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

Simon Luo*
The University of Sydney
Data61, CSIRO
sluo4225@uni.sydney.edu.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].

*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
Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
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] Sugiyama, Nakahara, and Tsuda 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 $E \subseteq \{\{x, y\} | x, y \in V\}$. 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 $x = (x_1, x_2, \ldots, x_{n+m}) \in \{0,1\}^{n+m}$, which is a concatenation of $v \in \{0,1\}^n$ and $h \in \{0,1\}^m$. The generalized expression for the energy of the joint configuration $(v, h)$ of the network is defined by:

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

where the parameters $b = (b_1, b_2, \ldots, b_{n+m})$ are the biases and $w = (w_{1,2}, w_{1,3}, \ldots, w_{n+m-1,n+m})$ 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:

$$p(x; b, w) = \frac{\exp(-\Phi(x; b, w))}{Z}, \quad (2)$$

where $Z$ is the partition function given by:

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

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

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:

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

where $\sigma$ 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:

$$\Delta W = \epsilon (v h^T - v' h'^T), \quad \Delta b = \epsilon (x - x').$$

Where $v'$, $h'$, $x'$ represents the samples reconstructed from the model and $\epsilon$ 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; Salakhutdinov and Hinton 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),
The Boltzmann machine is defined as:

\[ S \] 

which allows any parameters to be included or removed from the model, \( S^+ \) to denote \( S \setminus \{ \perp \} \).

The zeta function \( \zeta : S \times S \to \{0,1\} \) and the Möbius function \( \mu : S \times S \to \mathbb{Z} \) 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 \( \mu \) is defined to be the convolution inverse of the zeta function, i.e:

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

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 the constraint \( \sum_{x \in S} p(x) = 1 \). Each probability \( p(x) \) for \( x \in S \) is defined as:

\[
\log p(x) = \sum_{s \in S} \zeta(s,x) \theta(s) = \sum_{s \leq x} \theta(s),
\]

\[
\theta(x) = \sum_{s \in S} \mu(s,x) \log p(s),
\]

\[
\eta(x) = \sum_{s \in S} \zeta(s,x) p(s),
\]

\[
p(x) = \sum_{s \in S} \mu(s,x) \eta(s).
\]

Sugiyama et al. (2016) have shown that the set of distributions \( S \) with \( 0 \leq \theta(x) < 1 \) and \( \sum p(x) = 1 \) always become a dually flat Riemannian manifold. This makes the two functions \( \theta \) and \( \eta \) a dual coordinate system on \( S \) 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 inference 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 \( k \)th order Boltzmann machine is defined as:

\[
\Phi(x; b, w) = -\sum_{i_1 \in V} b_{i_1} x_{i_1} - \sum_{i_1, i_2 \in V} w_{i_1 i_2} x_{i_1} x_{i_2} - \sum_{i_1, i_2, i_3 \in V} w_{i_1 i_2 i_3} x_{i_1} x_{i_2} x_{i_3} - \cdots - \sum_{i_1, i_2, \ldots, i_k \in V} w_{i_1 i_2 \ldots i_k} x_{i_1} x_{i_2} \cdots x_{i_k}.
\]

Gibbs sampling for \( \eta_B \) 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 \( P_B \).
where \( x_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \). We apply Gibbs sampling to generate samples for \( x = (x_1, \ldots, x_n) \).

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. \( P = \frac{Z}{f^*} \propto f^* \). We will later provide a solution using AIS to approximate the normalization constant \( Z \). To update each \( x_i \), we use the difference between the energy functions in each node.

\[
\Delta \Phi_i (x; \theta) = \Phi_i (x_{x_i = 0}; \theta) - \Phi_i (x_{x_i = 1}; \theta)
\]

\[
= -C \log (P (x_{x_i = 0}; \theta)) - (-C \log (P (x_{x_i = 1}; \theta)))
\]

\[
= C \log (P (x_{x_i = 1}; \theta)) - C \log (1 - P (x_{x_i = 1}; \theta)),
\]

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

\[
\exp \left( \frac{\Delta \Phi_i (x; \theta)}{C} \right) = \frac{1 - P (x_{x_i = 1}; \theta)}{P (x_{x_i = 1}; \theta)},
\]

\[
P (x_{x_i = 1}; \theta) = \frac{\exp (\Delta \Phi_i (x; \theta) / C)}{1 + \exp (\Delta \Phi_i (x; \theta) / C)}.
\]

The term with the change in energy is calculated by:

\[
\exp \left( \frac{\Delta \Phi_i (x; \theta)}{C} \right) = \exp \left( \frac{1}{C} [\Phi_i (x_{x_i = 0}; \theta) - \Phi_i (x_{x_i = 1}; \theta)] \right)
\]

\[
= \exp (- \log P (x_{x_i = 0}; \theta) - (- \log P (x_{x_i = 1}; \theta)))
\]

\[
= \exp \left( \sum_{s \in S} \zeta(s, x_{x_i = 1}) \theta(s) - \sum_{s \in S} \zeta(s, x_{x_i = 0}) \theta(s) \right).
\]

A set of \( M \) samples can be generated to approximate \( \eta_B \) by using Equation (9) by empirically estimating the distribution of \( P_B \). The overall run-time for each interaction for the Gibbs sampling step is \( O \left( M |S(B)|^2 \right) \).

**Annealed Importance Sampling to Approximate \( P_B \)**

We propose to use AIS (Neal 2001; Salakhutdinov 2008) to overcome the numerical problems in computing the partition function parameter in the partition function \( Z \). By inspecting Equation (8), 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 \( T_k (x_{x_i + 1} | x_n) \) 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.

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,

\[
f_k (x) \propto f_0 (x)^{1 - \beta_k} \prod_{k=1}^{K} f_k (x)^{\beta_k}
\]

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 \( w_{\text{AIS}} \) can be calculated by using

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

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

\[
\hat{Z}_K \approx \frac{1}{M} \sum_{i=1}^{M} w_{\text{AIS}}^{(i)} = \hat{r}_{\text{AIS}}
\]

Neal (Neal 2001; Neal 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 (10), 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, \( \theta (\bot) = -\log Z = N \log (2) \) and \( \theta (x) = 0, \forall x \in S^+ \).

**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.
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 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 $P$ 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,

$$
E \left[ D_{KL} \left( P^*, \hat{P}_B \right) \right] = E \left[ D_{KL} \left( P^*, \hat{P}_B \right) \right] + E \left[ D_{KL} \left( \hat{P}_B, \hat{P}_B \right) \right]
$$

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

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 $N = \{1 \times 10, 3 \times 10, 5 \times 10, 1 \times 10^2, 3 \times 10^2, 5 \times 10^2, 1 \times 10^3, 3 \times 10^3, 5 \times 10^3, 1 \times 10^4, 3 \times 10^4, 5 \times 10^4\}$ are drawn from the discrete probability distribution $P^*$ 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 $1 \times 10^6$ 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
Figure 5: Empirical evaluation of the error generated from the bias and variance for varying hidden nodes in the RBM

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

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

Figure 8: Empirical evaluation of the error generated from the bias and variance for varying sample size in the HBM
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. $|b| + |w|$ for RBM and $|S(B)|$ 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, Hinton, and Sejnowski 1987] 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 2001] Amari, S. 2001. Information geometry on hierarchy of probability distributions. IEEE Transactions on Information Theory 47(5):1701–1711.

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

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

[Geman and Geman 1984] 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 et al. 2003] 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, Maddison, and Salakhutdinov 2013] 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 2002] Hinton, G. E. 2002. Training products of experts by minimizing contrastive divergence. Neural Computation 14(8):1771–1800.

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

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

[Min et al. 2014] 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, Amari, and Richmond 2006] 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 and Amari 2002] Nakahara, H., and Amari, S. 2002. Information-geometric measure for neural spikes. Neural Computation 14(10):2269–2316.

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

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