Interpretable Sparse High-Order Boltzmann Machines

Martin Renqiang Min
NEC Labs America

Xia Ning
NEC Labs America

Chao Cheng
Dartmouth College

Mark Gerstein
Yale University

Abstract

Fully-observable high-order Boltzmann Machines are capable of identifying explicit high-order feature interactions theoretically. However, they have never been used in practice due to their prohibitively high computational cost for inference and learning. In this paper, we propose an efficient approach for learning a fully-observable high-order Boltzmann Machine based on sparse learning and contrastive divergence, resulting in an interpretable Sparse High-order Boltzmann Machine, denoted as `SHBM`. Experimental results on synthetic datasets and a real dataset demonstrate that `SHBM` can produce higher pseudo-log-likelihood and better reconstructions on test data than the state-of-the-art methods. In addition, we apply `SHBM` to a challenging bioinformatics problem of discovering complex Transcription Factor interactions. Compared to conventional Boltzmann Machine and directed Bayesian Network, `SHBM` can identify much more biologically meaningful interactions that are supported by recent biological studies. To the best of our knowledge, `SHBM` is the first working Boltzmann Machine with explicit high-order feature interactions applied to real-world problems.

1 Introduction

Identifying high-order feature interactions is an important problem in machine learning and effective solutions to this problem have a large set of use scenarios. Particularly, in biomedical applications, interactions among multiple proteins play critical roles in many biological processes, and thus the identification of such high-order interactions itself becomes critical. Theoretically, fully-observable high-order Boltzmann Machines (HBM) [24] are capable of identifying explicit high-order feature interactions. However, they have never been applied to any real problems because they have too many energy terms even for a fair number of features and the learning procedure is prohibitively slow.

In this paper, we propose an interpretable Sparse High-order Boltzmann Machine, denoted as `SHBM`, and an efficient learning algorithm to learn an `SHBM` for Big-Data problems. We extend the energy function of an HBM as in [24] to have a combination of different orders of feature interactions up to a maximum order allowed. We introduce sparsity constraints on the feature interaction terms so as to construct a sparse model. The learning algorithm for `SHBM` is decoupled into two steps: high-order interaction neighborhood estimation and interaction weight learning. We propose an efficient sparse high-order logistic regression method, denoted as `shooter`, for identifying interpretable high-order feature interactions and thus to determine the energy function of an `SHBM`. The `shooter` method greedily explores the structures among feature interactions via solving a set of ℓ_1-regularized logistic regression problems. Significant speed-up is enabled by organizing the search space within a tree structure as well as a block-wise expansion of the possible interactions conforming to the tree. Given the energy function determined by `shooter`, we propose different sampling algorithms that scale to large number of features and interactions in order to finally learn the interaction weights within an `SHBM`. Our experiments on both large synthetic and real datasets demonstrate that `SHBM` and its sub-routine `shooter` can effectively identify problem-inherent high-order feature interactions in large-scale settings, which has great potential to be applied to many Big-Data problems.

The paper is organized as follows. In Section 2, we present the interpretable sparse high-order Boltzmann Machines. In Section 3, we present the `shooter` method for identifying high-order feature interactions within an `SHBM`. In Section 4, we present sampling

Appearing in Proceedings of the 17^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33. Copyright 2014 by the authors.

methods for solving SHBM. In Section 6, experimental results are presented. Section 7 concludes the paper with some discussions and future work.

2 Sparse High-order Boltzmann Machines

2.1 Review of Boltzmann Machines and High-order Boltzmann Machines

In this section, we review traditional fully observable Boltzmann Machines (BMs) and High-order BM (HBMs). A fully-observable BM [1] is an undirected graphical model with symmetric connections between p visible units $\mathbf{v} \in \{0, 1\}$. The joint probability distribution of a configuration $\mathbf{v}$ is defined as follows:

$$p(\mathbf{v}) = \frac{1}{Z} exp(-E(\mathbf{v})), \tag{1}$$

where $Z = \sum_{\mathbf{u}} \exp(-E(\mathbf{u}))$ is the partition function. The energy $E(\mathbf{v})$ is defined as

$$-E(\mathbf{v}) = \sum_{ij} W_{ij} v_i v_j + \sum_i b_i v_i, \tag{2}$$

where b_i is the bias on unit v_i, and W_{ij} is the connection weight between unit v_i and v_j. The weights are updated via maximizing the log-likelihood of the observed input data using the following gradient descent

$$\Delta W_{ij} = \epsilon(\langle v_j v_j \rangle_{\text{data}} - \langle v_i v_j \rangle_{\infty}),$$

where ϵ is the learning rate, $\langle\cdot\rangle_{\text{data}}$ is the expectation with respect to the data distribution and $\langle\cdot\rangle_{\infty}$ is the expectation with respect to the model distribution.

BMs are conventionally used to model pairwise interactions between input features. They have been extended in [24] to model high-order interactions by incorporating higher-order energy functions. For example, the quadratic energy function as in Equation 2 can be replaced by a sum of energy functions with orders from 1 to m as follows:

$$-E(\mathbf{v}) = \sum_{j=1}^{m} \sum_{i_1 i_2 \cdots i_j} W_{i_1 i_2 \cdots i_j} v_{i_1} v_{i_2} \cdots v_{i_j},$$

where $W_{i_1 i_2 \cdots i_j}$ is the weight for the order-j interaction among units v_{i_1}, v_{i_2}, $\cdots$, and v_{i_j}. The derived model is the so-called High-order Boltzmann Machine (HBM), and its learning rule with respect to order-j interactions correspondingly becomes

$$\Delta W_{i_1 i_2 \cdots i_j} = \epsilon(\langle v_{i_1} v_{i_2} \cdots v_{i_j} \rangle_{\text{data}} - \langle v_{i_1} v_{i_2} ... v_{i_j} \rangle_{\infty}). \tag{3}$$

However, due to the painfully slow Gibbs Sampling procedure to get samples from the model distribution, HBMs have never been applied to any interesting practical problems.

2.2 Sparse High-Order Boltzmann Machines

In this section, we propose our methods for solving HBMs with sparsity constraint. In practice, it is typically infeasible for HBMs to include all possible energy functions of different orders. Thus, we need to perform structure learning, which is a challenging task for high-dimensional discrete graphical models. Following [17], the structure learning of HBMs could be conducted by minimizing the following ℓ_1-regularized negative log-likelihood

$$\min_{\mathbf{W}} E(\mathbf{v}) + \log Z + \lambda \|\mathbf{W}\|_1.$$

That is, we constrain the HBM to have only a sparse set of all possible high-order interactions. However, calculating the above negative log-likelihood and its gradient is intractable. To address this, we convert the problem of minimizing the negative log-likelihood of observed data into that of minimizing the negative pseudo log-likelihood as proposed in [13]. Specifically, we solve the following optimization function

$$\min_{W} \sum_i \log p(v_i | \mathbf{v}_{-i}, W) + \lambda \|W\|_1,$$

where $\mathbf{v}_{-i}$ is the set of visible units except v_i. Essentially, the above optimization takes the form of a set of ℓ_1-regularized logistic regression problems that are not independent due to the shared parameters W.

Due to the extremely large space of the parameters for the high-order interactions, we approximate the above pseudo log-likelihood further by utilizing a strategy proposed by Wainwright *et al* [26] and propose the following decoupled 2-step method for learning a Sparse High-order Boltzmann Machine, denoted as SHBM.

Step 1: high-order interaction neighborhood estimation: we first estimate the high-order interaction neighborhood structure of each visible unit, i.e., the Markov blanket of each unit. We formulate this problem as a high-order feature selection problem and propose a learning algorithm, denoted as shooter, as described in Section 3.2. In particular, for each visible unit (i.e., each feature), we consider a regression problem from all the other visible units and their high-order interactions.

Step 2: SHBM weight learning: once the high-order interaction neighborhood structure of each visible unit is identified, we add the corresponding energy functions with respect to the high-order interaction of that unit into the energy function of HBM. Then we use Maximum-Likelihood Estimation updates as in Equation 3 to learn the weights associated with the identified high-order energy functions, which requires drawing samples from the model distribution. In Section 4, we present Gibbs Sampling and Mean-Field updates for obtaining samples. Instead of drawing samples exactly from the equilibrium model distribution, we only perform sampling a few steps and use Contrastive Divergence (CD) [11] to update the weights.

3 Sparse High-Order Logistic Regression for SHBM

3.1 Review of ℓ_1-regularized Logistic Regression

Given a dataset of n data points $\{\mathbf{x}_i, y_i\}$, where $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \{+1, -1\}$, and $i = 1, \ldots, n$, ℓ_1-regularized Logistic Regression, denoted as ℓ_1-LR, seeks a classification function $f(\mathbf{w}, b)$ by solving the following ℓ_1-regularized optimization problem:

$$\min_{\mathbf{w},b} f(\mathbf{w}, b) = \min_{\mathbf{w},b} L(\mathbf{w}, b) + \lambda\|\mathbf{w}\|_1,$$

where

$$L(\mathbf{w}, b) = \sum_{i=1}^{n} \log(1 + exp(-y_i(\mathbf{w}^T\mathbf{x}_i + b))),$$

and $\|\mathbf{w}\|_1$ is the ℓ_1-norm of $\mathbf{w}$. The sub-differential of $f(\mathbf{w}, b)$ with respect to w_j is

$$\partial_j f(\mathbf{w}, b) = \nabla_j L(\mathbf{w}, b) + \lambda \text{sign}(w_j), \qquad (4)$$

where $\nabla_j L(\mathbf{w}, b)$ is the standard gradient of the loss function $L(\mathbf{w}, b)$ with respect to w_j. Since the pseudo-gradient [2] of $f(\mathbf{w}, b)$ is the sub-differential of $f(\mathbf{w}, b)$ at $\mathbf{w}$ with minimum norm, and because the sub-differential in Equation 4 is separable in the variables w_j's, the pseudo-gradient of $f(\mathbf{w}, b)$ with respect to each variable w_j can be calculated in a closed form [21].

Among many algorithms solving the above optimization problem (see [21] for a comprehensive review), Projected Scaled Sub-Gradient (PSSG) method is one of the most efficient [21]. In specific, in PSSG, during each iteration, the weight vector $\mathbf{w}$ is split into two sets: a working set that contains all sufficiently non-zero weights and an active set that is the complement of the working set. Then an L-BFGS update is performed on the working set and a diagonally-scaled pseudo-gradient update is performed on the active set so as to get the the descent direction $\mathbf{d}$. Finally, orthant projections are applied on both sets. The orthant projection $\mathcal{P}$ on weight vector $\mathbf{w}$ with the descent direction $\mathbf{d}$ takes the following form:

$$\mathcal{P}(\mathbf{w} + \mathbf{d})_j = \begin{cases} 0 & \text{if } w_j(w_j + d_j) < 0, \\ w_j + d_j & \text{otherwise}, \end{cases} \qquad (5)$$

which ensures that some weights are set to exactly 0 and the weight updates never cross points of non-differentiability.

3.2 Sparse High-Order ℓ_1-Regularized Logistic Regression

We extend the conventional ℓ_1-LR to have both single features and multiplicative feature interactions of orders up to m as predictors with ℓ_1 regularization, and this method is denoted as **s**parse **h**igh-**o**rder l**o**gis**t**ic r**e**g**r**ession (shooter). The optimization problem of shooter with feature interactions of maximum order m is as follows:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \sum_{i=1}^{n} \log\{1+ \\ & \exp[-y_i(\sum_{k=1}^{m} \sum_{j_1<j_2<\cdots<j_k} w_{j_1 j_2 \cdots j_k} x_i^{j_1} x_i^{j_2} \cdots x_i^{j_k} + b)]\} \\ & + \sum_{k=1}^{m} \lambda_k \sum_{j_1<j_2<\cdots<j_k} |w_{j_1 j_2 \cdots j_k}|, \end{aligned} \qquad (6)$$

where x_i^j denotes the j-th feature of $\mathbf{x}_i$. Solving the problem in Equation 6 directly is intractable even for fair feature set size p and small interaction order m (e.g. $p = 500$, $m = 6$). Thus, we propose a greedy block-wise optimization method to solve Equation 6.

We decompose the above problem into several sub-problems and solve the sub-problems greedily from the lowest order 1 up to the maximum order m as follows.

Step 1: first, we denote the set of all the single features as $F_0^{(1)}$, that is,

$$F_0^{(1)} = \{x_{\cdot}^j | \forall j\}$$

We use PSSG to solve the optimization problem as in Equation 7.

$$\begin{aligned} \min_{\mathbf{w}^{(1)},b^{(1)}} \quad & \sum_{i=1}^{n} \log\{1 + \exp[-y_i(\sum_{x_i^j \in F_0^{(1)}} w_j^{(1)} x_i^j + b^{(1)})]\} \\ & + \lambda_1 \sum_{x_i^j \in F_0^{(1)}} |w_j^{(1)}|. \end{aligned} \qquad (7)$$

The discriminative single features are identified as the ones which have non-zero weights $w_j^{(1)}$ across all the data points. We denote this set of identified single features by $F^{(1)}$, that is,

$$F^{(1)} = \{x_{\cdot}^j | x_{\cdot}^j \in F_0^{(1)}, w_j^{(1)} \neq 0\},$$

where $j = 1, ..., p_1$, $p_1 = |F^{(1)}|$.

Step 2: then we multiply each discriminative feature in $F^{(1)}$ with all the rest $p - 1$ single features in $F_0^{(1)}$ to construct the set of all possible second-order feature interactions $F_0^{(2)}$, that is

$$F_0^{(2)} = \{x_{\cdot}^{j_1} x_{\cdot}^{j_2} | x_{\cdot}^{j_1} \in F^{(1)}, x_{\cdot}^{j_2} \in F_0^{(1)}, j_1 \neq j_2\}$$

We solve the optimization problem as in Equation 8

$$\begin{aligned}\min_{\mathbf{w}^{(2)},b^{(2)}} \quad & \sum_{i=1}^{n} \log\{1+ \\ & \exp[-y_i(\sum_{x_i^{j_1} \in F^{(1)}} w_{j_1}^{(2)} x_i^{j_1} \\ & \quad + \sum_{x_i^{j_1} x_i^{j_2} \in F_0^{(2)}} w_{j_1 j_2}^{(2)} x_i^{j_1} x_i^{j_2} + b^{(2)})]\} \\ & + \lambda_1 \sum_{x_i^{j_1} \in F^{(1)}} |w_{j_1}^{(2)}| + \lambda_2 \sum_{x_i^{j_1} x_i^{j_2} \in F_0^{(2)}} |w_{j_1 j_2}^{(2)}|.\end{aligned} \tag{8}$$

so as to identify discriminative second-order feature interaction set $F^{(2)}$, that is,

$$F^{(2)} = \{x_\cdot^{j_1} x_\cdot^{j_2} | x_\cdot^{j_1} x_\cdot^{j_2} \in F_0^{(2)}, w_{j_1 j_2}^{(2)} \neq 0\}.$$

Step 3: similarly, we multiply each discriminative $(k-1)$-th order feature interaction in set $F^{(k-1)}$ with $p-k+1$ other single features in $F_0^{(1)}$ to construct the set of all possible k-th order interactions $F_0^{(k)}$, that is,

$$\begin{aligned}F_0^{(k)} \quad =& \{x_\cdot^{j_1} x_\cdot^{j_2} \cdots x_\cdot^{j_k} | x_\cdot^{j_1} x_\cdot^{j_2} \cdots x_\cdot^{j_{k-1}} \in F^{(k-1)}, \\ & \qquad x_\cdot^{j_k} \in F_0^{(1)}, \\ & \qquad j_k \neq j_{k-q}, \forall q = 1, \cdots, k-1\}\end{aligned}$$

Then from $F_0^{(k)}$ we identify discriminative feature interaction set $F^{(k)}$ by solving the optimization problem as in Equation 9.

$$\begin{aligned}\min_{\mathbf{w}^{(k)},b^{(k)}} \quad & \sum_{i=1}^{n} \log\{1 + \exp[-y_i(\sum_{q=1}^{k-1} \\ & \sum_{x_i^{j_1} x_i^{j_2} \cdots x_i^{j_q} \in F^{(q)}} w_{j_1 j_2 \cdots j_q}^{(k)} x_i^{j_1} x_i^{j_2} \cdots x_i^{j_q} \\ & + \sum_{x_i^{j_1} x_i^{j_2} \cdots x_i^{j_k} \in F_0^{(k)}} w_{j_1 j_2 \cdots j_q}^{(k)} x_i^{j_1} x_i^{j_2} \cdots x_i^{j_k} \\ & + b^{(k)})]\} \\ & + \sum_{q=1}^{k-1} \lambda_q \sum_{x_i^{j_1} x_i^{j_2} \cdots x_i^{j_q} \in F^{(q)}} |w_{j_1 j_2 \cdots j_q}^{(k)}| \\ & + \lambda_k \sum_{x_i^{j_1} x_i^{j_2} \cdots x_i^{j_k} \in F_0^{(k)}} |w_{j_1 j_2 \cdots j_k}^{(k)}|.\end{aligned} \tag{9}$$

and the order-k discriminative feature interaction set $F^{(k)}$ is identified as

$$F^{(k)} = \{x_\cdot^{j_1} x_\cdot^{j_2} \cdots x_\cdot^{j_k} | x_\cdot^{j_1} x_\cdot^{j_2} \cdots x_\cdot^{j_k} \in F_0^{(k)}, w_{j_1 j_2 \cdots j_k}^{(k)} \neq 0\}.$$

Note that in Equation 9 we include discriminative single features and discriminative lower-order interactions $F^{(1)}, \cdots, F^{(k-1)}$ into the ℓ_1-regularized optimization problem for order k so as to optimally remove less important lower-order interactions when high-order interactions present. To speed up the optimization, we divide each identified discriminative feature interaction set F into equal-sized blocks, and we expand each block and solve the ℓ_1-regularized optimization problem for the particular block.

The above greedy optimization approach sequentially identifies discriminative feature interactions of different orders that essentially form a tree structure, because each k-th order discriminative feature interactions must have at least one of its $(k-1)$-th order constituents belonging to $F^{(k-1)}$, where $k > 1$. Although this greedy approach can only identify a sub-optimal solution to the original intractable optimization problem in Equation 6, it performs very well in practice as demonstrated by our experimental results.

4 Sampling Methods for SHBM

In this section, we present Contrastive Divergence (CD) learning [11] based on Gibbs Sampling (GS) and damped Mean-Field updates (MF). The weight updates in SHBM based on CD are as follows,

$$\Delta W_{i_1 i_2 \cdots i_j} = \epsilon(\langle v_{i_1} v_{i_2} \cdots v_{i_j} \rangle_{\text{data}} - \langle v_{i_1} v_{i_2} ... v_{i_j} \rangle_T), \tag{10}$$

where $\langle v_{i_1} v_{i_2} ... v_{i_j} \rangle_T$ is calculated using the samples obtained from different sampling methods after T steps. Although CD updates do not exactly follow the gradient of data log-likelihood, it works well in practice.

Gibbs sampling (GS) can be used within CD for drawing samples. To perform Gibbs Sampling, we initialize $\mathbf{r}^{(0)}$ to be a random data vector, and we sample each visible unit v_j sequentially using the conditional probability

$$p^{(t)}(v_j | r_1^{(t)}, \cdots, r_{j-1}^{(t)}, r_{j+1}^{(t-1)}, \cdots, r_p^{(t-1)})$$

to get the sample for unit v_j in step t, where $j = 1, \ldots, p, t = 1, \ldots, T$, and p is the total number of visible units. Then we use the statistics in the T-step samples to calculate the second term in Equation 10 for weight updates.

However, standard GS cannot be performed in parallel due to the sequential sampling procedure over all the visible units. To speed up learning, we use mean-field approximations (MF) [27] to calculate the sampled values for all the visible units in each step in parallel given the sample values in the previous step (please note that GS and MF have the same computational complexity without parallelization). In specific, we use the damped version of mean-field updates [20] to draw samples to increase sampling stability. Starting from a random data vector $\mathbf{r}^{(0)}$, we calculate the t-step sample for each visible unit v_j as follows,

$$r_j^{(t)} = \lambda r_j^{(t-1)} + (1-\lambda)p(v_j = 1|\mathbf{v}_{-j}, \mathbf{W}),$$

where $t = 1, \dots, T$, and $p(v_i = 1|\mathbf{v}_{-i}, \mathbf{W})$ is the conditional probability of $v_i = 1$ given its neighborhood interactions. Please note that, unlike in GS, we can calculate $\mathbf{r}^{(t)}$ for all the visible units in parallel to speed up our computation because the calculation for $\mathbf{r}^{(t)}$ is only dependent on $\mathbf{r}^{(t-1)}$. In all our experiments, we set $\lambda = 0.2$ for parameter learning based on Damped MF updates.

5 Related Work

For continuous features, undirected ℓ_1-regularized Gaussian graphical model [9] and its extension [18] have been developed for pairwise feature interaction identification by estimating nonzero entries of inverse feature covariance matrix. For discrete binary features, undirected graphical models with only pairwise feature interactions are equivalent to traditional Markov Random Field (MRF) [16] and Boltzmann Machine [12]. Most existing work on undirected graphical models [26, 21] focuses on pairwise interactions and their extensions to high dimensions (see chapter 5 and 6 of [21] for related literature and recent work in [14] for large-scale Gaussian Graphical Models), and there are only a few exceptions on high-order interactions. Dahinden *et al* [6] utilized the log-linear models to learn the structure for discrete data using group ℓ_1-regularization, where all potentials up to a fixed order are considered. However, their methods work on tiny toy datasets and are not scalable to even medium-size problems. Schmidt *et al* [22] addressed the high-order interaction problem among features via convex optimization, which is the state of the art for high-order structure learning, but they have a strong hierarchical assumption on the high-order interactions and compute the partition function in an expensive manner. Ding [7] proposed a method to learn the high-order interactions among data labels, not on data features. Schmidt *et al* [23] also proposed a conditional random field method to learn the high-order interactions among labels.

6 Experiments

We conduct three sets of experiments to demonstrate the performance of shooter and SHBM on interaction neighborhood estimation, feature interaction identification (structure learning) and data reconstruction, respectively. In particular, to test the performance of shooter for interaction neighborhood estimation, we test the classification performance of shooter in the presence of true high-order feature interactions.

6.1 Datasets

We use three datasets, i.e., $\texttt{syn}_{\texttt{small}}$, $\texttt{syn}_{\texttt{large}}$ and mnist for interaction neighborhood identification via classification; and one dataset TF for interaction network learning and data reconstruction.

The dataset $\texttt{syn}_{\texttt{small}}$ is synthetic, in which high-order feature interactions are explicitly designed and encoded. In specific, 10,000 data points of 2,000 binary features were randomly generated. Out of the 2,000 features, 1% of them were randomly selected as informative features at level 1. We randomly pick two informative features from level 1 as seed features of level 2. For each seed feature at level 2, another set of 1% of the entire 2,000 features was randomly selected to be the ones that interact with the seed, and thus to generate the 2nd-order interactions. Such expansion was kept going up to the 3rd-order interactions. Then for each data point, a new feature vector was generated by considering all selected feature interactions as new feature dimensions. The values assigned to these new features were calculated as the product of the values that the data point has for the original component features. In the end, a weighting vector was randomly generated for all the new feature vectors of each data point, and positive (negative) label was assigned to the data point if the sum of the product between the weighting vector and the new feature vector is positive (negative). Finally, we randomly flipped the labels of 0.1% of the generated data points to introduce noise.

The dataset $\texttt{syn}_{\texttt{large}}$ was also synthetic and generated in a same way as to $\texttt{syn}_{\texttt{small}}$, except there are 10,000 data points and 200,000 features. Out of these 20,000 features 0.01% were selected as informative features, and another set of 0.01% of the entire 200,000 features were selected interactively for constructing higher-order interactions. We also introduced noise as we did for generating $\texttt{syn}_{\texttt{small}}$. This dataset is considered as very large and used to test the scalability of shooter.

The dataset mnist is constructed from the MNIST database of handwritten digits[1]. We chose all the images of digit 0 and 6 and gave label +1 to the images of digit 0 and label -1 to the images of digit 6, respectively. We vectorized all the image pixels so as to convert each originally 28 × 28 image to a vector of length 784, with all the pixel values normalized from range [0, 255] to range [0, 1] by dividing the maximal possible pixel value (i.e., 255). We used all such vectors and their labels in the experiments, considering each image as a data point and each pixel as a feature. The dataset mnist has in total 13,779 data points.

[1] http://yann.lecun.com/exdb/mnist/

The dataset TF is downloaded from Gerstein *et al* [10]. Against a set of regulatory targets which have promoter-proximal binding sites, 116 human TFs were tested through ChIP-seq experiments. In TF, each gene is considered as a data point, each TF is considered as a feature dimension, and each data point is represented by the interaction profile (i.e., 1 for interaction and 0 for non-interaction) of the corresponding gene with respect to the TFs. The dataset TF has in total 14153 data points and 116 features with density 30%.

6.2 Classification

We compare the performance of shooter for classification against other three alternatives: Support Vector Machines (SVM) [5] with a linear kernel, logistic regression and ℓ_1-regularized logistic regression [19]. We chose these three methods for comparison because they are either strong classifiers and/or they have feature selection mechanisms. These three comparison methods are denoted as lin-SVM, LR and ℓ_1-LR, respectively. In this set of experiments, classification error is used as the metric to evaluate the performance.

For these datasets, we randomly selected 20% of the entire data as test set. We used 30% of the remaining 80% data as a validation set and the rest 70% as training set. The optimal parameters are first identified by training a model from the training set and testing it on the validation set. With the optimal parameters, another model is trained using both the training data and the validation data and tested on the test set.

Table 1 presents the performance of different methods on syn$_{\text{small}}$, syn$_{\text{large}}$ and mnist. Overall, shooter outperforms lin-SVM, LR and ℓ_1-LR on all the classification tasks. On the synthetic dataset syn$_{\text{small}}$ and syn$_{\text{large}}$, in which the high-order interactions exist by design, shooter performs dramatically better than the others. The error on the test set is 84.4% and 63.2% smaller than the best of the other methods for syn$_{\text{small}}$ and syn$_{\text{large}}$ respectively. In particular, shooter is significantly faster than lin-SVM on syn$_{\text{large}}$, which demonstrates its scalability on large-scale classification tasks. For the cases of dataset mnist, where shooter still outperforms others, the improvement is 18.2% compared with the best of the other three methods. By introducing layers of hidden units on learned interactions, the classification performance can be potentially significantly improved. In addition to superior classification performance, for all the three datasets, shooter successfully identifies order-3 feature interactions (i.e., 3 parameters for shooter in Table 1). The significant improvement of shooter over others demonstrates the effectiveness of shooter in identifying beneficial high-order interactions for neighborhood identification purpose for SHBM.

6.3 Interaction Network Learning

We evaluate the performance of SHBM with different sampling methods against BM and BN on unsupervised feature interaction identification and interaction weight learning on the TF dataset. We first use enrichment to evaluate the performance. Enrichment for TF-TF interactions [4] is calculated as

$$enrichment = \frac{n_{observed}}{n_{expected}},$$

where $n_{observed}$ is the number of true interactions that are correctly predicted by different methods, and $n_{expected}$ is the expected number of physical interactions by chance, which is calculated as

$$n_{expected} = N \times \frac{n_0}{\binom{n_{TF_0}}{2}},$$

where N is the number of putative TF-TF interactions under consideration, n_0 is the total number of true physical interactions among human TFs, and n_{TF_0} is the total number of human TFs. Higher enrichment score corresponds to more accurate interaction predictions.

Table 2 shows the enrichment of SHBM with GS, SHBM with MF, shooter (i.e, SHBM without weight refitting, that is, only use shooter to identify interaction neighborhood; if one interaction appears multiple times, we use the maximum of the weights as the final weight for the interaction, which is proposed in [26]), BM with GS, BM with GS, and BN, for top 50 up to top 250 ranked order-2 interactions based on interaction weights, respectively. SHBM outperforms other methods consistently except that for top 100 ranked interactions it is slightly worse than shooter. In particular, SHBM with GS is notably better than the others for top ranked interactions. BM is competitive, but it is not able to identify any higher-order interactions.

Table 2: Enrichment results

Method	top-50	top-100	top-150	top-200	top-250
SHBM with GS	**33.653**	18.696	**22.435**	**24.305**	**20.192**
SHBM with MF	**33.653**	18.696	19.942	19.631	18.696
shooter	**33.653**	**20.566**	16.203	15.892	17.200
BM with GS	18.696	18.696	16.203	17.761	16.453
BM with MF	18.696	18.696	16.203	18.696	18.696
BN	7.478	5.609	3.739	3.739	2.991

The columns corresponding to top-50, top-100, top-150, etc, represent the enrichment for top 50, top 100, top 150, etc, ranked interactions predicted by different methods. The **bold** numbers correspond to the best performance across the methods.

Figure 1, Figure 2 and Figure 3 show the networks of top 100 interactions identified by SHBM with GS, BM

Table 1: Classification performance

Method	syn$_{\text{small}}$			syn$_{\text{large}}$			mnist		
	param	verr	terr	param	verr	terr	param	verr	terr
shooter	80, 100, 100	0.122	**0.048**	130, 130, 110	0.133	**0.125**	3.5, 0.7, 0.7	0.005	**0.009**
lin-SVM	1	0.408	0.364	10	0.464	0.483	1	0.003	0.014
LR	-	0.405	0.367	-	0.460	0.480	-	0.002	0.014
ℓ_1-LR	80	0.325	0.308	130	0.349	0.332	3.5	0.008	0.011

The column corresponding to "param" represents the values for the parameters in the corresponding methods: for lin-SVM, the parameter is C, the cost factor; for ℓ_1-LR, the parameter is λ, the parameter for the ℓ_1 regularization; and for shooter, it is a list of parameter λ's, corresponding to the regularization parameters for each order of interactions. The column corresponding to "verr" represents the errors on validation set corresponding to the optimal parameters. The column corresponding to "terr" represents the errors on testing set using the optimal parameters. The **bold** numbers correspond to the best performance on the testing set.

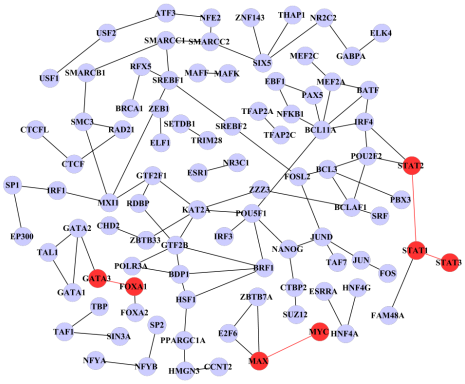

Figure 1: Interactions from SHBM with GS

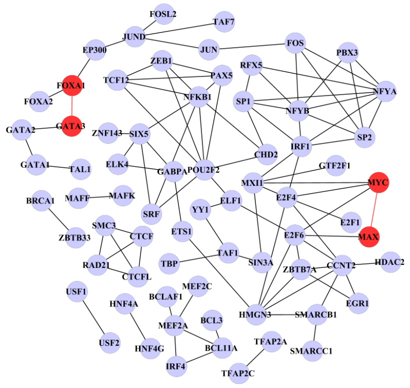

Figure 2: Interactions from BM with GS

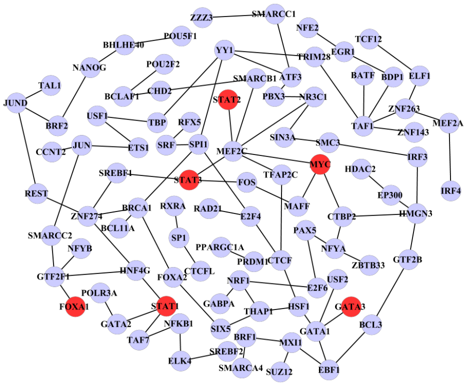

Figure 3: Interactions from BN

with GS and BN on dataset TF, respectively[2]. BN has been the state of the art in bioinformatics for identifying interactions. However, compared to BN and BM, SHBM identifies more biologically meaningful and significant interactions. As an example, four important interactions: MYC vs MAX, STAT1 vs STAT2, STAT1 vs STAT3 and FOXA1 vs GATA3, are successfully identified (highlighted in Figure 1 by SHBM. BM only identifies MYC vs MAX and FOXA1 vs GATA3 but misses the others. BN misses all the four important interactions. MYC and MAX form a MAX/MYC heterodimer, which has been discovered and studied recently in literature [3]. The interaction between MYC and MAX is ranked 11 by SHBM among all the identified interactions. STAT1 and STAT2 also form an heterodimer and the interaction has been studied in literature [15] and this interaction is ranked 8 by SHBM. The interaction between STAT1 and STAT3 was studied in [28]. The interaction between FOXA1 and GATA3 is highly ranked by SHBM, which also has literature support from several recent studies [25] SHBM also identifies some famous high-order interactions such as USF1-USF2-NFE2 [29], and FOS-NFYA-STAT3 [8], etc.

6.4 Data Reconstruction

We compare SHBM and BM with GS and MF, respectively, on how well they can fit the data and accordingly generate new interactions that are true with high probabilities. This set of experiments is conducted on TF dataset, which has no labels but its interaction network has important biological significance. 80% of the entire TF dataset is used for SHBM and BM training, whereas the rest 20% is held out for testing.

Table 4 presents the pseudo negative log-likelihood (pNLL) of different methods on the test data. SHBM with GS outperforms others in term of pNLL, demonstrating that SHBM with GS better fits the data. SHBM with GS has an improvement 13.9% over BM with MF,

[2] The networks identified by SHBM with MF and BM with MF are presented in the supplementary materials

Table 3: Reconstruction performance

Method	5	10	15	20	25	30
SHBM with GS	**1385.0**	**2757.6**	**4257.6**	**5561.5**	**7037.6**	**8228.7**
SHBM with MF	1618.8	3284.5	5044.2	6588.1	8413.3	9928.3
BM with GS	1600.6	3217.6	4930.7	6566.2	8139.6	9467.0
BM with MF	1574.0	3155.1	4839.2	6440.6	7997.8	9302.1

The columns corresponding to 5, 10, 15, etc, represent the reconstruction errors when the corresponding number of features are marked off and to be constructed. The **bold** numbers correspond to the best performance across the methods.

Table 4: Model fitting performance

	SHBM with GS	SHBM with MF	BM with GS	BM with MF
pNLL	**0.9829e+05**	1.1846e+05	1.1607e+05	1.1421e+05

The row corresponding to "pNLL" represents the pseudo negative log-likelihood on the testing set. The **bold** numbers correspond to the best performance across the methods.

which is the second best performing method in term of pNLL.

For the data reconstruction, first a random set of features is selected and masked off from the entire data, that is, all the corresponding binding between TFs and proteins are reset as none, and it is to utilize the information of the rest features in the data and the interaction relations among features to recover the masked-off part of the original interactions. 5, 10, 15, 20, 25 and 30 features out of 116 are randomly selected and masked off, and then reconstructed by SHBM and BM. Such procedure is repeated 20 times and the average squared errors over the 20 times from SHBM and BM are presented in Table 3. Consistently, SHBM with GS outperforms others on all the cases, and improvement from the second best method, which is BM with MF, is 12.0%, 12.6%, 12.0%, 13.6%, 12.0%, 11.5%, respectively. This demonstrates that SHBM is able to better fit the data and accordingly generate most possible interactions.

6.5 Comparison with Other Methods

We compare SHBM with the hierarchical log-linear model proposed by Schmidt *et al* [22], denoted by HLLM. HLLM is not scalable to even medium-size datasets and we couldn't get results from HLLM on TF within three days. Due to this, we only compared SHBM and HLLM on a small synthetic dataset generated from the source code provided with HLLM, but with only 1000 data points and 10 features. The performance is presented in Table 5.

Table 5 shows that for the small dataset, SHBM and HLLM are very comparable. However, SHBM is scalable to very large datasets, while HLLM significantly suffers from scalability issues. This makes SHBM particularly useful in practice for real large-scale problems.

Table 5: Comparison results

	SHBM with GS	HLLM
pNLL	2329.39	2334.81

7 Conclusion

In this paper, we present SHBM, an interpretable sparse high-order Boltzmann machine, and propose a two-step learning algorithm for SHBM. In the first step of SHBM learning, we propose shooter, a greedy sparse learning approach via ℓ_1-regularized logistic regression to identify high-order feature interactions so as to identify interaction neighborhood structure for SHBM. In the second step of SHBM, different sampling methods are proposed to learn the interaction weights in SHBM. experimental results demonstrate that shooter outperforms other methods in identifying interaction neighborhood by exploring high-order interactions during classification. In addition, weight learning in SHBM produces better rankings among interactions and better generative models than other competing models. In particular, SHBM successfully identifies biologically meaningful and significant interactions from a real biological dataset, whereas other state-of-the-art methods miss such interactions. SHBM is also demonstrated to be scalable to very large problems, while the state-of-the-art method for high-order interactions fail.

In the future, we will incorporate abundant group information of features to enhance the power of shooter, when limited data points are available. Moreover, we will add hidden units and gated hidden units to increase the generative power of SHBM for unsupervised feature interaction identification and for collaborative filtering applications.

References

[1] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

[2] G. Andrew and J. Gao. Scalable training of l 1-regularized log-linear models. In *Proceedings of the 24th international conference on Machine learning*, pages 33–40. ACM, 2007.

[3] A. Cascón and M. Robledo. Max and myc: a heritable breakup. *Cancer research*, 72(13):3119–3124, 2012.

[4] C. Cheng, R. Min, and M. Gerstein. A probabilistic method for identifying transcription factor target genes from chip-seq binding profiles. *Bioinformatics*, 2011.

[5] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[6] C. Dahinden, G. Parmigiani, M. Emerick, and P. Bühlmann. Penalized likelihood for sparse contingency tables with an application to full-length cdna libraries. *BMC bioinformatics*, 8(1):476, 2007.

[7] S. Ding, G. Wahba, and X. J. Zhu. Learning higher-order graph structure with features by structure penalty. In *NIPS*, pages 253–261, 2011.

[8] J. D. Fleming, G. Pavesi, P. Benatti, C. Imbriano, R. Mantovani, and K. Struhl. Nf-y coassociates with fos at promoters, enhancers, repetitive elements, and inactive chromatin regions, and is stereo-positioned with growth-controlling transcription factors. *Genome Research*, 2013.

[9] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[10] M. B. Gerstein and *et al.* Architecture of the human regulatory network derived from ENCODE data. *Nature*, 489(7414):91–100, Sept. 2012.

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

[12] G. E. Hinton and T. J. Sejnowski. Learning and relearning in boltzmann machines. *MIT Press, Cambridge, Mass*, 1:282–317, 1986.

[13] H. Höfling and R. Tibshirani. Estimation of sparse binary pairwise markov networks using pseudo-likelihoods. *The Journal of Machine Learning Research*, 10:883–906, 2009.

[14] C.-J. Hsieh, M. A. Sustik, I. Dhillon, P. Ravikumar, and R. Poldrack. Big & quic: Sparse inverse covariance estimation for a million variables. *NIPS*, 2013.

[15] M. G. Katze, Y. He, and M. Gale. Viruses and interferon: a fight for supremacy. *Nature Reviews Immunology*, 2(9):675–687, 2002.

[16] R. Kindermann, J. L. Snell, et al. *Markov random fields and their applications*, volume 1. American Mathematical Society Providence, RI, 1980.

[17] S. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of markov networks using l1 regularization. In *In NIPS*. Citeseer, 2006.

[18] H. Liu, J. Lafferty, and L. Wasserman. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *The Journal of Machine Learning Research*, 10:2295–2328, 2009.

[19] A. Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pages 78–, New York, NY, USA, 2004. ACM.

[20] S. Osindero and G. E. Hinton. Modeling image patches with a directed hierarchy of markov random fields. *Advances in neural information processing systems*, 20:1121–1128, 2008.

[21] M. Schmidt. *Graphical model structure learning with l1-regularization.* PhD thesis, UNIVERSITY OF BRITISH COLUMBIA, 2010.

[22] M. Schmidt and K. Murphy. Convex structure learning in log-linear models: Beyond pairwise potentials. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[23] M. Schmidt, K. Murphy, G. Fung, and R. Rosales. Structure learning in random fields for heart motion abnormality detection. *CVPR. IEEE Computer Society*, 2008.

[24] T. J. Sejnowski. Higher-order boltzmann machines. In *Neural Networks for Computing*, pages 398–403. American Institute of Physics, 1986.

[25] V. Theodorou, R. Stark, S. Menon, and J. S. Carroll. Gata3 acts upstream of foxa1 in mediating esr1 binding by shaping enhancer accessibility. *Genome research*, 23(1):12–22, 2013.

[26] M. Wainwright, P. Ravikumar, and J. Lafferty. High-dimensional graphical model selection using l1̃-regularized logistic regression. *Advances in neural information processing systems*, 19:1465, 2007.

[27] M. Welling and G. E. Hinton. A new learning algorithm for mean field boltzmann machines. In *Artificial Neural NetworksICANN 2002*, pages 351–357. Springer, 2002.

[28] L. Xia, L. Wang, A. S. Chung, S. S. Ivanov, M. Y. Ling, A. M. Dragoi, A. Platt, T. M. Gilmer, X.-Y. Fu, and Y. E. Chin. Identification of both positive and negative domains within the epidermal growth factor receptor cooh-terminal region for signal transducer and activator of transcription (stat) activation. *Journal of Biological Chemistry*, 277(34):30716–30723, 2002.

[29] Z. Zhou, X. Li, C. Deng, P. A. Ney, S. Huang, and J. Bungert. Usf and nf-e2 cooperate to regulate the recruitment and activity of rna polymerase ii in the -globin gene locus. *Journal of Biological Chemistry*, 285(21):15894–15905, 2010.