}_{-i}; \theta) = \frac{P(\mathbf{x}; \theta)}{P(\mathbf{x}_{-i}; \theta)} \propto P(\mathbf{x}; \theta),$$

where $\mathbf{x}_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, , x_n)$. We apply Gibbs sampling to generate samples for $\mathbf{x} = (x_1, \ldots, x_n)$. By generating samples, we are able to approximate the un-normalized distribution f^*. The un-normalized probability distribution is proportional to the normalized distribution by a constant Z, i.e. $P = \frac{1}{Z} f^* \propto f^*$. We will later provide a solution using AIS to approximate the normalization constant Z. To update each $\mathbf{x}_i$, we use the difference between the energy functions in each node.

$$\begin{aligned}
&\Delta\Phi_i(\mathbf{x};\theta) = \Phi_i(\mathbf{x}_{x_i=0};\theta) - \Phi_i(\mathbf{x}_{x_i=1};\theta) \\
&= -C\log(P(\mathbf{x}_{x_i=0};\theta)) - (-C\log(P(\mathbf{x}_{x_i=1};\theta))) \\
&= C\log(P(\mathbf{x}_{x_i=1};\theta) - C\log(1 - P(\mathbf{x}_{x_i=1};\theta))),
\end{aligned}$$

where C represents the constant in the Boltzmann distribution. Rearranging the equation to solve for $P(\mathbf{x}_{x_i=1};\theta)$, we have

$$\exp\left(-\frac{\Delta\Phi_i(\mathbf{x};\theta)}{C}\right) = \frac{1 - P(\mathbf{x}_{x_i=1};\theta)}{P(\mathbf{x}_{x_i=1};\theta)},$$

$$P(\mathbf{x}_{x_i=1};\theta) = \frac{\exp(\Delta\Phi_i(\mathbf{x};\theta)/C)}{1 + \exp(\Delta\Phi_i(\mathbf{x};\theta)/C)}.$$

The term with the change in energy is calculated by:

$$\begin{aligned}
&\exp\left(\frac{\Delta\Phi_i(\mathbf{x};\theta)}{C}\right) \\
&= \exp\left(\frac{1}{C}\left[\Phi_i(\mathbf{x}_{\mathbf{x}_i=0};\theta) - \Phi_i(\mathbf{x}_{\mathbf{x}_i=1};\theta)\right]\right) \\
&= \exp\left(-\log P(\mathbf{x}_{\mathbf{x}_i=0};\theta) - (-\log P(\mathbf{x}_{\mathbf{x}_i=1};\theta))\right) \\
&= \exp\left(\sum_{s\in S}\zeta(s,\mathbf{x}_{\mathbf{x}_i=1})\theta(s) - \sum_{s\in S}\zeta(s,\mathbf{x}_{\mathbf{x}_i=0})\theta(s)\right).
\end{aligned}$$

A set of M samples can be generated to approximate η_B by using Equation (5) by empirically estimating the distribution of P_B^*. The overall run-time for each interaction for the Gibbs sampling step is $\mathcal{O}\left(M\left|\mathcal{S}(B)\right|^2\right)$.

Annealed Importance Sampling to Approximate P_B We propose to use AIS (Neal 2001; Salakhutdinov 2008) to overcome the numerical problems in computing the normalization parameter in the partition function Z. By inspecting Equation (3), we can identify a number of problems in computing Z. Firstly, it is clear that the value of Z is extremely large because it takes the sum of the exponential of all energy functions. The large value of Z often creates numerical problems for most implementation. Secondly, computing the energy function $\Phi(x)$ for all nodes is extremely computationally expensive. AIS provides a solution to approximate the value of $\log(Z)$ without having to compute Z or evaluate the energy function $\Phi(x)$.

AIS approximates the normalization constant by tracking the gradual changes of an MCMC transition $T_k(\mathbf{x}_{n+1}|\mathbf{x}_n)$ operation such as Gibbs sampling. AIS uses a sequence of intermediate probability distributions to evaluate the importance weight $w_{\text{AIS}}^{(i)}$ which is an estimation of the ratio between the first and last distribution.

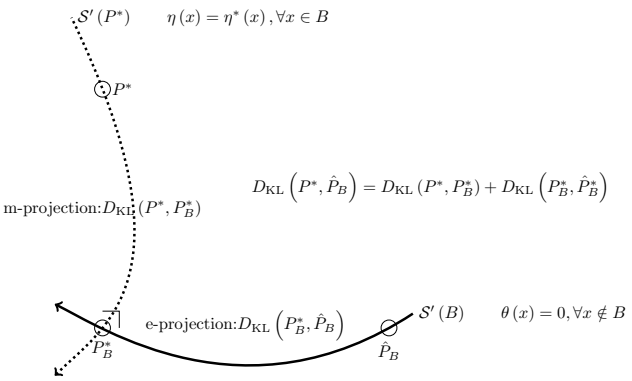


Figure 2: An illustration of the decomposition of the bias and variance.

For our study, we use one of the most prevalent methods to generate a sequence of intermediate probability distributions for $k = 0, \ldots, K$ by using the following geometric property,

$$f_k(x) \propto f_0^*(x)^{1-\beta_k} f_k^*(x)^{\beta_k}$$

where $0 = \beta_0 < \beta_1 < \ldots < \beta_K = 1$. There have been several other more advanced techniques to model the path of the intermediate distributions (Grosse, Maddison, and Salakhutdinov 2013). However, this is not the focus of our study and can be subjected to further study in future work. The AIS weight w_{AIS} can be calculated by using

$$w_{\text{AIS}}^{(i)} = \frac{f_1^*(x_1)}{f_0^*(x_1)}\frac{f_2^*(x_2)}{f_1^*(x_2)} \cdots \frac{f_{K-1}^*(x_{K-1})}{f_{K-2}^*(x_{K-1})}\frac{f_K^*(x_K)}{f_{K-1}^*(x_K)},$$

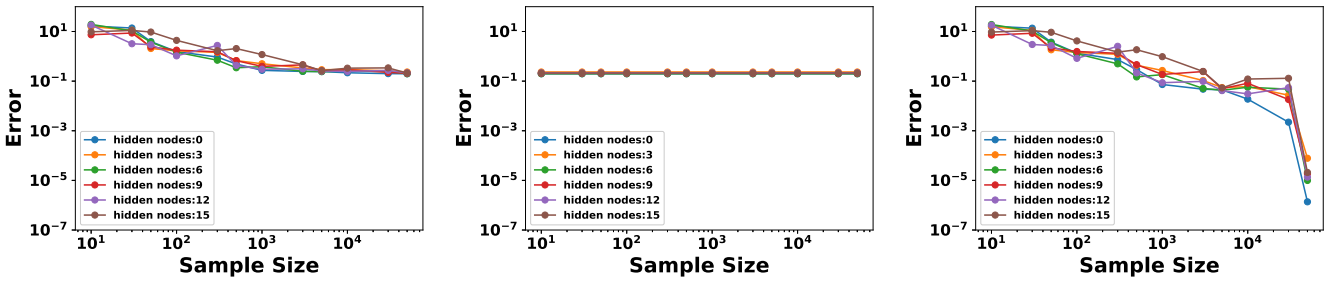
where f^* denotes a function to calculate the un-normalized probability distribution. After completing M runs of AIS, the ratio of the first and final constant of the partition function can be estimated as

$$\frac{Z_K}{Z_0} \approx \frac{1}{M}\sum_{i=1}^{M} w_{\text{AIS}}^{(i)} = \hat{r}_{\text{AIS}} \tag{8}$$

Neal (Neal 2001; 2005) has theoretically shown that the $\text{Var}(\hat{r}_{\text{AIS}}) \propto 1/MK$. For practical implementations Equation (8) should be in log scale to avoid numerical problems. The standard form is shown here for conciseness. From Equation (8), the final $\log Z$ can be estimated without computing Z if Z_0 is known. Then $\log Z_0$ can be calculated efficiently if we initialize P_B uniformly, i.e. for HBM, $\theta(\perp) = -\log Z = N\log(2)$ and $\theta(x) = 0, \forall x \in \mathcal{S}^+$.

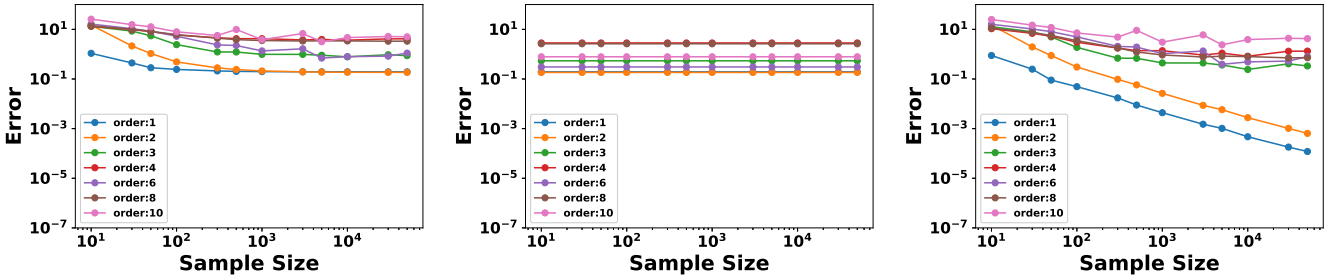
Experiments

Here we present the main results of the paper. This section presents the formulation of the *bias-variance decomposition*, the set-up of the experiment and the experimental results and discussion.



(a) Total Error (b) Bias (c) Variance

Figure 3: Empirical evaluation of the error generated from the bias and variance of the RBM



(a) Total Error (b) Bias (c) Variance

Figure 4: Empirical evaluation of the error generated from the bias and variance for the HBM

Bias-Variance Decomposition

We use a *bias-variance decomposition* to compare the behavior of the higher-order feature interactions between the RBM and HBM by varying the complexity of the model. We focus on the expectation of the KL divergence $\mathbf{E}[D_{\mathrm{KL}}(P^*, \hat{P}_B)]$ from the true (unknown) distribution P^* to the maximum likelihood estimation (MLE) $\hat{P}_B$ of the empirical distribution $\hat{P}$ by a Boltzmann machine with the parameter set B. This term represents the total error in the model accumulated from the bias and variance in the model.

The KL divergence for probabilities in the exponential family can be decomposed into bias and variance using its information geometric properties. We can calculate the true variance of the model by replacing $\hat{P}$ with P^* in Equation (7). The bias and variance can be separated into two components which are orthogonal as illustrated in Figure 2. Using this decomposition of the bias and variance, the total error in the model can be calculated using the *Generalized Pythagorean Theorem*,

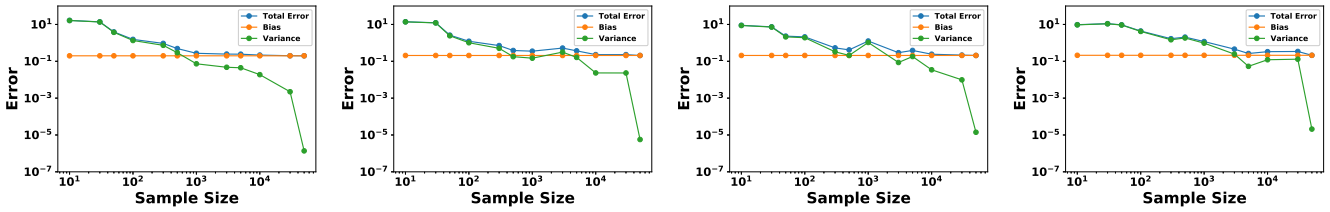
$$\begin{aligned}
&\mathbf{E}\left[D_{\mathrm{KL}}\left(P^*, \hat{P}_B\right)\right] \\
&= \mathbf{E}\left[D_{\mathrm{KL}}\left(P^*, P_B^*\right)\right] + \mathbf{E}\left[D_{\mathrm{KL}}\left(P_B^*, \hat{P}_B\right)\right] \\
&= D_{\mathrm{KL}}\left(P^*, P_B^*\right) + \mathbf{E}\left[D_{\mathrm{KL}}\left(P_B^*, \hat{P}_B\right)\right] \\
&= \underbrace{D_{\mathrm{KL}}\left(P^*, P_B^*\right)}_{\text{bias}} + \underbrace{\mathrm{var}\left(P_B^*, B\right)}_{\text{variance}}.
\end{aligned}$$

Experiment Setup

A synthetic dataset is generated to study the bias-variance trade-off in HBM and RBM. The synthetic data is created by drawing a random probability $[0, 1]$ from a uniform distribution for each $P^* \in \mathcal{P}^*$ such that $\sum_{P^* \in \mathcal{P}^*} P^* = 1$, where $\mathcal{P}^*$ represents the set of probabilities for all possible feature combinations. A sample size of $N = \{\, 1\times10, 3\times10, 5\times10, 1\times10^2, 3\times10^2, 5\times10^2, 1\times10^3, 3\times10^3, 5\times10^3, 1\times10^4, 3\times10^4, 5\times10^4 \,\}$ are drawn from the descrete probability distribution $\mathcal{P}^*$ using a multinomial. For each sample size N, we create 24 independent datasets to be used for both the HBM and RBM. The MLE of the HBM is calculated analytically by directly placing the true probability distribution P^* into the model. While, for the RBM, it is not analytically tractable to calculate the MLE, it is instead approximated by placing a dataset several orders of magnitude larger than the experimental dataset, in our case, we have generated a dataset with the sample size of 1×10^6 and have assumed this to be the MLE. Both the HBM and the RBM has been run using $10,000$ Gibbs samples, a learning rate of 0.1 and $10,000$ iterations.

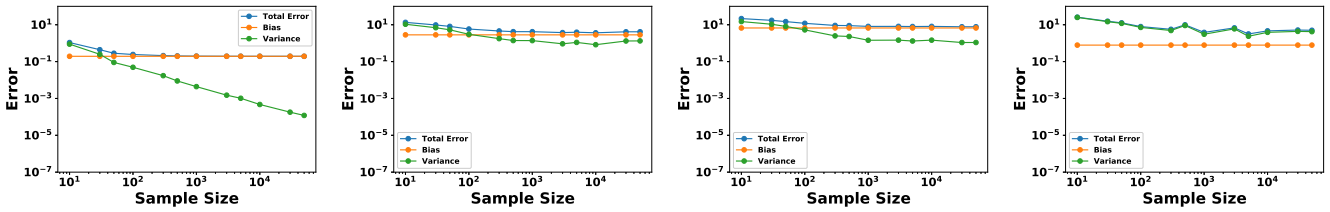
Experiment Results

The empirical evaluation in Figure 3 and Figure 4 have shown that both RBM and HBM have shown similar trends, with a positive correlation between the number of model parameters and variance and an inverse relationship between the sample size and variance. Surprisingly, the bias does not



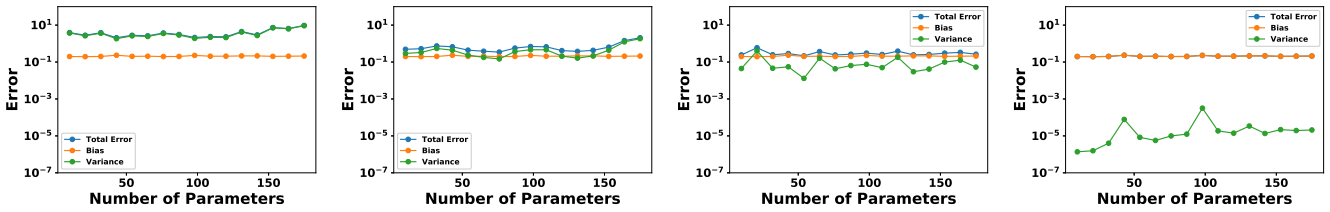
(a) 0 Hidden Nodes (b) 5 Hidden Nodes (c) 10 Hidden Nodes (d) 15 Hidden Nodes

Figure 5: Empirical evaluation of the error generated from the bias and variance for varying hidden nodes in the RBM



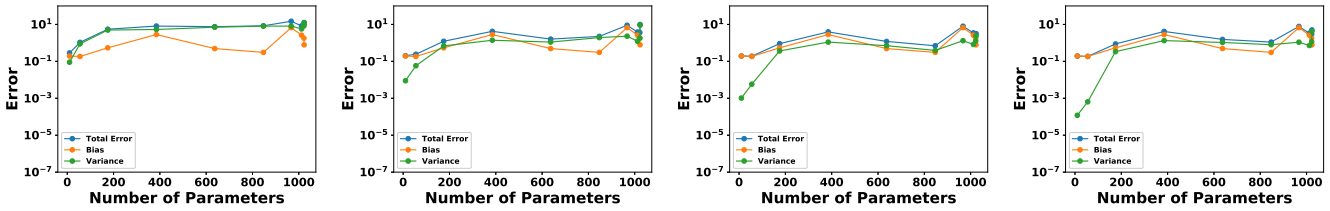
(a) Order 1 (b) Order 4 (c) Order 7 (d) Order 10

Figure 6: Empirical evaluation of the error generated from the bias and variance for varying order of interactions in the HBM



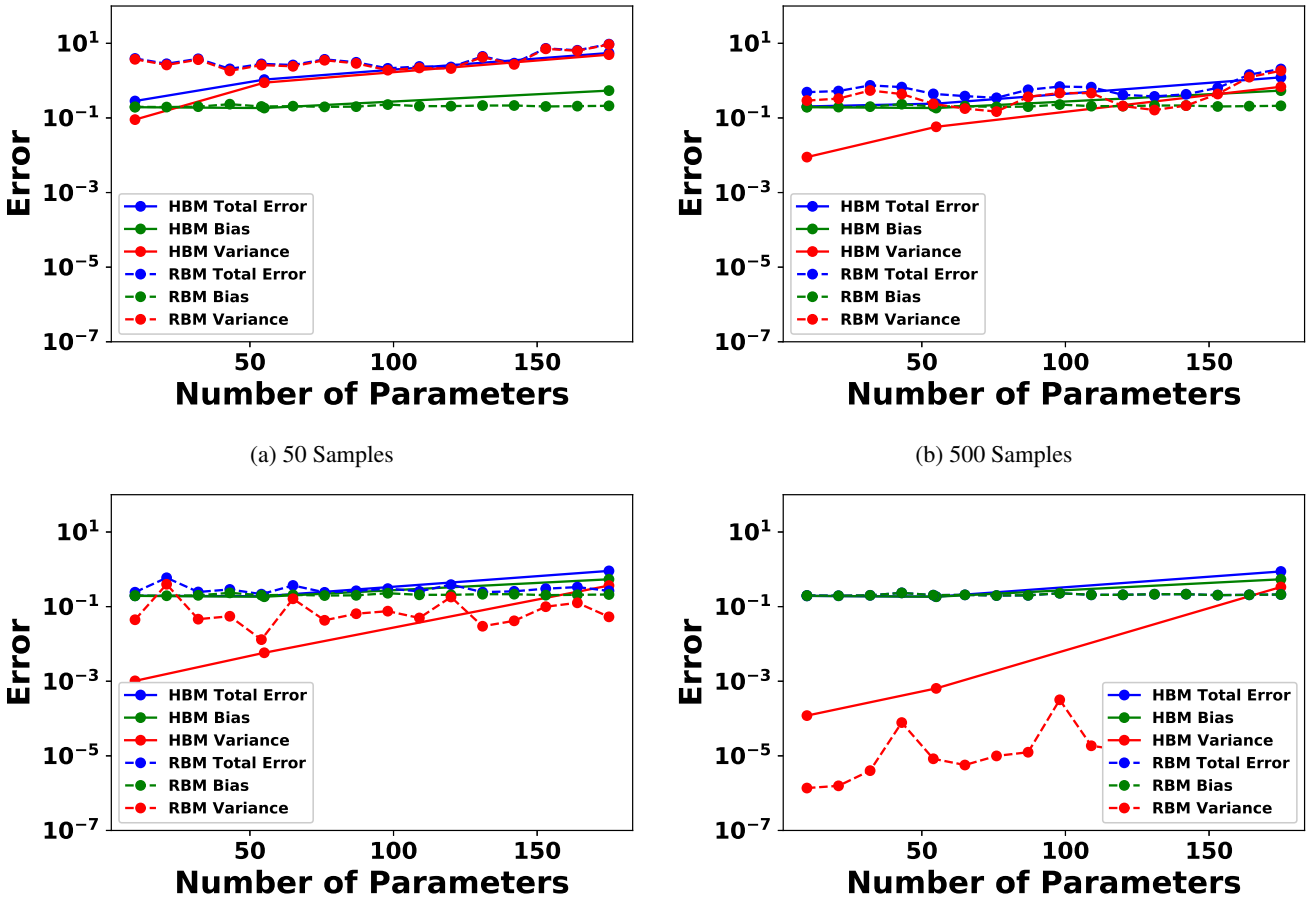
(a) 50 Samples (b) 500 Samples (c) 5000 Samples (d) 50000 Samples

Figure 7: Empirical evaluation of the error generated from the bias and variance for varying sample size in the RBM



(a) 50 Samples (b) 500 Samples (c) 5000 Samples (d) 50000 Samples

Figure 8: Empirical evaluation of the error generated from the bias and variance for varying sample size in the HBM



(a) 50 Samples

(b) 500 Samples

(c) 5000 Samples

(d) 50000 Samples

Figure 9: Comparing empirical error in model for the HBM with RBM against the number of model parameters

show any clear correlation between the number of model parameters and the sample size.

The error generated from the variance is much more dominant for a smaller sample size and a larger number of model parameters. Comparing Figure 5 and Figure 6, the RBM has shown to be more effective at reducing the variance with a larger sample size. The importance sampling used to estimate the partition function in HBM may have led to the higher variance in empirical results for the HBM. Figure 7 and Figure 8 shows a positive correlation between the model parameters and variance. The HBM shows to have a larger correlation between the model parameters and variance.

The total error in the model is the sum of the bias and the variance. The total number of model parameters is a natural way to compare the total error generated by the RBM and HBM (i.e. $|\mathbf{b}| + |\mathbf{w}|$ for RBM and $|\mathcal{S}(B)|$ for HBM). Figure 9 shows that for small sample size, HBM has shown to have produced a lower error in the model. The higher error in the RBM is generated by the larger variance. For a larger sample size, the error from the bias is more much dominate. Since the bias in both the RBM and HBM is in a similar order of magnitude, both models have a total error in the same order of magnitude. RBM has shown to be much more effective at reducing the variance with a larger sample size, however, this does not reduce the total error significantly because it is several orders of magnitude smaller than the bias.

Conclusion

In this paper, we have first proposed using a combination of Gibbs sampling and importance sampling to overcome the computational issues in training the information geometric formulation of the higher-order Boltzmann machine (HBM). The experimental results have shown that our proposed approach is effective in estimating the probability distribution of the model. Our proposed approach has been compared with the RBM to compare using *hidden layers* and *higher-order interactions* to model higher-order feature interactions. Our experimental results have shown that both models have produced a total error with similar orders of magnitude and using *higher-order interactions* may be more effective at minimizing the variance for smaller sample size.

References

Ackley, D. H.; Hinton, G. E.; and Sejnowski, T. J. 1987. A learning algorithm for Boltzmann machines. In *Readings in Computer Vision*. Elsevier. 522–533.

Amari, S. 2001. Information geometry on hierarchy of probability distributions. *IEEE Transactions on Information Theory* 47(5):1701–1711.

Davey, B. A., and Priestley, H. A. 2002. *Introduction to Lattices and Order*. Cambridge University Press.

Friedman, J.; Hastie, T.; and Tibshirani, R. 2001. *The Elements of Statistical Learning*. Springer.

Geman, S., and Geman, D. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6(6):721–741.

Gierz, G.; Hofmann, K. H.; Keimel, K.; Lawson, J. D.; Mislove, M.; and Scott, D. S. 2003. *Continuous Lattices and Comains*, volume 93. Cambridge University Press.

Grosse, R. B.; Maddison, C. J.; and Salakhutdinov, R. R. 2013. Annealing between distributions by averaging moments. In *Advances in Neural Information Processing Systems (NIPS)*, 2769–2777.

Hinton, G. E. 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation* 14(8):1771–1800.

Hinton, G. E. 2012. A practical guide to training restricted Boltzmann machines. In *Neural Networks: Tricks of the Trade*. Springer. 599–619.

Le Roux, N., and Bengio, Y. 2008. Representational power of restricted Boltzmann machines and deep belief networks. *Neural Computation* 20(6):1631–1649.

Min, M. R.; Ning, X.; Cheng, C.; and Gerstein, M. 2014. Interpretable sparse high-order Boltzmann machines. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 614–622.

Nakahara, H.; Amari, S.; and Richmond, B. J. 2006. A comparison of descriptive models of a single spike train by information-geometric measure. *Neural Computation* 18(3):545–568.

Nakahara, H., and Amari, S. 2002. Information-geometric measure for neural spikes. *Neural Computation* 14(10):2269–2316.

Neal, R. M. 2001. Annealed importance sampling. *Statistics and Computing* 11(2):125–139.

Neal, R. M. 2005. Estimating ratios of normalizing constants using linked importance sampling. *arXiv:math/0511216*.

Salakhutdinov, R., and Hinton, G. E. 2009. Deep Boltzmann machines. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 448–455.

Salakhutdinov, R., and Hinton, G. E. 2012. An efficient learning procedure for deep Boltzmann machines. *Neural Computation* 24(8):1967–2006.

Salakhutdinov, R. 2008. Learning and evaluating Boltzmann machines. *Technical Report UTML TR 2008-002, Department of Computer Science, University of Toronto*.

Sejnowski, T. J. 1986. Higher-order Boltzmann machines. In *AIP Conference Proceedings*, volume 151, 398–403. AIP.

Sugiyama, M.; Nakahara, H.; and Tsuda, K. 2016. Information decomposition on structured space. In *2016 IEEE International Symposium on Information Theory (ISIT)*, 575–579. IEEE.

Sugiyama, M.; Nakahara, H.; and Tsuda, K. 2017. Tensor balancing on statistical manifold. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70, 3270–3279.

Tieleman, T. 2008. Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, 1064–1071.