

---

# Training Restricted Boltzmann Machine by Perturbation

---

**Siamak Ravanbakhsh, Russell Greiner**  
 Department of Computing Science  
 University of Alberta  
 {mravanba, rgreiner@ualberta.ca}

**Brendan J. Frey**  
 Prob. and Stat. Inf. Group  
 University of Toronto  
 frey@psi.toronto.edu

## Abstract

A new approach to maximum likelihood learning of discrete graphical models and RBM in particular is introduced. Our method, Perturb and Descend (PD) is inspired by two ideas (I) perturb and MAP method for sampling (II) learning by Contrastive Divergence minimization. In contrast to perturb and MAP, PD leverages training data to learn the models that do not allow efficient MAP estimation. During the learning, to produce a sample from the current model, we start from a training data and descend in the energy landscape of the “perturbed model”, for a fixed number of steps, or until a local optima is reached. For RBM, this involves linear calculations and thresholding which can be very fast. Furthermore we show that the amount of perturbation is closely related to the temperature parameter and it can regularize the model by producing robust features resulting in sparse hidden layer activation.

## 1 Introduction

The common procedure in learning a Probabilistic Graphical Model (PGM) is to maximize the likelihood of observed data, by updating the model parameters along the gradient of the likelihood. This gradient step requires inference on the current model, which may be performed using deterministic or a Markov Chain Monte Carlo (MCMC) procedure [1]. Intuitively, the gradient step attempts to update the parameters to *increase* the unnormalized probability of the observation, while *decreasing* the sum of unnormalized probabilities over all states –i.e., the partition function. The first part of the update is known as positive phase and the second part is referred to as the negative phase. An efficient alternative is Contrastive Divergence (CD) [2] training, in which the negative phase only decreases the probability of the configurations that are in the vicinity of training data. In practice these neighboring states are sampled by taking few steps on Markov chains that are initialized by training data.

Recently Perturbation methods combined with efficient Maximum A Posteriori (MAP) solvers were used to efficiently sample PGMs [3, 4, 5]. Here a basic idea from extreme value theory is used, which states that the MAP assignments for particular perturbations of any Gibbs distribution can replace unbiased samples from the unperturbed model [6]. In practice, however, models are not perturbed in the ideal form and approximations are used [4]. Hazan *et al.* show that lower order approximations provide an upper bound on the partition function [7]. This suggest that perturb and MAP sampling procedure can be used in the negative phase to maximize a lower bound on the log-likelihood of the data. However, this is feasible only if efficient MAP estimation is possible (e.g., PGMs with submodular potentials [8]), and even so, repeated MAP estimation at each step of learning could be prohibitively expensive.

Here we propose a new approach closely related to CD and perturb and MAP to sample the PGM in the negative phase of learning. The basic idea is to perturb the model and starting at training

data, find lower perturbed-energy configurations. Then use these configurations as fantasy particles in the negative phase of learning. Although this scheme may be used for arbitrary discrete PGMs with and without hidden variables, here we consider its application to the task of training Restricted Boltzmann Machine (RBM) [9].

## 2 Background

### 2.1 Restricted Boltzmann Machine

RBM is a bipartite Markov Random Field, where the variables  $\mathbf{x} = \{\mathbf{v}, \mathbf{h}\}$  are partitioned into visible  $\mathbf{v} = [v_1, \dots, v_n]$  and hidden  $\mathbf{h} = [h_1, \dots, h_m]$  units. Because of its representation power ([10]) and relative ease of training [11], RBM is increasing used in various applications. For example as generative model for movie ratings [12], speech [13] and topic modeling [14]. Most importantly it is used in construction of deep neural architectures [15, 16].

RBM models the joint distribution over hidden and visible units by

$$\mathbb{P}(\mathbf{h}, \mathbf{v} | \theta) = \frac{1}{Z(\theta)} e^{-E(\mathbf{h}, \mathbf{v}, \theta)}$$

where  $Z(\theta) = \sum_{\mathbf{h}, \mathbf{v}} e^{-E(\mathbf{h}, \mathbf{v}, \theta)}$  is the normalization constant (*a.k.a.* partition function) and  $E$  is the energy function. Due to its bipartite form, conditioned on the visible (hidden) variables the hidden (visible) variables in an RBM are independent of each other:

$$\mathbb{P}(\mathbf{h} | \mathbf{v}, \theta) = \prod_{1 \leq j \leq m} \mathbb{P}(h_j | \mathbf{v}, \theta) \quad \text{and} \quad \mathbb{P}(\mathbf{v} | \mathbf{h}, \theta) = \prod_{1 \leq i \leq n} \mathbb{P}(v_i | \mathbf{h}, \theta) \quad (1)$$

Here we consider the energy function of binary RBM, where  $h_j, v_i \in \{0, 1\}$

$$E(\mathbf{v}, \mathbf{h}, \theta) = - \left( \sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} v_i W_{i,j} h_j + \sum_{1 \leq i \leq n} a_i v_i + \sum_{1 \leq j \leq m} b_j h_j \right) = - \left( \mathbf{v}^T \mathbf{W} \mathbf{h} + \mathbf{a}^T \mathbf{v} + \mathbf{b}^T \mathbf{h} \right)$$

The model parameter  $\theta = (\mathbf{W}, \mathbf{a}, \mathbf{b})$ , consists of the matrix of  $n \times m$  real valued pairwise interactions  $\mathbf{W}$ , and local fields (*a.k.a.* bias terms)  $\mathbf{a}$  and  $\mathbf{b}$ . The marginal over visible units is

$$\mathbb{P}(\mathbf{v} | \theta) = \frac{1}{Z(\theta)} \sum_{\mathbf{h}} \mathbb{P}(\mathbf{v}, \mathbf{h} | \theta)$$

Given a data-set  $\mathcal{D} = \{\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(N)}\}$ , maximum-likelihood learning of the model seeks the maximum of the averaged log-likelihood:

$$\ell(\theta) = \frac{1}{N} \sum_{\mathbf{v}^{(k)} \in \mathcal{D}} \log(\mathbb{P}(\mathbf{v}^{(k)} | \theta)) \quad (2)$$

$$= -\frac{1}{N} \sum_{\mathbf{v}^{(k)} \in \mathcal{D}} \prod_{1 \leq j \leq m} \left( 1 + \exp \left( \sum_{1 \leq i \leq n} v_i^{(k)} W_{i,j} \right) \right) - \log(Z(\theta)) \quad (3)$$

Simple calculations gives us the derivative of this objective wrt  $\theta$ :

$$\begin{aligned} \partial \ell(\theta) / \partial W_{i,j} &= \frac{1}{N} \sum_{\mathbf{v}^{(k)} \in \mathcal{D}} \mathbb{E}_{\mathbb{P}(h_j | \mathbf{v}^{(k)}, \theta)} [v_i^{(k)} h_j] - \mathbb{E}_{\mathbb{P}(v_i, h_j | \theta)} [v_i h_j] \\ \partial \ell(\theta) / \partial a_i &= \frac{1}{N} \sum_{\mathbf{v}^{(k)} \in \mathcal{D}} v_i^{(k)} - \mathbb{E}_{\mathbb{P}(v_i | \theta)} [v_i] \\ \partial \ell(\theta) / \partial b_j &= \frac{1}{N} \sum_{\mathbf{v}^{(k)} \in \mathcal{D}} \mathbb{E}_{\mathbb{P}(h_j | \mathbf{v}^{(k)}, \theta)} [h_j] - \mathbb{E}_{\mathbb{P}(h_j | \theta)} [h_j] \end{aligned}$$

where the first and the second terms in each line correspond to positive and negative phase respectively. It is easy to calculate  $\mathbb{P}(h_j|\mathbf{v}^{(k)}, \theta)$ , required in the positive phase. The negative phase, however, requires unconditioned samples from the current model, which may require long mixing of the Markov chain.

Note that the same form of update appears when learning any Markov Random Field, regardless of the form of graph and presence of hidden variables. In general the gradient update has the following form

$$\nabla_{\theta_I} \ell(\theta) = \mathbb{E}_{\mathcal{D}, \theta} [\phi_I(\mathbf{x}_I)] - \mathbb{E}_{\theta} [\phi_I(\mathbf{x}_I)] \quad (4)$$

where  $\phi_I(\mathbf{x}_I)$  is the sufficient statistics corresponding to parameter  $\theta_I$ . For example the sufficient statistics for variable interactions  $W_{i,j}$  in an RBM is  $\phi_{i,j}(v_i, h_j) = v_i h_j$ . Note that  $\theta$  in calculating the expectation of the first term appears only if hidden variables are present.

## 2.2 Contrastive Divergence Training

In estimating the second term in the update of eq(4), we can sample the model with the training data in mind. To this end, CD samples the model by initializing the Markov chain to data points and running it for  $K$  steps. This is repeated each time we calculate the gradient. At the limit of  $K \rightarrow \infty$ , this gives unbiased samples from the current model, however using only few steps, CD performs very well in practice [2]. For RBM this Markov chain is simply a block Gibbs sampler with visible and hidden units are sampled alternatively using eq(1).

It is also possible to initialize the chain to the training data at the beginning of learning and during each calculation of gradient run the chain from its previous state. This is known as persistent CD [17] or stochastic maximum likelihood [18].

## 2.3 Sampling by Perturb and MAP

Assuming that it is possible to efficiently obtain the MAP assignment in an MRF, it is possible to use perturbation methods to produce unbiased samples. These samples then may be used in the negative phase of learning.

Let  $\tilde{E}(\mathbf{x}) = E(\mathbf{x}) - \epsilon(\mathbf{x})$  denote the perturbed energy function, where the perturbation for each  $\mathbf{x}$  is a sample from standard Gumbel distribution  $\epsilon(\mathbf{x}) \sim \gamma(\epsilon) = \exp(\epsilon - \exp(-\epsilon))$ . Also let  $\tilde{\mathbb{P}}(\mathbf{x}) \propto \exp(-\tilde{E})$  denote the perturbed distribution. Then the MAP assignment  $\arg_{\mathbf{x}} \max \tilde{\mathbb{P}}(\mathbf{x})$  is an unbiased sample from  $\mathbb{P}(\mathbf{x})$ . This means we can sample  $\mathbb{P}(\mathbf{x})$  by repeatedly perturbing it and finding the MAP assignment. To obtain samples from a Gumbel distribution we transform samples from uniform distribution  $u \sim \mathcal{U}(0, 1)$  by  $\epsilon \leftarrow -\log(-\log(u))$ .

The following lemma clarifies the connection between Gibbs distribution and the MAP assignment in the perturbed model.

**Lemma 1** ([6]). *Let  $\{E(\mathbf{x})\}_{\mathbf{x} \in \mathcal{X}}$  and  $E(\mathbf{x}) \in \mathbb{R}$ . Define the perturbed values as  $\tilde{E}(\mathbf{x}) = E(\mathbf{x}) - \epsilon(\mathbf{x})$ , when  $\epsilon(\mathbf{x}) \sim \gamma(\epsilon) \forall \mathbf{x} \in \mathcal{X}$  are IID samples from standard Gumbel distribution. Then*

$$Pr(\arg_{\mathbf{x} \in \mathcal{X}} \max \{-\tilde{E}(\mathbf{x})\} = \hat{\mathbf{x}}) = \frac{\exp(-E(\hat{\mathbf{x}}))}{\sum_{\mathbf{y} \in \mathcal{X}} \exp(-E(\mathbf{y}))} \quad (5)$$

Since the domain  $\mathcal{X}$ , of joint assignments grows exponentially with the number of variables, we can not find the MAP assignment efficiently. As an approximation we may use fully decomposed noise  $\epsilon(\mathbf{x}) = \sum_i \epsilon(x_i)$  [4]. This corresponds to adding a Gumbel noise to each assignment of unary potentials. In the case of RBM's parametrization, this corresponds to adding the difference of two random samples from a standard Gumbel distribution (which is basically a sample from a *logistic distribution*) to biases (e.g.,  $\tilde{a}_i = a_i + \epsilon(v_i = 1) - \epsilon(v_i = 0)$ ). Alternatively a second order approximation may perturb a combination of binary and unary potentials such that each variable is included once (Section 3.2)

### 3 Perturb and Descend Learning

Feasibility of sampling using perturb and MAP depends on availability of efficient optimization procedures. However MAP estimation is in general NP-hard [19] and only a limited class of MRFs allow efficient energy minimization [8]. We propose an alternative to perturb and MAP that is suitable when inference is employed within the context of learning. Since first and second order perturbations in perturb and MAP, upper bound the partition function [7], likelihood optimization using this method is desirable (e.g., [20]). On the other hand since the model is trained on a data-set, we may leverage the training data in sampling the model.

Similar to CD at each step of the gradient we start from training data. In order to produce fantasy particles of the negative phase we perturb the current model and take several steps towards lower energy configurations. We may take enough steps to reach a local optima or stop midway.

Let  $\tilde{\theta} = (\tilde{\mathbf{W}}, \tilde{\mathbf{a}}, \tilde{\mathbf{b}})$  denote the perturbed model. For RBM, each step of this block coordinate descend takes the following form

$$\mathbf{v} \leftarrow \tilde{\mathbf{a}} + \tilde{\mathbf{W}}\mathbf{h} > 0 \quad (6)$$

$$\mathbf{h} \leftarrow \tilde{\mathbf{b}} + \tilde{\mathbf{W}}^T \mathbf{v} > 0 \quad (7)$$

where starting from  $\mathbf{v} = \mathbf{v}^{(k)} \in \mathcal{D}$ ,  $\mathbf{h}$  and  $\mathbf{v}$  are repeatedly updated for  $K$  steps or until the update above has no effect (i.e., a local optima is reached). The final configuration is then used as the fantasy particle in the negative phase of learning.

#### 3.1 Amount of Perturbations

To see the effect of the amount of perturbations we simply multiplied the noise  $\epsilon$  by a constant  $\beta$  – i.e.,  $\beta > 1$  means we perturbed the model with larger noise values. Going back to Lemma1, we see that any multiplication of noise can be compensated by a change of temperature of the energy function – i.e., for  $\beta = \frac{1}{T}$ , the  $\arg_{\mathbf{x}} \max \tilde{E}(\mathbf{x}) = \arg_{\mathbf{x}} \max \frac{1}{T} E(\mathbf{x}) - \beta \epsilon(\mathbf{x})$  remains the same. However here we are only changing the noise without changing the energy.

Here we provide some intuition about the potential effect of increasing perturbations. Experimental results seem to confirm this view. For  $\beta > 0$ , in the negative phase of learning, we are lowering the probability of configurations that are at a “larger distance” from the training data, compared to training with  $\beta = 1$ . This can make the model more robust as it puts more effort into removing false valleys that are distant from the training data, while less effort is made to remove (false) valleys that are closer to the training data.

#### 3.2 Second Order Perturbations for RBM

As discussed in Section 2.3 a first order perturbation of  $\theta$ , only injects noise to local potentials:

$$\tilde{a}_i = a_i + \epsilon(v_i = 1) - \epsilon(v_i = 0) \quad \text{and} \quad \tilde{b}_i = b_i + \epsilon(h_i = 1) - \epsilon(h_i = 0)$$

In a second order perturbation we may perturb a subset of non-overlapping pairwise potentials as well as unary potentials over the remaining variables. In doing so it is desirable to select the pairwise potentials with higher influence – i.e., larger  $|W_{i,j}|$  values. With  $n$  visible and  $m$  hidden variables, we can use Hungarian maximum bipartite matching algorithm to find  $\min(m, n)$  most influential interactions [21].

Once influential interactions are selected, we need to perturb the corresponding  $2 \times 2$  factors with Gumbel noise as well as the bias terms for all the variables that are not covered. A simple calculation shows that perturbation of the  $2 \times 2$  potentials in RBM corresponds to perturbing  $W_{i,j}$  as well as  $a_i$  and  $b_j$  as follows

$$\begin{aligned} \tilde{W}_{i,j} &= W_{i,j} + \epsilon(1,1) - \epsilon(0,1) - \epsilon(1,0) + \epsilon(0,0) \\ \tilde{a}_i &= a_i - \epsilon(0,0) + \epsilon(0,1) \\ \tilde{b}_j &= b_j - \epsilon(0,0) + \epsilon(1,0) \quad \epsilon(0,0), \epsilon(0,1), \epsilon(1,0), \epsilon(1,1) \sim \gamma(\epsilon) \end{aligned}$$

where  $\epsilon(y, z)$  is basically the injected noise to the pairwise potential assignment for  $v_i = y$  and  $h_j = z$ .

## References

- [1] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, T. Dietterich, Ed. The MIT Press, 2009, vol. 2009, no. 4.
- [2] G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.
- [3] G. Papandreou and A. L. Yuille, “Gaussian sampling by local perturbations,” in *Advances in Neural Information Processing Systems*, 2010, pp. 1858–1866.
- [4] —, “Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models,” in *ICCV*. IEEE, 2011, pp. 193–200.
- [5] T. Hazan, S. Maji, and T. Jaakkola, “On sampling from the gibbs distribution with random maximum a-posteriori perturbations,” *arXiv preprint arXiv:1309.7598*, 2013.
- [6] E. J. Gumbel, *Statistical theory of extreme values and some practical applications: a series of lectures*. US Government Printing Office Washington, 1954, vol. 33.
- [7] T. Hazan and T. Jaakkola, “On the partition function and random maximum a-posteriori perturbations,” *arXiv preprint arXiv:1206.6410*, 2012.
- [8] V. Kolmogorov and R. Zabini, “What energy functions can be minimized via graph cuts?” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 2, pp. 147–159, 2004.
- [9] P. Smolensky, “Information processing in dynamical systems: Foundations of harmony theory,” 1986.
- [10] N. Le Roux and Y. Bengio, “Representational power of restricted boltzmann machines and deep belief networks,” *Neural Computation*, vol. 20, no. 6, pp. 1631–1649, 2008.
- [11] G. Hinton, “A practical guide to training restricted boltzmann machines,” *Momentum*, vol. 9, no. 1, 2010.
- [12] R. Salakhutdinov, A. Mnih, and G. Hinton, “Restricted boltzmann machines for collaborative filtering,” in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 791–798.
- [13] A.-R. Mohamed and G. Hinton, “Phone recognition using restricted boltzmann machines,” in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4354–4357.
- [14] G. E. Hinton and R. Salakhutdinov, “Replicated softmax: an undirected topic model,” in *Advances in neural information processing systems*, 2009, pp. 1607–1614.
- [15] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [16] Y. Bengio, “Learning deep architectures for AI,” *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.
- [17] T. Tieleman, “Training restricted boltzmann machines using approximations to the likelihood gradient,” in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1064–1071.
- [18] L. Younes, “Parametric inference for imperfectly observed gibbsian fields,” *Probability Theory and Related Fields*, vol. 82, no. 4, pp. 625–645, 1989.
- [19] S. E. Shimony, “Finding maps for belief networks is np-hard,” *Artificial Intelligence*, vol. 68, no. 2, pp. 399–410, 1994.
- [20] M. J. Wainwright and M. I. Jordan, “Graphical models, exponential families, and variational inference,” *Foundations and Trends® in Machine Learning*, vol. 1, no. 1-2, pp. 1–305, 2008.
- [21] J. Munkres, “Algorithms for the assignment and transportation problems,” *Journal of the Society for Industrial & Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.