

Empirical Analysis of Sampling Based Estimators for Evaluating RBMs

Vidyadhar Upadhyaya and P. S. Sastry

Indian Institute of Science, Bangalore, India

Abstract. The Restricted Boltzmann Machines (RBM) can be used either as classifiers or as generative models. The quality of the generative RBM is measured through the average log-likelihood on test data. Due to the high computational complexity of evaluating the partition function, exact calculation of test log-likelihood is very difficult. In recent years some estimation methods are suggested for approximate computation of test log-likelihood. In this paper we present an empirical comparison of the main estimation methods, namely, the AIS algorithm for estimating the partition function, the CSL method for directly estimating the log-likelihood, and the RAISE algorithm that combines these two ideas. We use the MNIST data set to learn the RBM and then compare these methods for estimating the test log-likelihood.

1 Introduction

The Restricted Boltzmann Machines (RBM) are among the basic building blocks of deep learning models. They can be used as classifiers as well as generative models. The parameters of an RBM are learnt with an objective of maximizing the log-likelihood using, e.g., the contrastive divergence [5]. The quality of the learnt generative RBM is evaluated based on the average log-likelihood which, for N test samples, is given by,

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N \log p(\mathbf{v}^{(i)}) \quad (1)$$

where $p(\mathbf{v}^{(i)})$ is the probability (or the likelihood) of the i^{th} test sample, $\mathbf{v}^{(i)}$. The model with higher average test log-likelihood is better. The test log-likelihood can also be used in devising a stopping criterion for the learning and for fixing the hyper-parameters through cross validation.

The likelihood $p(\mathbf{v})$ can be written as $p^*(\mathbf{v})/Z$. While $p^*(\mathbf{v})$ is easy to evaluate, the normalizing constant Z , called the partition function, is computationally expensive. Therefore, various sampling based estimators have been proposed for the estimation of the log-likelihood. In this paper we present an empirical analysis of the two main approaches for estimating the log-likelihood.

The first approach of estimating the average test log-likelihood is to approximately estimate the partition function of the model distribution and then use it to calculate the log-likelihood. The partition function can be estimated with the Monte Carlo method using the samples obtained from the model distribution. However, generating independent

samples is difficult since the model distribution which the RBM represents is complicated and high dimensional in most applications. A useful sampling technique in such case is the importance sampling where samples obtained from a simple distribution, called the proposal distribution, are used to estimate the partition function. However, for large models, the variance of this estimator is very high and may even be infinite [6] when the proposal distribution is not a good approximation of the target distribution. To overcome this difficulty of choosing a good proposal distribution, a sequence of intermediate distributions is used. The annealed importance sampling (AIS) based estimator is one such estimator [7].

The second approach is to estimate the average test log-likelihood directly by marginalizing over the hidden variables from the model distribution, because the conditional distribution of the visible units given the hidden units is simple to evaluate in an RBM. However, the computational complexity grows exponentially with the number of hidden units present in the model. Also, this computation has to be repeated for each test sample. Hence, an approximate method which uses a sample based estimator called conservative sampling based likelihood estimator (CSL) is proposed in [1]. A more efficient method called reverse annealed importance sampling estimator (RAISE) implements CSL by formulating the problem of marginalization as a partition function estimation problem.

In this paper, we present an empirical comparison of the performance of the sampling based estimators, namely, annealed importance sampling estimator(AIS) [7], conservative sampling based likelihood estimator (CSL) [1], reverse annealed importance sampling estimator (RAISE)[2]. Since our main aim is to evaluate the learnt RBM, the issues associated with the learning are not addressed in this paper. Initially we learn RBM models (with different number of hidden units) using the standard CD algorithm on the MNIST dataset with suitably chosen hyperparameters. We evaluate the average test log-likelihood of each of the learnt models using the estimators mentioned above and compare their performance. For models with small number of hidden units we can calculate the test log-likelihood through the brute force approach. When the ground truth is known we can readily evaluate the performance of different estimators. However, for large models we do not know the ground truth.

The rest of the paper is organized as follows. In section 2 we first briefly describe the RBM model. We then explain the problem of computing the average test log likelihood for an RBM and explain the two approaches used for solving it. We then describe the different sampling methods considered in this study in section 3. In section 4 we describe the simulation setting, the parameters used in different estimators and then present results of our empirical study. Finally we conclude the paper in section 5.

2 Restricted Boltzmann Machines

The Restricted Boltzmann Machine (RBM) is a special case of the Boltzmann Machine where the intra-layer connections are restricted [9,4,5]. The visible stochastic units (\mathbf{v}) are connected to the hidden stochastic units (\mathbf{h}) through bidirectional links. We restrict our study to the binary RBM with $\mathbf{v} \in \{0, 1\}^m$ and $\mathbf{h} \in \{0, 1\}^n$, being the states of visible and hidden units respectively. Note that, m and n are the number of visible

and hidden units. For the binary RBM, the energy function and the probability that the model assigns to $\{\mathbf{v}, \mathbf{h}\}$ is given by,

$$E(\mathbf{v}, \mathbf{h}; \theta) = - \sum_{i,j} w_{ij} h_i v_j - \sum_{j=1}^m b_j v_j - \sum_{i=1}^n c_i h_i \quad (2)$$

$$p(\mathbf{v}, \mathbf{h}|\theta) = \frac{e^{-E(\mathbf{v}, \mathbf{h}; \theta)}}{Z} \quad (3)$$

where $Z = \sum_{\mathbf{v}, \mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h}; \theta)}$ is the partition function and $\theta = \{\mathbf{w} \in \mathbb{R}^{n \times m}, \mathbf{b} \in \mathbb{R}^m, \mathbf{c} \in \mathbb{R}^n\}$ is the set of model parameters.

2.1 Evaluation of the Average Test Log-likelihood for an RBM

As mentioned in section 1 the parameters of an RBM are learnt using the contrastive divergence algorithm. Once the model is learnt, for a particular test data \mathbf{v} the log-likelihood can be calculated as (see [3] for details),

$$\begin{aligned} \log p(\mathbf{v}) &= \log \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h}; \theta)} - \log Z \\ &= \sum_j b_j v_j + \log \sum_{\mathbf{h}} e^{\sum_i c_i h_i + \sum_j w_{ij} h_i v_j} - \log Z \\ &= \sum_j b_j v_j + \sum_i \log(1 + e^{c_i + \sum_j w_{ij} v_j}) - \log Z \end{aligned} \quad (4)$$

The difficulty in evaluating the above equation is due to the presence of intractable log partition function. The log partition function can be expanded as,

$$\begin{aligned} \log Z &= \log \sum_{\mathbf{v}, \mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h}; \theta)} \\ &= \log \sum_{\mathbf{v}} e^{\sum_j b_j v_j} \sum_{\mathbf{h}} e^{\sum_i c_i h_i + \sum_j w_{ij} h_i v_j} \\ &= \log \sum_{\mathbf{v}} e^{\sum_j b_j v_j} \prod_i \left(1 + e^{c_i + \sum_j w_{ij} v_j}\right) \end{aligned} \quad (5)$$

Similarly, it can be written in terms of \mathbf{h} as,

$$\log Z = \log \sum_{\mathbf{h}} e^{\sum_i c_i h_i} \prod_j \left(1 + e^{b_j + \sum_i w_{ij} h_i}\right) \quad (6)$$

The above equation can also be written in terms of \mathbf{h} . Hence, to evaluate the partition function by using either eq.(5) or eq.(6) we have to sum 2^L terms, where $L = \min\{m, n\}$, which is computationally expensive for large models. However, if we can estimate the partition function by some other method then we can efficiently estimate the log-likelihood using eq.(4).

The test log-likelihood can also be estimated directly by marginalizing the model distribution over the latent variable \mathbf{h} through a sample average as,

$$\log p(\mathbf{v}) = \log \sum_{\mathbf{h}} p(\mathbf{h})p(\mathbf{v}|\mathbf{h}) \approx \log \frac{1}{N} \sum_{i=1}^N p(\mathbf{v}|\mathbf{h}^{(i)}) \quad (7)$$

where N samples of \mathbf{h} are obtained as $\mathbf{h}^{(i)} \sim p(\mathbf{h}), \forall i$.

3 Sampling Based Estimators for Evaluating RBMs

In this section we briefly discuss various sampling based estimators for the test log-likelihood. As mentioned earlier, we either need a method to estimate Z , so that we can use eq.(4); or we need a method to sample from $p(\mathbf{h})$ so that we can use eq.(7) for evaluating the test log-likelihood. We discuss

3.1 Annealed Importance Sampling for Estimating Partition Function

Suppose two distributions $f_A(\mathbf{x}) = f_A^*(\mathbf{x})/Z_A$ and $f_B(\mathbf{x}) = f_B^*(\mathbf{x})/Z_B$ are given such that $f_A(\mathbf{x}) \neq 0$ when $f_B(\mathbf{x}) \neq 0$ and it is possible to obtain independent samples from $f_A(\mathbf{x})$. It is also assumed that $f_A^*(\mathbf{x}), f_B^*(\mathbf{x})$ and Z_A are easy to evaluate. Then the ratio of partition functions can be written as,

$$\frac{Z_B}{Z_A} = \int \frac{f_B^*(\mathbf{x})}{Z_A} d\mathbf{x} = \int \frac{f_B^*(\mathbf{x})}{f_A^*(\mathbf{x})} f_A(\mathbf{x}) d\mathbf{x} \approx \frac{1}{M} \sum_{i=1}^M \frac{f_B^*(\mathbf{x}^{(i)})}{f_A^*(\mathbf{x}^{(i)})} \approx \frac{1}{M} \sum_{i=1}^M w_{\text{imp}}^{(i)} \quad (8)$$

where $w_{\text{imp}}^{(i)} = f_B^*(\mathbf{x}^{(i)})/f_A^*(\mathbf{x}^{(i)})$ is termed as the i^{th} importance weight and $\mathbf{x}^{(i)}$ are independent samples from the distribution f_A . However, the variance of this estimator is very high unless the proposal distribution, f_A , is a good approximation of the target distribution, f_B [6]. Hence, unless we can find a proposal distribution, f_A , that is easy to sample from, and at the same time, it is close to the target distribution, f_B , the importance sampling method does not give a good estimate. In order to overcome this issue with choosing a good proposal distribution, a sequence of intermediate probability distributions is introduced to assist in moving gradually from the proposal to the target distribution [7]. This method is called annealed importance sampling and it is the state of the art estimator for the partition function.

Suppose there are $K - 1$ intermediate distributions, $f_A = f_0, f_1, \dots, f_{K-1}, f_K = f_B$. As earlier, let each distribution be given by $f_i(\mathbf{x}) = f_i^*(\mathbf{x})/Z_i, i = 0, 1, \dots, K$. Then using eq.(8), we have, for every $k, 0 \leq k \leq K - 1$,

$$\frac{Z_{k+1}}{Z_k} \approx \frac{1}{M} \sum_{i=1}^M \frac{f_{k+1}^*(\mathbf{x}_k^{(i)})}{f_k^*(\mathbf{x}_k^{(i)})}, \text{ where } \mathbf{x}_k^{(i)} \sim f_k \quad (9)$$

If we can ensure that the f_k and f_{k+1} are close to each other then the above is a good estimate of the ratio of the partition functions. Further, we can write,

$$\frac{Z_K}{Z_0} = \frac{Z_1}{Z_0} \frac{Z_2}{Z_1} \dots \frac{Z_K}{Z_{K-1}} = \prod_{k=0}^{K-1} \frac{Z_{k+1}}{Z_k} \quad (10)$$

Since $f_0 = f_A$ is a simple distribution, we can calculate Z_0 and hence the above gives us a good estimate for Z_K .

The next question is, how does one sample from each of the f_k , $0 \leq k \leq K - 1$? A standard method for this is as follows. Initially a sample \mathbf{x}_1 is drawn from $f_0(\mathbf{x}) = f_A(\mathbf{x})$. Then for $1 \leq k \leq K - 1$, the $(k + 1)$ th sample, \mathbf{x}_{k+1} , is obtained by sampling from $T_k(\mathbf{x}_{k+1}|\mathbf{x}_k)$ where T_k is a transition function of a (reversible) Markov chain for which f_k is the invariant distribution. This sequence of states, $\mathbf{x}_1, \mathbf{x}_1, \dots, \mathbf{x}_K$ is one sample to estimate Z_k using eq.(9) and eq.(10). Then we can get M samples by starting in an initial state sampled from f_A and then going through the K Markov transitions, M times.

The key idea in this method is that by moving gradually from the proposal to the target distribution, the variance of importance weights can be kept small if the two consecutive intermediate distributions differ by a small amount. For the detailed analysis and conditions for this estimator to work, refer to [7].

The choice of intermediate probability distributions is problem specific. The geometric average of $f_A(\mathbf{x})$ and $f_B(\mathbf{x})$ is a popular choice in literature. Thus we keep,

$$f_k(\mathbf{x}) \propto f_A(\mathbf{x})^{1-\beta_k} f_B(\mathbf{x})^{\beta_k} \quad (11)$$

where β_k 's are chosen such that $0 = \beta_0 < \dots < \beta_K = 1$.

The following two facts give some insight about the partition function estimate obtained using the AIS estimator [2]. The AIS estimator yields unbiased estimate of Z . However, the Jensen's inequality shows that, on the average, AIS estimator underestimates the log partition function,

$$\mathbb{E} \left[\log \hat{Z} \right] \leq \log \mathbb{E} \left[\hat{Z} \right] = \log Z \quad (12)$$

The underestimate of the log partition function results in overestimate for the test log-likelihood (see eq.(4)). Then it is difficult to infer whether the model is good or we got an optimistic likelihood estimate. However, the Markov inequality shows that overestimating the log partition function by a large amount is not very likely,

$$P \left[\log \hat{Z} > \log Z + k \right] = P \left[\hat{Z} > e^{\log Z + k} \right] < \frac{\mathbb{E} \left[\hat{Z} \right]}{Z e^k} = e^{-k} \quad (13)$$

Estimating the Partition Function of an RBM: The partition function of an RBM can be evaluated using the AIS estimator [8]. Consider the two RBMs with parameters $\theta^A = \{\mathbf{w}^A, \mathbf{b}^A, \mathbf{c}^A\}$ and $\theta^B = \{\mathbf{w}^B, \mathbf{b}^B, \mathbf{c}^B\}$ corresponding to the proposal distribution (f_A) and the target distribution (f_B) respectively. The proposal RBM distribution is chosen such that obtaining independent samples from it, is simple. The objective is to estimate the partition function of the RBM given by the parameter θ^B , which is nothing but the set of model parameters of the learnt RBM.

The two choices for the proposal RBM used in the present study are the *uniform* and the *base rate* RBM. The uniform distribution as a proposal can be implemented by taking $\mathbf{w}^A = 0, \mathbf{b}^A = 0, \mathbf{c}^A = 0$. If we consider all zero weight matrix and nonzero

biases the proposal distribution is given as,

$$f_A(\mathbf{v}) = \frac{e^{\sum_j b_j v_j}}{\prod_j (1 + e^{-b_j})} \quad (14)$$

The ML solution for b_j (solving $\frac{\partial f_A}{\partial b_j} = 0$) turns out to be $\log(\bar{v}) - \log(1 - \bar{v})$ where \bar{v} is the mean of the training samples. This is termed as the ‘base rate’ RBM.

The intermediate distributions are constructed using the geometric average path. It corresponds to averaging the energy function $E(\mathbf{v}, \mathbf{h}^A; \theta^A)$ of the proposal RBM and $E(\mathbf{v}, \mathbf{h}^B; \theta^B)$ of the target RBM, i.e.,

$$E_k(\mathbf{v}, \mathbf{h}) = -((1 - \beta_k) E(\mathbf{v}, \mathbf{h}^A; \theta^A) + \beta_k E(\mathbf{v}, \mathbf{h}^B; \theta^B)) \quad (15)$$

where $E_k(\mathbf{v}, \mathbf{h})$ denotes the energy function for the intermediate distribution f_k and \mathbf{h} denotes the stacked hidden units $\{\mathbf{h}^A, \mathbf{h}^B\}$. The k^{th} intermediate distribution is given by,

$$f_k(\mathbf{v}) = \frac{f_k^*(\mathbf{v})}{Z_k} = \frac{1}{Z_k} \sum_{\mathbf{h}} e^{-E_k(\mathbf{v}, \mathbf{h})} \quad (16)$$

Since each of the intermediate distributions f_k is defined by the RBM, the samples are obtained from the invariant distribution through Gibbs sampler. More details on the Gibbs sampler implementation and the exact form of the intermediate distributions are available in [8].

The algorithm works as follows. Initially, a sample \mathbf{v}_1 is generated from the proposal RBM. Then, for $1 \leq k \leq K - 1$, the $(k + 1)^{\text{th}}$ sample $\mathbf{v}^{(k+1)}$ is to be generated by f_k (given in eq. (16)). This is done by sampling from $T_k(\mathbf{v}^{(k+1)} | \mathbf{v}^{(k)})$ where T_k refers to the transition operator for which f_k is the invariant distribution. Since f_k is naturally defined for an RBM, the $T_k(\mathbf{v}^{(k+1)} | \mathbf{v}^{(k)})$ corresponds to Gibbs sampler for the RBM whose energy function is E_k (given in eq. (15)). Once the samples are obtained, the ratio in eq.(10) is used to calculate the partition function.

3.2 Conservative Sampling Based Log-likelihood Estimator

Given the conditional probability $p(\mathbf{v}|\mathbf{h})$ and a set of samples $\{\mathbf{h}^{(1)}, \mathbf{h}^{(2)} \dots \mathbf{h}^{(N)}\}$ from the distribution $p(\mathbf{h})$, the log-likelihood on a test sample using the CSL estimator is given in eq.(7).

The hidden units samples are obtained using the Gibbs sampler for the RBM. The state of the visible units can be initialized randomly (called the unbiased CSL) or by using the training samples (called the biased CSL). Then, the hidden units and the visible units are sampled alternately from the corresponding conditional distributions for a large number of iterations.

The Jensen’s inequality shows that the CSL estimator underestimates the test log-likelihood [1], i.e.,

$$\mathbb{E}_{\mathbf{h}}[\log \hat{p}(\mathbf{v})] \leq \log \mathbb{E}_{\mathbf{h}}[p(\mathbf{v}|\mathbf{h})] = \log p(\mathbf{v}) \quad (17)$$

3.3 Reverse AIS Estimator

The RAISE estimates the log-likelihood similar to the CSL approach while implementing it through the AIS approach [2]. Consider the likelihood on a given test data \mathbf{v} as in eq.(7), $p(\mathbf{v}) = \sum_{\mathbf{h}} p(\mathbf{h})p(\mathbf{v}|\mathbf{h})$. Finding this $p(\mathbf{v})$ is equivalent to finding the partition function of a distribution given by $f(\mathbf{h}) = p(\mathbf{h})p(\mathbf{v}|\mathbf{h})$ for a fixed \mathbf{v} . Hence, the AIS estimator (with suitable modifications) can be used. For the analysis and the implementation details of the method refer [2]. The brief description of the algorithm is given below.

The test data \mathbf{v} is assumed to be a sample from the target distribution. This test sample is used as the initial state of the AIS Markov chain which is defined to move gradually from the target to the proposal distribution as opposed to moving from the proposal to the target distribution as in the AIS estimator discussed in section 3.1. The visible and hidden units, \mathbf{h} and \mathbf{v} , of intermediate distributions are sampled alternately from this chain using the transition operators similar to the ones defined for the AIS estimator. Once the samples are obtained the estimate of the likelihood is given as [2],

$$p(\hat{\mathbf{v}}) = \frac{f_K(\mathbf{v})}{Z_0} \prod_{k=1}^K \frac{f_{k-1}(\mathbf{x}_k)}{f_k(\mathbf{x}_k)} \quad (18)$$

4 Experimental Results and Discussions

In this section we present the results of estimating the partition function and the average test log-likelihood using the different methods discussed in the previous section. We first learn the RBMs with different number of hidden units (20, 200 and 500, denoted as RBM20, RBM200, RBM500) using the standard CD-20 algorithm on the MNIST dataset. We fix the learning rate, $\eta = 0.1$, batch size = 100, weight decay = 0.001, initial momentum = 0.5, final momentum = 0.9 and change the momentum at 5th step. Initial weights are sampled from $U[-1/\sqrt{L}, 1/\sqrt{L}]$ where $L = \min\{m, n\}$.

Then, for each of the learnt RBMs, we estimate the test log-likelihood on the MNIST test dataset using the different estimators. Note that, even though learning the RBM with different hyperparameter settings produce (significantly) different models, we have observed that the behaviour of estimators are similar on these models. Therefore we present results for one such RBM which is learnt using the above hyperparameter settings. For the case of RBM20, the ground truth is calculated by summing over all 2^{20} states so that we can assess accuracy of the estimates.

4.1 AIS

The performance of the AIS estimator depends on the distribution of the proposal RBM, the number of intermediate distributions (K) and the number of samples (M , also called AIS runs). We fix $M = 500$ and use uniform or baserate as the proposal distribution. The value of K is chosen from $\{100, 1000, 10000\}$ and we use the linear path $\beta = [0 : 1/K : 1]$ for all the experiments. The handcrafted schedule having $K = 14500$ given in [8] with $\beta = [0 : 1/1000 : 0.5, 0.5 : 1/10000 : 0.9, 0.9 : 1/100000 : 1]$ is

also implemented for comparison. In order to estimate the variance of the estimator we repeat the experiment 50 times with a random initial state each time. Table 1 gives the estimate of $\log Z$, \mathcal{L} and σ (the standard deviation of the estimate of $\log Z$).

Table 1. AIS estimate of \mathcal{L} , **Ground truth:** $\log Z = 230.61$, $\mathcal{L} = -141.24$ for **RBM20**

n	K	<i>Uniform Proposal</i>			<i>Base rate Proposal</i>		
		$\widehat{\log Z}$	$\hat{\mathcal{L}}$	σ	$\widehat{\log Z}$	$\hat{\mathcal{L}}$	σ
20	1000	229.2660	-139.8933	0.2905	230.5353	-141.1626	0.2114
	10000	229.5245	-140.1518	0.2423	230.6192	-141.2465	0.0629
	14500	229.8978	-140.5250	0.6095	230.6082	-141.2355	0.0431
200	1000	174.1720	-111.2346	0.6724	173.5514	-110.6140	1.2344
	10000	174.7664	-111.8290	0.1528	174.6819	-111.7445	0.2408
	14500	174.7424	-111.8050	0.2623	174.7512	-111.8138	0.1300
500	1000	173.4807	-114.8451	0.3071	172.2157	-113.5801	1.1597
	10000	173.4802	-114.8446	0.1207	173.4767	-114.8411	0.2244
	14500	173.4567	-114.8211	0.1902	173.4240	-114.7883	0.1717

Based on the ground truth available for RBM20, we observe that the AIS estimator, on the average, overestimates the test log-likelihood. The use of base rate proposal distribution gives slightly better estimate with less variance compared to using the uniform proposal distribution for the RBM20. However, for the RBM200 and RBM500 the proposal distribution has no significant effect on the estimated value though it affects the variance of the estimate. On the whole, the linear annealing schedule with $K = 10,000$ seems to perform as well as the hand-crafted annealing schedule with $K = 14,500$.

4.2 CSL

The samples required for the CSL estimator are obtained through the Gibbs sampler which alternately samples the hidden and the visible units. We ignore the first B samples to allow burn-in and then collect samples after every T (called 'Thin' parameter) steps, discarding the samples in between, to avoid correlation. We observed that the estimates obtained with a single chain is poor even if we run the Gibbs sampler for a large number of steps. This possibly indicates the poor mixing rate of the Gibbs chain. Therefore we experiment with many parallel Gibbs chains with different initial states.

We first simulate S_M parallel Gibbs chains with S_T steps for both the biased and the unbiased CSL. Under various values of the parameters M and T , the samples of \mathbf{h} (to estimate the log-likelihood) are selected from these simulated chains. The experiment is repeated 100 times to find the variance of the estimate, where for each experiment we select a random burn-in value and select chains randomly. We consider $S_M = 5000$ and

$S_T = 25000$ for RBM20 and $S_M = 2500$ and $S_T = 50000$ for the other two RBMs. We consider both small and large burn-in setup for the biased CSL estimator.

We fix the value of T to 100 and vary the number of chains, M , keeping the number of samples, N , constant and these results are presented in Table 2. We observe from the table that better estimates with less variance can be obtained by increasing the number of chains. Note that we are keeping the total number of samples fixed even when we vary the number of chains. Thus, for example, for RBM500, we get lower variance by having a total of 100,000 samples from 1000 chains rather than having 200,000 samples but from only 500 chains. Thus having more chains also reduces the total computational effort (because we can do with less number of samples).

The final estimates are close to ground truth in case of RBM20, even though the accuracy here is a bit poorer than that of AIS. For RBM200 and RBM500, the estimates differ by a large amount from those obtained with the AIS, even when N is very large. This deviation is smaller for the biased CSL estimates obtained with small burn-in than that of the unbiased CSL estimates and the biased CSL estimates obtained with large burn-in (refer Table. 2). This may be due to high level of correlation among the samples. Further, we observe that, for the RBM20, the biased CSL estimate gives a lower bound.

We also experimented with varying T keeping M and N constant (not presented here). We found that the improvement in the accuracy of estimates obtained with higher values of T is not very significant though it uses higher number of Gibbs steps. when N is large and a very small value of burn-in is used.

4.3 RAISE

The RAISE estimator requires implementation of the AIS chain for each test sample. This makes the estimator computationally expensive because the MNIST dataset contains 10,000 test samples. Therefore, for the experiment we randomly select a subset of size 500 (50 from each class) from the test dataset and then estimate the average test log-likelihood on this subset. We experiment with both uniform and base rate proposal distribution by fixing the number of AIS runs and varying the number of intermediate distributions, K . We also estimate the test log-likelihood on the chosen test subset using the AIS and CSL estimators, for comparison with the RAISE estimates. We keep the number of Gibbs steps used in the AIS and CSL estimators equal.

We observe that, only when K is very large and proposal distribution is uniform, the estimator provides conservative estimates. However, the baserate proposal gives overestimates even when $K = 10000$. The lower bound on the test log-likelihood is similar to that of CSL estimate for RBM20. However, unlike the case for CSL, for larger RBMs, the RAISE estimates matches closely with the AIS estimates.

5 Conclusion

Calculating the average test log-likelihood of a learnt RBM is important for evaluating different learning strategies. In this paper we present extensive empirical analysis of the sampling based estimators for average test log-likelihood to gain insight into the performance of these methods. We experiment with RBMs with 20, 200 and 500 hidden

Table 2. CSL estimate of \mathcal{L} for $T = 100$ and for various values of M , keeping N constant. **Ground truth:** $\log \mathbf{Z} = 230.61$, $\mathcal{L} = -141.24$ for **RBM20**

a. Unbiased CSL

N	$n = 500$			$n = 200$		$n = 20$		
	M	$\hat{\mathcal{L}}$	σ^2	$\hat{\mathcal{L}}$	σ^2	M	$\hat{\mathcal{L}}$	σ^2
100×10^3	250	-296.18	5.90	-183.67	2.31	500	-153.83	0.83
	500	-284.25	5.87	-175.54	1.65	1000	-150.39	0.58
	1000	-272.62	3.92	-167.70	1.45	2000	-147.94	0.31
200×10^3	500	-270.14	4.56	-172.41	1.65	1000	-148.61	0.35
	1000	-257.29	3.19	-165.04	1.01	2000	-146.31	0.20
	2000	-244.87	2.49	-158.76	0.68	4000	-144.44	0.09

b. Biased CSL with large Burn-in

100×10^3	250	-219.34	4.64	-176.94	1.89	500	-151.18	0.53
	500	-208.12	3.86	-169.26	1.47	1000	-148.41	0.37
	1000	-197.22	2.17	-162.05	0.92	2000	-146.22	0.25
200×10^3	500	-198.84	2.61	-167.04	1.56	1000	-147.17	0.26
	1000	-188.32	1.65	-160.11	0.93	2000	-145.27	0.17
	2000	-179.69	1.24	-154.05	0.54	4000	-143.52	0.09

c. Biased CSL with small Burn-in

100×10^3	250	-187.83	2.52	-166.91	1.04	500	-147.82	0.22
	500	-178.12	1.99	-160.23	0.98	1000	-145.54	0.23
	1000	-169.56	1.50	-154.03	0.54	2000	-143.74	0.20
200×10^3	500	-177.09	1.89	-159.31	0.99	1000	-145.31	0.19
	1000	-168.82	1.42	-153.27	0.59	2000	-143.57	0.14
	2000	-161.87	1.45	-148.01	0.39	4000	-142.06	0.13

units. We observed that the AIS estimator delivers good estimates with low variance. We also observe that the proposal distribution does not seem to have much influence on the estimate.

Compared to the AIS estimate the CSL estimate is poorer and its variance is high especially for the RBM200 and RBM500. The estimated value also differ significantly with that of AIS. However, CSL is a much simpler estimator computationally. We also showed that better estimates can be obtained with less computational effort by using multiple independent chains to generate samples. The biased CSL with a small burn-in provides the best estimate.

Unlike AIS, the RAISE gives conservative estimates. More importantly, for large RBMs, the deviation of RAISE estimate from the AIS estimate is not very large com-

Table 3. RAISE estimate of \mathcal{L} with 500 test set, **Ground truth:** $\mathcal{L} = -142.39$ for **RBM20**. The baserate AIS with $K = 10000$, $M = 500$ is used. For the CSL, 5000×10^3 samples are used to make the number of Gibbs steps equal to that of AIS.

n	K		1000	10000	AIS	CSL
	Proposal					
20	Uniform		-147.25	-145.99	-142.38	-143.58
	Baserate		-146.97	-144.14		
200	Uniform		-109.29	-112.46	-112.96	-142.64
	Baserate		-110.42	-109.01		
500	Uniform		-114.75	-118.02	-116.46	-154.76
	Baserate		-108.75	-112.04		

pared to that of the deviation of CSL estimate from the AIS estimate. It means, for large K , the RAISE estimate will have tighter lower bound than the CSL estimate. However it is computationally much more expensive. The conservativeness of RAISE may not be enough to justify the high computational cost.

Since large hidden unit RBMs are an important part of deep networks such as stacked-RBMs and DBNs, one needs efficient estimators to evaluate the learnt networks. Our empirical study indicates that there may be much scope for improving CSL like estimators to come up with computationally simple methods to get good estimates of the average test log-likelihood.

References

1. Bengio, Y., Yao, L., Cho, K.: Bounding the test log-likelihood of generative models. arXiv preprint arXiv:1311.6184 (2013)
2. Burda, Y., Grosse, R.B., Salakhutdinov, R.: Accurate and Conservative Estimates of MRF Log-likelihood using Reverse Annealing. arXiv preprint arXiv:1412.8566 (2014)
3. Fischer, A., Igel, C.: An Introduction to Restricted Boltzmann Machines. In: Alvarez, L., Mejail, M., Gomez, L., Jacobo, J. (eds.) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Lecture Notes in Computer Science, vol. 7441, pp. 14–36. Springer Berlin Heidelberg (2012)
4. Freund, Y., Haussler, D.: Unsupervised learning of distributions on binary vectors using two layer networks. In: Moody, J., Hanson, S., Lippmann, R. (eds.) Advances in Neural Information Processing Systems 4, pp. 912–919. Morgan-Kaufmann (1992)
5. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. *Neural computation* 14(8), 1771–1800 (2002)
6. MacKay, D.J.: Information theory, inference, and learning algorithms, vol. 7. Cambridge university press Cambridge (2003)
7. Neal, R.: Annealed importance sampling. *Statistics and Computing* 11(2), 125–139 (2001)
8. Salakhutdinov, R., Murray, I.: On the quantitative analysis of deep belief networks. In: Proceedings of the 25th international conference on Machine learning. pp. 872–879. ACM (2008)
9. Smolensky, P.: Information processing in dynamical systems: Foundations of harmony theory (1986)