



# Low-Dimensional Perturb-and-MAP Approach for Learning Restricted Boltzmann Machines

Jakub M. Tomczak<sup>1</sup> · Szymon Zaręba<sup>1</sup> · Siamak Ravanbakhsh<sup>2</sup> · Russell Greiner<sup>3</sup>

Published online: 3 October 2018  
© The Author(s) 2018

## Abstract

This paper introduces a new approach to maximum likelihood learning of the parameters of a restricted Boltzmann machine (RBM). The proposed method is based on the Perturb-and-MAP (PM) paradigm that enables sampling from the Gibbs distribution. PM is a two step process: (i) perturb the model using Gumbel perturbations, then (ii) find the maximum a posteriori (MAP) assignment of the perturbed model. We show that under certain conditions the resulting MAP configuration of the perturbed model is an unbiased sample from the original distribution. However, this approach requires an exponential number of perturbations, which is computationally intractable. Here, we apply an approximate approach based on the first order (low-dimensional) PM to calculate the gradient of the log-likelihood in binary RBM. Our approach relies on optimizing the energy function with respect to observable and hidden variables using a greedy procedure. First, for each variable we determine whether flipping this value will decrease the energy, and then we utilize the new local maximum to approximate the gradient. Moreover, we show that in some cases our approach works better than the standard coordinate-descent procedure for finding the MAP assignment and compare it with the Contrastive Divergence algorithm. We investigate the quality of our approach empirically, first on toy problems, then on various image datasets and a text dataset.

**Keywords** Unsupervised deep learning · Gumbel perturbation · Restricted Boltzmann machine · Greedy optimization

---

✉ Jakub M. Tomczak  
jakub.tomczak@pwr.edu.pl

Szymon Zaręba  
szymon.zareba@pwr.edu.pl

Siamak Ravanbakhsh  
siamakx@cs.ubc.ca

Russell Greiner  
rgreiner@ualberta.ca

<sup>1</sup> Faculty of Computer Science and Management, Wrocław University of Science and Technology, wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland

<sup>2</sup> University of British Columbia, Vancouver, BC V6T1Z4, Canada

<sup>3</sup> University of Alberta, Edmonton, AB T6G 2E8, Canada

## 1 Introduction

The commonly used procedure for learning parameters of Markov random fields (MRFs) is to maximize the log-likelihood function for observed data, by updating the model parameters along the gradient of the objective. This gradient step requires inference on the current model, which is often approximated using a deterministic or Markov Chain Monte Carlo (MCMC) procedure [10]. In general, the gradient step attempts to update the parameters to increase the unnormalized probability of the observations (*clamped* or *positive phase*), while decreasing the sum of unnormalized probabilities over all states, i.e., the partition function (*unclamped* or *negative phase*). The positive phase is rather straightforward, but the negative phase is difficult to perform, mainly because of the complexity of computing the partition function. Therefore, in order to overcome the issue, alternative approaches have been proposed, such as Contrastive Divergence (CD) [7] or Pseudo-likelihood or ratio matching (see [19] for a review). In the context of Restricted Boltzmann Machines (RBMs), the widely-used training procedure is CD, which utilizes the MCMC in the negative phase to decrease the probability of the configurations that are in the vicinity of training data.

Recently, perturbation methods combined with efficient maximum a posteriori (MAP) solvers (Perturb-and-MAP, PM) were used to efficiently sample from the MRF [3,5,6,17,23,24]. The main idea behind PM is based on extreme value theory, which states that the MAP assignments for particular perturbations of any Gibbs distribution can produce unbiased samples from the unperturbed model [23]. In practice, however, models cannot be perturbed in the ideal form and  $k$ th-order approximations are used. In [5] these order approximations are used to bound from above the partition function, suggesting that PM-based sampling procedures 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 solvers are accessible, e.g., MRF with submodular potentials [11], and even so, repeated MAP estimation at each step of learning could be prohibitively expensive. Nonetheless, this PM approach has been successfully applied to different problems, e.g., computer vision [24], feature learning using Cardinality RBM [15], structured prediction [1,30], boundary of object annotation in images [18], image segmentation [8,20]. The idea of PM was also used in a preliminary study on learning RBMs [26].

In [26] an approach closely related to CD and Perturb-and-MAP for sampling from RBM in the negative phase of learning was proposed. The basic idea was to perturb the parameters of the model, then starting from the training data, find the local optima of the energy function using block-coordinate descent method. This produces samples from the joint distribution (over both the hidden and the visible variables) in the RBM. We call this approach perturb-and-coordinate descent (P&CD).

In this work, we rely on a different strategy to find MAP assignments, namely, *greedy search for local optimal assignments for both visible and hidden variables*. Since learning RBM using the PM approach requires performing optimization in each mini-batch, it is critical that we approximate MAP assignments quickly. That is why we exploit the greedy optimization technique, hoping it will produce reasonable local optimal assignments. Moreover, we show a close relationship of our proposed greedy-based method to the coordinate descent method in the context of the RBM.

Using three experiments, we show that using the PM approach for learning the RBM is promising: (1) a toy problem in which the exact log-likelihood can be computed; (2) several black-and-white image benchmark datasets: letters, MNIST, Omniglot, Frey Face, Handwritten character recognition; and (3) a text analysis of blog posts. Our empirical studies

show that our proposed PM-based learning procedure for RBM in general performs similarly to CD, but in some cases can be better.

The contributions of the paper are:

- We propose to use a *greedy optimization* method for finding MAP solutions in the PM approach for learning RBMs.
- We perform an empirical comparison of the PM approach against the Contrastive Divergence. We compare the methods both on unsupervised and supervised data. Given that [26] do not report any experimental results, this is the first empirical evaluation of the PM approach utilized for learning RBMs.

## 2 Background

### 2.1 Restricted Boltzmann Machine

*The model* The binary restricted Boltzmann machine (RBM) is a bipartite MRF that defines the joint distribution over binary visible and hidden units [29], where  $\mathbf{x} \in \{0, 1\}^D$  are the visibles and  $\mathbf{h} \in \{0, 1\}^M$  are the hidden. The relationships among units are specified through the *energy function*:

$$E(\mathbf{x}, \mathbf{h}|\Theta) = -\mathbf{x}^\top \mathbf{W} \mathbf{h} - \mathbf{b}^\top \mathbf{x} - \mathbf{c}^\top \mathbf{h}, \tag{1}$$

where  $\Theta = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$  is a set of parameters,  $\mathbf{W} \in \mathbb{R}^{D \times M}$ ,  $\mathbf{b} \in \mathbb{R}^D$ , and  $\mathbf{c} \in \mathbb{R}^M$  are, respectively, weights, visible biases, and hidden biases. For the energy function in Eq. 1, RBM is defined by the *Gibbs distribution*:

$$p(\mathbf{x}, \mathbf{h}|\Theta) = \frac{1}{Z(\Theta)} \exp(-E(\mathbf{x}, \mathbf{h}|\Theta)), \tag{2}$$

where  $Z(\Theta) = \sum_{\mathbf{x}} \sum_{\mathbf{h}} \exp(-E(\mathbf{x}, \mathbf{h}|\Theta))$  is the *partition function*. The marginal probability over visibles (the likelihood of an observation) is again the Gibbs distribution  $p(\mathbf{x}|\Theta) = \frac{1}{Z(\Theta)} \exp(-F(\mathbf{x}|\Theta))$ , where  $F(\cdot)$  is the *free energy*:<sup>1</sup>

$$F(\mathbf{x}|\Theta) = -\mathbf{b}^\top \mathbf{x} - \sum_j \log \left( 1 + \exp(\mathbf{c}_j + (\mathbf{W}_{\cdot j})^\top \mathbf{x}) \right). \tag{3}$$

RBM possesses the very useful property that the conditional distribution over the hidden units factorizes given the visible units and *vice versa*, which yields the following:<sup>2</sup>

$$p(\mathbf{h}_m = 1|\mathbf{x}, \Theta) = \sigma(\mathbf{c}_m + (\mathbf{W}_{\cdot m})^\top \mathbf{x}), \tag{4}$$

$$p(\mathbf{x}_d = 1|\mathbf{h}, \Theta) = \sigma(\mathbf{b}_d + \mathbf{W}_d \cdot \mathbf{h}). \tag{5}$$

*Learning* Given data  $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$ , we can train RBM using the maximum likelihood approach that seeks the maximum of the averaged log-likelihood (LL):

$$\ell(\Theta) = \frac{1}{N} \sum_{\mathbf{x}_n \in \mathcal{D}} \log p(\mathbf{x}_n|\Theta). \tag{6}$$

<sup>1</sup> We use the following notation: for given matrix  $\mathbf{A}$ ,  $\mathbf{A}_{ij}$  is its element at location  $(i, j)$ ,  $\mathbf{A}_{\cdot j}$  denotes its  $j$ th column,  $\mathbf{A}_i$  denotes its  $i$ th row, and for given vector  $\mathbf{a}$ ,  $a_i$  is its  $i$ th element.

<sup>2</sup>  $\sigma(\mathbf{x}) = \left[ \frac{1}{1+\exp(-x_1)}, \dots, \frac{1}{1+\exp(-x_D)} \right]^\top$  is the element-wise sigmoid function.

The gradient of the learning objective  $\ell(\Theta)$  wrt  $\theta \in \Theta$  takes the following form:

$$\nabla_{\theta} \ell(\Theta) = \frac{-1}{N} \sum_{n=1}^N \left( \nabla_{\theta} F(\mathbf{x}_n | \Theta) - \sum_{\hat{\mathbf{x}}} p(\hat{\mathbf{x}} | \Theta) \nabla_{\theta} F(\hat{\mathbf{x}} | \Theta) \right). \quad (7)$$

In general, the gradient in Eq. 7 cannot be computed analytically because the second term requires summing over all configurations of visibles. One way to sidestep this issue is the standard stochastic approximation of replacing the expectation under  $p(\mathbf{x} | \Theta)$  by a sum over  $S$  samples  $\{\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_S\}$  drawn according to  $p(\mathbf{x} | \Theta)$  [e.g.19]:

$$\nabla_{\theta} \ell(\Theta) \approx - \left( \frac{1}{N} \sum_{n=1}^N \nabla_{\theta} F(\mathbf{x}_n | \Theta) - \frac{1}{S} \sum_{s=1}^S \nabla_{\theta} F(\hat{\mathbf{x}}_s | \Theta) \right). \quad (8)$$

A different approach, *Contrastive Divergence* (CD), approximates the expectation under  $p(\mathbf{x} | \Theta)$  in Eq. 7 by a sum over samples  $\bar{\mathbf{x}}_n$  drawn from a distribution obtained by applying  $K$  steps of the block-Gibbs sampling procedure:

$$\nabla_{\theta} \ell(\Theta) \approx \frac{-1}{N} \sum_{n=1}^N \left( \nabla_{\theta} F(\mathbf{x}_n | \Theta) - \nabla_{\theta} F(\bar{\mathbf{x}}_n | \Theta) \right). \quad (9)$$

The original CD [7] used  $K$  steps of the Gibbs chain, starting from each data point  $\mathbf{x}_n$  to obtain a sample  $\bar{\mathbf{x}}_n$  and is restarted after every parameter update. An alternative approach, *Persistent Contrastive Divergence* (PCD) does not restart the chain after each update; this typically results in slower convergence rate but eventually better performance [32].

## 2.2 Perturb-and-MAP Approach

Sampling from a MRF, including a RBM, is problematic due to the difficulty of calculating the partition function. However, assuming that a MAP assignment in the MRF can be found efficiently, it is possible to take advantage of random perturbation methods to obtain unbiased samples [23]. Further, the unbiased samples can be utilized in the standard stochastic approximation of the log-likelihood gradient to calculate the second sum in Eq. 8.

Let us consider a system described by variables  $\mathbf{z}$ , and an energy  $\mathcal{E}(\mathbf{z})$ .<sup>3</sup> It turns out that a probability distribution of MAP assignments of the perturbed energy using Gumbel-distributed random variables is equivalent to the Gibbs distribution. The following theorem clarifies the connection between the Gumbel distribution and the Gibbs distribution [23]:

**Theorem 1** [4] *Let  $\mathcal{E}(\mathbf{z}) \in \mathbb{R}$  be the energy of a system where  $\mathbf{z} \in \mathcal{Z}$  is a discrete-valued vector. If  $\gamma(\mathbf{z})$  are i.i.d. random variables with standard Gumbel distribution whose cdf is given by  $G(z; 0, 1) = \exp(-\exp(-z))$ , then*

$$\Pr(\mathbf{z} = \arg \max_{\hat{\mathbf{z}}} \{\mathcal{E}(\hat{\mathbf{z}}) + \gamma(\hat{\mathbf{z}})\}) = \frac{\exp(\mathcal{E}(\mathbf{z}))}{\sum_{\hat{\mathbf{z}}} \exp(\mathcal{E}(\hat{\mathbf{z}}))}.$$

In other words, the Perturb-and-MAP (PM) approach can be seen as a two-step generative process [24]:

(Perturb step) Add a random perturbation  $\gamma(\mathbf{z})$  to the energy  $\mathcal{E}(\mathbf{z})$ .

<sup>3</sup> Here we use a general notation for variables and energy as it refers to visibles and hidden, and the energy function Eq. 1 in RBM. Notice that in order to be consistent with the literature of the Perturb-and-MAP we use an energy without the minus sign to solve the maximization problem instead of the minimization problem.

(MAP step) Find the maximum of the perturbed energy:

$$\mathbf{z} = \arg \max_{\hat{\mathbf{z}}} \{ \mathcal{E}(\hat{\mathbf{z}}) + \gamma(\hat{\mathbf{z}}) \}. \tag{10}$$

Eventually, the theorem states that the solutions of the MAP step in the Perturb-and-MAP procedure can be seen as unbiased samples of the Gibbs distribution.

Since the domain of  $\mathbf{z}$  grows exponentially with the number of variables, it is troublesome to find the MAP assignment of the perturbed energy efficiently. Therefore, first order (low-dimensional) Gumbel perturbations are often employed [5]. Here, for the first order perturbation, the joint perturbation is fully decomposable  $\gamma(\mathbf{z}) = \sum_i \gamma(z_i)$  and it corresponds to perturbing unary potentials (i.e., biases) only [5,6,23].

### 3 Learning RBM Using PM

In order to apply the PM approach for learning RBMs we need to specify both steps of the PM process, namely, Perturb and MAP. Here, we consider low-dimensional Gumbel perturbations and two optimization methods for finding MAP assignments: coordinate-descent method [26] and greedy optimization.

#### 3.1 Perturb Step

In the case of RBM, an application of the low-dimensional Gumbel perturbations results in adding the difference of two random variables from the standard Gumbel distribution to biases  $\mathbf{b}$  and  $\mathbf{c}$  in Eq. 1:

$$\tilde{\mathbf{b}}_d = \mathbf{b}_d + \gamma(\mathbf{x}_d = 1) - \gamma(\mathbf{x}_d = 0), \quad d = 1, \dots, D, \tag{11}$$

$$\tilde{\mathbf{c}}_m = \mathbf{c}_m + \gamma(\mathbf{h}_m = 1) - \gamma(\mathbf{h}_m = 0), \quad m = 1, \dots, M, \tag{12}$$

where  $\gamma(\cdot) \sim G(\cdot; 0, 1)$  is a standard Gumbel random variable. In order to speed up computations, instead of sampling two times from standard Gumbel distribution we can take advantage of the fact that a difference of two Gumbel random variables is a sample from a logistic distribution.

Recently it has been shown theoretically that the low-dimensional perturbations can be used to draw approximate sample from the Gibbs distribution:

**Theorem 2** [33] *Let  $\gamma(\mathbf{z})$  be a sum of  $N$  low-dimensional i.i.d. perturbations with standard Gumbel distribution, i.e.,  $\gamma(\mathbf{z}) = \sum_i \gamma_i(z_i)$ . Then the distribution of configurations maximizing the energy function  $\mathcal{E}(z)$  is approximately the Gibbs distribution:*

$$\mathbb{P}[\mathbf{z} = \arg \max_{\hat{\mathbf{z}}} \{ \theta(\hat{\mathbf{z}}) + \sum_i \gamma_i(\hat{z}_i) \}] \approx \frac{\exp(\mathcal{E}(\mathbf{z}))}{Z}. \tag{13}$$

The proof of this theorem takes advantage of approximating a sum of standard Gumbel distributions with single Gumbel distribution using the moment-matching method [22]. The approximation error between the distribution of the sum of Gumbel variables and the distribution for single Gumbel variable was bounded from above using the Berry-Esseen inequality and it was shown to be small enough to be used as a valid approximation [33]. Therefore, theoretically we can apply the low-dimensional perturbations for sampling from the RBM. However, the crucial part of the PM approach lies in finding MAP solutions of the perturbed energy, that is, in formulating fast and efficient optimization procedure in the MAP step.

### 3.2 MAP Step

As we have presented, a set of  $S$  solutions of the *MAP* step in Eq. 10 can be used in the stochastic approximation of the gradient in Eq. 8. Since the low-dimensional distributions lead to the almost exact Gibbs distribution according to the Theorem 2, the feasibility of the PM approach strongly depends on the efficiency of the optimization procedure. In general, MAP estimation in MRF is NP-hard [28] and only a limited class of MRFs allow efficient energy minimization [11]. In the case of RBM, the problem of finding a solution that maximizes the energy function can be cast as the *unconstrained binary quadratic programming problem* (BQP) that is known to be NP-hard as well [21]. In the following subsections we present two alternatives to MAP step that are suitable when inference is employed within the context of learning RBMs.

#### 3.2.1 Perturb and Coordinate Descent (P&CD) Learning

In [26] it was proposed to obtain samples from the model by first perturbing the unary potentials (see Eqs. 11 and 12), and further apply block coordinate descent to optimize the energy function  $E(\mathbf{x}, \mathbf{h}|\Theta)$ . The procedure is presented in Algorithm 1 ( $\mathbb{I}(\cdot)$  denotes the element-wise indicator function).

---

**Algorithm 1:** Finding MAP assignment for a single observation using Coordinate Descent

---

**Input** :  $\mathbf{x}_{\text{data}}$ : training datum,  $K$ : number of optimization steps,  $S$ : number of approximating samples  
**Output**:  $\{(\hat{\mathbf{x}}_s, \hat{\mathbf{h}}_s)\}_{s=1}^S$ : approximate MAP solutions of the perturbed energy

```

1 for  $s = 1, \dots, S$  do
2   Perturb biases using Eq. 11 and Eq. 12;
3    $\mathbf{x}_s^{(0)} \leftarrow \mathbf{x}_{\text{data}}$ ;
4    $\mathbf{h}_s^{(0)} \leftarrow p(\mathbf{h}|\mathbf{x}_s^{(0)})$ ;
5   for  $k = 1, \dots, K$  do
6      $\mathbf{x}_s^{(k)} \leftarrow \mathbb{I}(\mathbf{W}\mathbf{h}_s^{(k-1)} + \tilde{\mathbf{b}} > 0)$ ;
7      $\mathbf{h}_s^{(k)} \leftarrow \mathbb{I}(\mathbf{W}^\top \mathbf{x}_s^{(k)} + \tilde{\mathbf{c}} > 0)$ ;
8   end
9    $\hat{\mathbf{x}}_s \leftarrow \mathbf{x}_s^{(K)}$ ;
10   $\hat{\mathbf{h}}_s \leftarrow \mathbf{h}_s^{(K)}$ ;
11 end
12 return  $\{(\hat{\mathbf{x}}_1, \hat{\mathbf{h}}_1), \dots, (\hat{\mathbf{x}}_S, \hat{\mathbf{h}}_S)\}$ ;

```

---

The procedure starts from any  $\mathbf{x} \in \mathcal{D}$ , where  $\mathbf{h}$  and  $\mathbf{x}$  are repeatedly updated for  $K$  steps and the procedure is run  $S$  times. As a result,  $S$  final configurations of visibles are used to calculate the second sum in Eq. 8 while the hiddenes can be "effectively" discarded.<sup>4</sup>

Interestingly, the procedure utilizing the coordinate descent method resembles the Contrastive Divergence algorithm in which the consecutive steps in the for-loop are performed according to the Gibbs sampler. The main difference between these two approaches is that, in the P&CD approach, the stochasticity is injected to biases and the solution of the optimization

---

<sup>4</sup> This corresponds to using mean-field approximation for the hidden variables (rather than using samples) in the parameter update.

problem (a sample) is found in a deterministic manner. On the contrary, in the Contrastive Divergence the "solution" (a sample) is found in a stochastic fashion.

### 3.2.2 Perturb and Greedy Energy Optimization (P&GEO) Learning

There are different approaches for solving the BQP, however, application of exact methods for large-scale problems is infeasible [9]. Here, motivated by the success of heuristics used for BQP [21], we propose to apply a greedy approach to search for local solutions to eventually obtain a good (local) optimum. In the context of RBM, we start from a data point and corresponding hidden variables calculated according to Eq. 4 and further, for each visible and hidden variable we look for states that greedily increase the energy. In order to get some insight of the procedure, let us consider  $d$ -th visible variable (analogical reasoning can be carried out for hidden variables). There are two possibilities: (i)  $x_d^{(k)} = 0$ , or (ii)  $x_d^{(k)} = 1$ , where  $k$  denotes the optimization step number. If  $x_d^{(k)} = 0$ , then in order to maximize the energy function we need to verify whether changing its value to 1 results in  $\mathbf{W}_d \cdot \mathbf{h}^{(k-1)} + b_d > 0$  or  $\mathbf{W}_d \cdot \mathbf{h}^{(k-1)} + b_d < 0$ . In the former case, we should change the value, while in the latter case we should keep its value unchanged. For the second possibility, i.e.,  $x_d^{(k)} = 1$ , we need to determine whether  $\mathbf{W}_d \cdot \mathbf{h}^{(k-1)} + b_d > 0$ . If it is so, we should keep its value, while if  $\mathbf{W}_d \cdot \mathbf{h}^{(k-1)} + b_d < 0$ , then we should flip the value of  $x_d^{(k)}$ . As a result we notice, that the crucial quantity here is the value of  $\mathbf{W}_d \cdot \mathbf{h}^{(k-1)} + b_d$  and if it is positive, then we should set  $x_d^{(k)}$  to 1 and 0 otherwise. The final greedy procedure is presented in Algorithm 2.

---

**Algorithm 2:** Finding MAP assignment for a single observation using Greedy Energy Optimization

---

```

Input :  $\mathbf{x}_{\text{data}}$ : training datum,  $K$ : number of optimization steps,  $S$ : number of approximating samples
Output:  $\{(\hat{\mathbf{x}}_s, \hat{\mathbf{h}}_s)\}_{s=1}^S$ : approximate MAP solutions of the perturbed energy
1 for  $s = 1, \dots, S$  do
2   Perturb biases using Eq. 11 and Eq. 12;
3    $\mathbf{x}_s^{(0)} \leftarrow \mathbf{x}_{\text{data}}$ ;
4    $\mathbf{h}_s^{(0)} \leftarrow p(\mathbf{h}|\mathbf{x}_s^{(0)})$ ;
5   for  $k = 1, \dots, K$  do
6      $\mathbf{x}_s^{(k)} \leftarrow \mathbb{I}(\mathbf{W}\mathbf{h}_s^{(k-1)} + \tilde{\mathbf{b}} > 0)$ ;
7      $\mathbf{h}_s^{(k)} \leftarrow \mathbb{I}(\mathbf{W}^T \mathbf{x}_s^{(k-1)} + \tilde{\mathbf{c}} > 0)$ ;
8   end
9    $\hat{\mathbf{x}}_s \leftarrow \mathbf{x}_s^{(K)}$ ;
10   $\hat{\mathbf{h}}_s \leftarrow \mathbf{h}_s^{(K)}$ ;
11 end
12 return  $\{(\hat{\mathbf{x}}_1, \hat{\mathbf{h}}_1), \dots, (\hat{\mathbf{x}}_S, \hat{\mathbf{h}}_S)\}$ ;

```

---

Interestingly, the greedy approach for finding MAP solutions is almost identical to the coordinate descent method (see steps 6 and 7 in Algorithms 1 and 2). The key difference lies in updating visible and hidden variables *at once*. This property may be profitable in case of training deep Boltzmann machines (DBM) because it is easy to apply the greedy method to optimize the energy function, while application of the coordinate descent for DBM is rather indirect. However, we leave investigating this issue for future research.

## 4 Experiments

In the empirical study we compare learning binary RBM using stochastic gradient descent (SGD) with P&CD and P&GEO, against the well-known Contrastive Divergence (CD). In the experiments we consider a toy problem in which exact log-likelihood can be calculated, as well as various image benchmark datasets and a text benchmark dataset.

We use the log-likelihood of the test data as the evaluation metric. For high-dimensional problems exact calculation of  $\log(Z)$  is intractable, therefore, we apply the *Annealed Importance Sampling* (AIS) procedure [27] for approximations. The AIS is performed 100 times with 10,000 temperature scales evenly spaced between 0 and 1 and 100 particles in each run. The base distribution in AIS is set to independent binary draws at the mean of the observations. Moreover, we evaluate all methods in a discriminative manner using  $k$  Nearest Neighbor classifier.

In all experiments we used the following values of the hyper-parameters. The learning rate of SGD was chosen from the set  $\{0.001, 0.01, 0.1\}$ . CD used  $K \in \{1, 5, 10\}$  steps of the block Gibbs sampling, while the number of optimization iterations of P&CD and P&GEO procedures were in the set  $\{1, 5, 10\}$ . Moreover, we performed the SGD procedure with the momentum coefficient in  $\{0, 0.9\}$  and we penalized the log-likelihood objective with the weight decay (the regularization coefficient in  $\{0, 10^{-5}, 10^{-4}, 10^{-3}\}$ ). The gradient approximation for all considered learning methods was calculated using  $S = 1$  samples.

The optimal hyper-parameters were selected using the evaluation metric. The number of iterations over the training set was determined using early stopping according to the validation data-set log-likelihood, with a look ahead of 30 epochs.

All experiments were performed on Nvidia GeForce GTX970. The code for the paper is available online: [https://github.com/szymonzareba/perturb\\_and\\_map\\_rbm](https://github.com/szymonzareba/perturb_and_map_rbm).

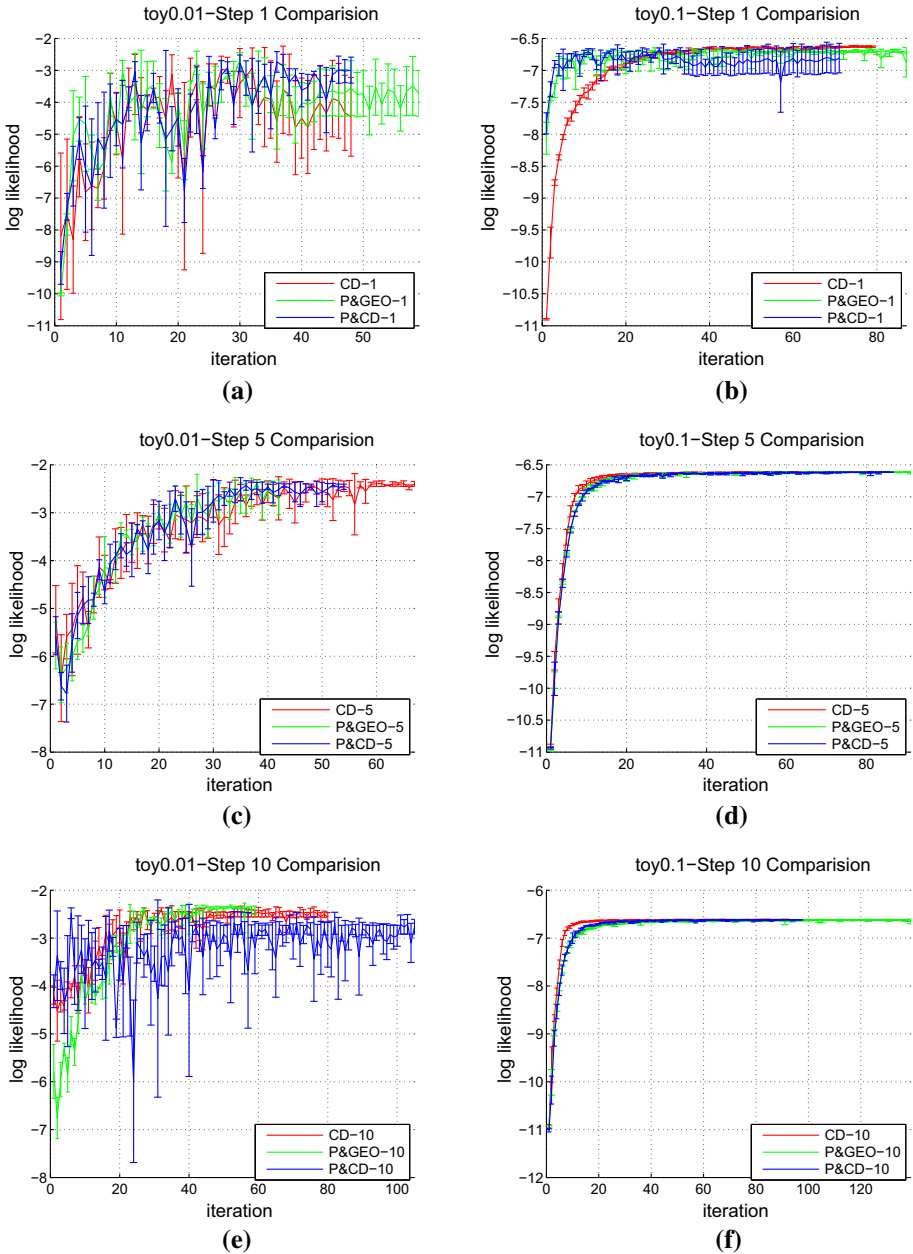
### 4.1 Toy Problem

*Setting* In order to get more insight into the performance of P&CD and P&GEO, we consider a toy example from [2], which deals with  $4 \times 4$  binary pixel images, and each image correspond to one of four possible basic modes (uncorrupted images). Further, we generate training, validation and test sets by replicating each of four basic modes and flipping each pixel independently with probability  $p \in \{0.01, 0.1\}$ . As a result, we get two datasets, in which the probability  $p$  controls the effective distance between the modes. Here, we can study the ability of the learning algorithm in dealing with various data-distributions, where smaller values of  $p$  corresponds to isolated modes, with longer mixing times for the Gibbs chain. The four basic modes and exemplary images are presented in Fig. 1.



**Fig. 1** Four basic modes (uncorrupted images) (top) and exemplary images from the toy dataset with the probability of flipping a pixel  $p = 0.01$  (middle) and  $p = 0.1$  (bottom). The probability  $p$  controls the difficulty of learning to distinguish the four basic modes





**Fig. 2** Results on toy problem with  $p = 0.01$  (left) and  $p = 0.1$  (right) using RBM with 10 hidden units using CD (red), P&GEO (green), and P&CD (blue). The average estimated test log-likelihood curves with one standard deviation over 5 repeated runs with random parameter initialization are reported. (Color figure online)

We generated 10,000 training images, 10,000 validation images and 10,000 test images. For training, we performed at most 500,000 weight updates for an RBM with 10 hidden units,

**Table 1** Average test log-likelihood with one standard deviation calculated using AIS

	Toy $p = 0.01$			Toy $p = 0.1$		
	Avg. LL	Avg. Std	Avg. Iters	Avg. LL	Avg. Std	Avg. Iters
CD-1	-4.45	1.218	48	-6.63	0.011	79
P&GEO-1	-3.70	0.697	59	-6.86	0.221	87
P&CD-1	-2.99	0.418	48	-6.81	0.233	71
CD-5	-2.43	0.053	67	-6.61	0.008	85
P&GEO-5	-2.67	0.359	42	-6.62	0.018	91
P&CD-5	-2.47	0.355	54	-6.62	0.003	86
CD-10	-2.56	0.093	80	-6.62	0.005	93
P&GEO-10	-2.36	0.089	59	-6.62	0.053	133
P&CD-10	-2.78	0.173	105	-6.62	0.004	98

and stochastic gradient descent with momentum term. We use the small number of hidden units because for 10 hidden units it is possible to calculate the exact value of the log-likelihood.

Experiments for two datasets were repeated 5 times. The averaged results with one standard deviation for  $p = 0.01$  and  $p = 0.1$  are presented in Fig. 2. A summary of final results, i.e., the average log-likelihood, the average standard deviation and the average number of iterations until convergence, are given in Table 1.

*Discussion* When the modes are close ( $p = 0.1$ ), mixing is fast and all training techniques perform similarly. However, for  $K = 1$  the PM approach converges much faster than the CD method but eventually obtains slightly worst result (see Fig. 2b). For  $K = 5$  and  $K = 10$ , the CD converges faster than the PM by about 10 and 30 iterations (epochs), however, all of the considered methods converges to the same result. In the case of  $p = 0.01$ , where good mixing is crucial, both the CD and the PM perform similarly. Nevertheless, the CD requires about 20 more iterations than P&GEO to converge (see Fig. 2c, e). Interestingly, the P&GEO seems to be more stable than the P&CD because it requires less iterations to converge and has smaller variance (see especially Fig. 2e). Moreover, for  $K = 10$ , P&CD appears to struggle with poor mixing.

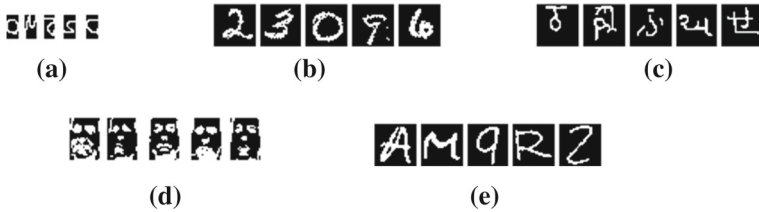
A possible explanation for unstable behavior of the P&CD is the way the update is performed: first visibles are calculated and then new hiddenes are computed for given new visibles. We presume that the algorithms "jumps" between poor (local) optima and that is why it is unable to reach a better solution. On the other hand, the greedy optimization performs an update *at once*, for both visibles and latents, and, therefore, its performance is stable and the variance is much lower.

## 4.2 Image Datasets

### 4.2.1 Unsupervised Evaluation

*Setting* In the second experiment, we evaluate the proposed approach on five image datasets, namely: Letters [31], MNIST [14], Omniglot [12], Frey Face<sup>5</sup>, and Handwritten Character Recognition (HCR) [34]. The Letters dataset contains black-and-white images of  $16 \times 10$  pixels, the MNIST dataset contains gray-scaled images of  $28 \times 28$  pixels of ten hand-written

<sup>5</sup> <http://www.cs.nyu.edu/~roweis/data.html>.



**Fig. 3** Exemplary images for the image benchmark datasets considered in the experiment. Notice that all data are binary in order to fit the binary RBM considered in this paper

digits (from 0 to 9)<sup>6</sup>, the Omniglot dataset contains black-and-white images rescaled to size  $28 \times 28$  pixels representing 1,623 handwritten characters from 50 writing systems, the Frey Face dataset contains gray-scaled images of size  $20 \times 28$  pixels representing faces<sup>7</sup>, and the HCR dataset contains black-and-white images of  $28 \times 28$  pixels of handwritten digits and characters. Each dataset is divided into fixed training set, validation set, and test set, i.e., Letters: 40,000, 5,000, 7,152, MNIST: 50,000, 10,000, 10,000, Omniglot: 19,476, 4,869, 8,115, Frey Face: 1,400, 200, 365, HCR: 24,000, 8,000, 8,134. Examples for each dataset are depicted in Fig. 3.

In the learning procedure we trained the RBM with 50 hidden units for Letters, 100 hidden units for Frey Face, and 500 hidden units for rest of datasets, and we used mini-batches of size 100. The number of hidden units were determined in such a way to match commonly used architectures in the literature [19]. Detailed results of the considered learning techniques are presented in Table 2 and wall-clock times for MNIST and Frey Face are reported in Table 3. Additionally, the results for sampling/optimization steps equal  $K = \{1, 5, 10\}$  are depicted in Fig. 4. All experiments were run 3 times with a random parameter initialization.

*Discussion* In Table 2 we see that the PM approach performs slightly better than CD. The wall-clock times of the considered methods are almost the same (see Table 3), i.e., the number of Gibbs sampler steps and the number of optimization steps take almost exactly the same amount of time, therefore we focused on the results of the same time complexity for  $K$  equal 1, 5 and 10. We found that, in three out of five datasets, the PM approach performs better than the CD method (see Fig. 4b, c, d), the CD slightly dominates in one case (see Fig. 4a) and there is one draw (see Fig. 4e). Hence, the PM approach seems to be favorable to the CD since it obtains better results within similar wall-clock time.

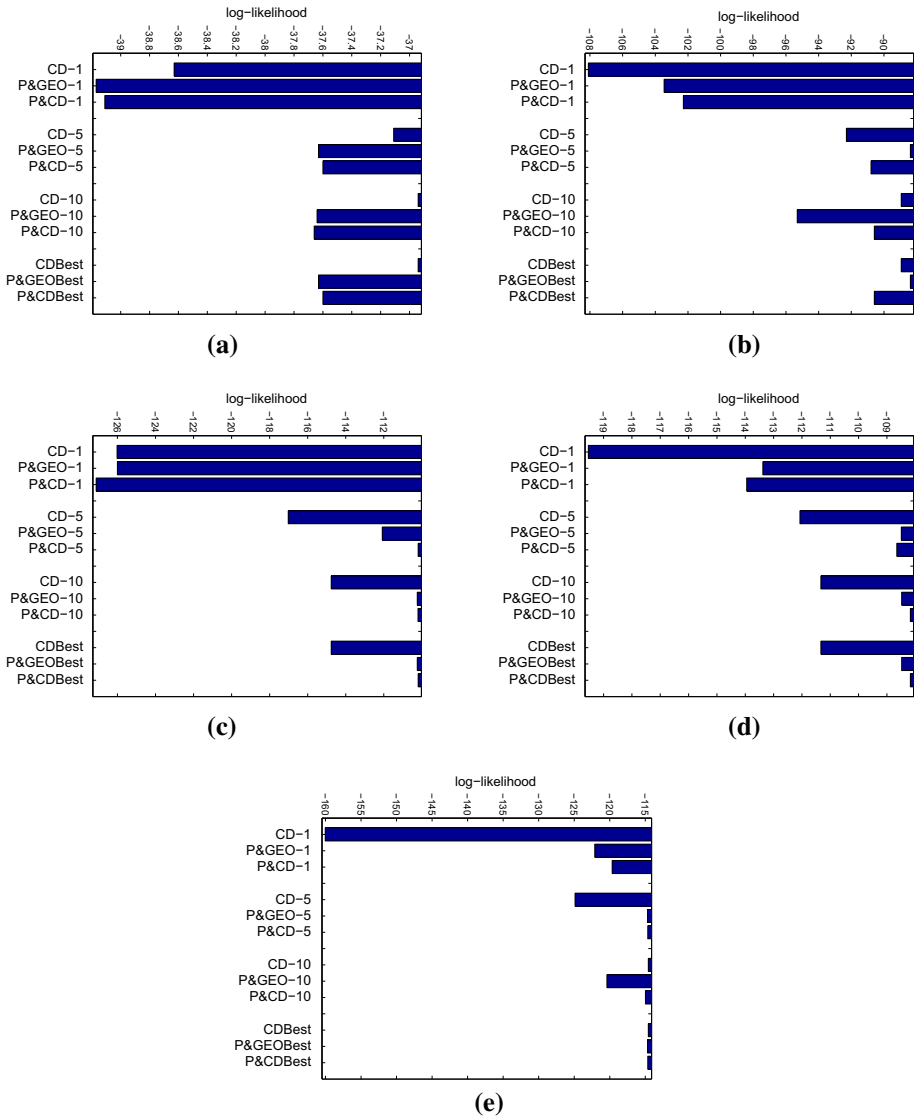
An interesting result was obtained in the case of the Omniglot dataset where CD performed worst than P&GEO and P&CD by about 5 nats. We hypothesize that the reason for this significant difference lies in the data itself. Notice that Omniglot contains images of 1623 different characters and in the training data we have only about 12 images per character. Moreover, there are 1623 modes in this data, which is drastically different from MNIST (10 modes), Letters (26 modes) or HCR (36 modes). Possibly, the PM approach better handles cases where there is less data for a mode than the CD. In other words, the PM approach may, to some extent, generate samples from a mode for which only a few examples are given, while the Gibbs sampler may "jump" between modes and so generate false samples. Similarly, in the case of Frey Face the difference in favor of the PM approach is about 4.5 nats. However, it is difficult to estimate a number of modes for Frey Face because this dataset contains faces representing various emotions that are shown from different angles. Nonetheless, we can

<sup>6</sup> The dataset was binarized according to [27].

<sup>7</sup> The dataset was binarized using a fixed threshold 0.55.

**Table 2** Average test log-likelihood with one standard deviation calculated using AIS

	LETTERS	MNIST	OMNIGLOT	FREY FACE	HCR
CD-1	-38.63 ± 0.18	-108.10 ± 0.08	-126.02 ± 0.21	-119.54 ± 1.79	-160.04 ± 1.09
CD-5	-37.11 ± 0.06	-92.30 ± 0.31	-117.02 ± 1.27	-112.07 ± 0.88	-124.88 ± 1.48
CD-10	-36.94 ± 0.05	-88.94 ± 0.09	-114.76 ± 0.36	-111.33 ± 1.99	-114.57 ± 0.74
P&GEO-1	-39.17 ± 0.25	-103.46 ± 1.80	-126.00 ± 1.66	-113.38 ± 1.40	-122.11 ± 0.60
P&GEO-5	-37.63 ± 0.01	-88.38 ± 1.48	-112.08 ± 0.82	-108.49 ± 0.24	-114.69 ± 3.52
P&GEO-10	-37.64 ± 0.13	-95.31 ± 1.30	-110.24 ± 0.76	-108.48 ± 0.64	-120.40 ± 0.91
P&CD-1	-39.11 ± 0.12	-102.28 ± 0.70	-127.12 ± 2.28	-113.95 ± 0.70	-119.66 ± 0.49
P&CD-5	-37.60 ± 0.19	-90.79 ± 0.77	-110.19 ± 0.21	-108.65 ± 0.69	-114.65 ± 1.39
P&CD-10	-37.66 ± 0.10	-90.59 ± 0.51	-110.20 ± 0.34	-108.17 ± 0.70	-114.98 ± 0.31



**Fig. 4** Results on image datasets using the approximated average test log-likelihood. The considered learning methods (CD, P&CD, P&GEO) are grouped according to number of sampling/optimization steps  $K = \{1, 5, 10\}$  that corresponds to the wall-clock time complexity. In each image bottom bars depict the best results

say that once again this dataset is characterized by a multi-modality property and this gives another evidence for our presumption. We leave thorough investigation of this hypothesis for future research.

Overall, we found that the proposed P&GEO and the P&CD perform very similarly. In fact, the P&GEO obtains better results than the P&CD only on MNIST by about 2 nats. Therefore, we cannot clearly conclude that the proposed optimization technique is indeed favorable. Nonetheless, we see two potential qualitative advantages over the coordinate descent method. First, an application of the greedy strategy to DBM is straightforward since we are interested

**Table 3** Average time in milliseconds per mini-batch for different datasets (in all cases size of images are provided)

Method $K$	CD			P&GEO			P&CD		
	1	5	10	1	5	10	1	5	10
MNIST ( $28 \times 28$ )	2.2	3.6	5.6	2.5	3.9	5.8	2.6	3.8	5.4
FREY FACE ( $20 \times 28$ )	1.1	1.7	2.6	1.6	2.0	2.7	1.5	2.0	2.6
LETTERS ( $16 \times 10$ )	0.9	1.2	1.6	0.9	1.2	1.5	1.0	1.2	1.5
TOY ( $4 \times 4$ )	0.4	0.5	0.7	0.5	0.6	0.7	0.5	0.6	0.7

The results are reported over 5 repeated runs with random parameter initialization. All reported values were within one standard deviation (i.e., there is no statistically significant difference among methods)

in calculating all variables *at once*, while the coordinate descent technique requires us to determine the order of updates, which could be cumbersome for DBMs. Second, there are other variants and similar heuristics for finding a solution for unconstrained BQP problem [21]. Here we have shown that the greedy method gives competitive results, which suggests following this line of thinking to develop new methods for the MAP step in the Perturb-and-MAP framework for learning RBMs.

#### 4.2.2 Supervised Evaluation

*Setting* In the third experiment we aim at evaluating latent representation given by  $p(\mathbf{h}|\mathbf{x})$  using labeled data. For this purpose we used two out of the considered image datasets that contain also labels, namely, MNIST and Omniglot. Notice that MNIST consists 10 labels while Omniglot possesses 1622 classes. We used exactly the same training procedure as described in Sect. 4.2.

We utilized  $k$ -nearest neighbor classifier ( $k$ -NN) with  $k \in \{1, 3, 5, 7, 9\}$  and the Euclidean metric. The application of the  $k$ -NN classifier, which is a non-parametric method, does not introduce any bias of a parametric classifier. This approach is a common practice in assessing unsupervised methods [25].

In order to compare the three considered learning algorithms we used three metrics, namely, *average precision score*, *classification accuracy* and *normalized mutual information score*. The average precision score (AvgPrec) corresponds to the area under the precision-recall curve in one-versus-all classification setting. The classification accuracy (ClassAcc) represents the total number of correctly classified instances divided by the number of all examples. Finally, the normalized mutual information score (NMI) that is a normalized version of the a measure of the similarity between two labels of the same data, which is extensively used in clustering. Obtaining a value of NMI closer to 0 results in a random assignment of an example to a class and 1 corresponds to an opposite case. In the experiment we used the implementation of the considered statistics in `scikit-learn` package<sup>8</sup>. The experiment was repeated 3 times and the best results are reported.

Additionally, for the MNIST dataset we present a two-dimensional embedding of the latent representation using t-SNE [16]. We expect to notice clusters that correspond to class labels.

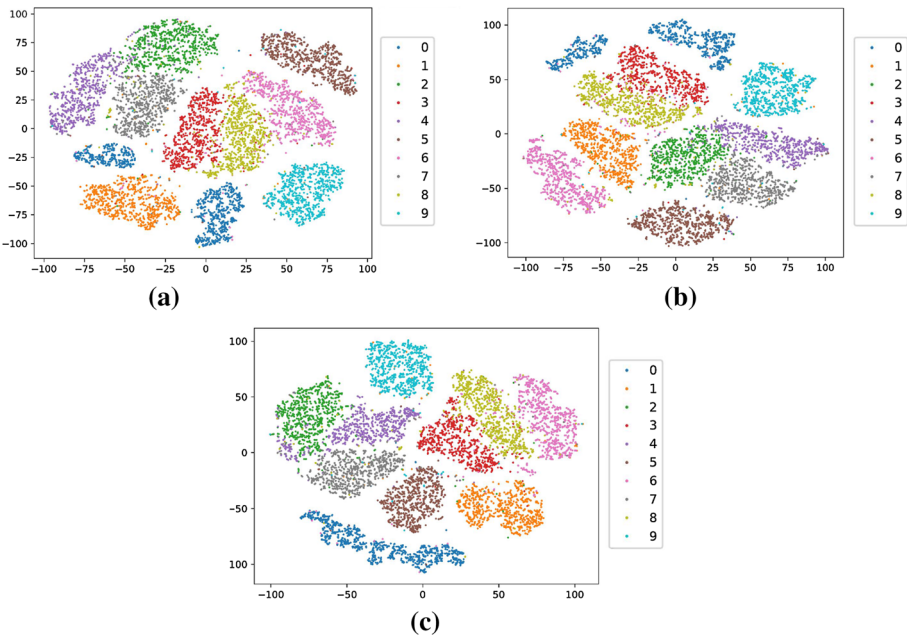
*Discussion* We present the results of the AvgPrec, ClassAcc and NMI in Table 4. The differences among the considered learning algorithms are small, however, differences are

<sup>8</sup> <http://scikit-learn.org/>.

**Table 4** The average precision score (AvgPrec), the classification accuracu (ClassAcc) and the normalized mutual information score (NMI) for the three considered learning algorithms and two datasets

Method	MNIST			OMNIGLOT		
	AvgPrec	ClassAcc	NMI	AvgPrec	ClassAcc	NMI
CD	0.476	0.969	0.921	0.052	0.203	0.794
P&GEO	0.481	0.969	0.923	0.052	0.205	0.796
P&CD	0.483	0.969	0.922	0.051	0.201	0.794

Presented results are averaged over 5 runs. All standard deviations were much smaller than 0.001 and that is why they were omitted



**Fig. 5** Two-dimensional visualization of the latent representations given by t-SNE method

significant. The P&CD methods performs slightly better than the CD on MNIST but worst on Omniglot. However, the P&GEO performs slightly better than the CD on both datasets. This result is especially apparent in terms of the NMI. Since we applied the *k*-NN classifier that does not introduce any additional burden of adaptive parameters, the achieved values indicate that the P&GEO helps to better represent data than other methods. Images

In Fig. 5 the t-SNE 2D visualizations are presented. As expected, all representations are similar and each cluster is almost ideally associated with a single class label. Moreover, we notice that some classes tend to be close to each other, e.g., 3s with 8s and 6s that are depicted by red, yellow and pink color, respectively, in Fig. 5. We notice, however, that for the CD and the P&CD zeros denoted by the blue color in Fig. 5 are grouped in two separate clusters while for the P&GEO they form one cluster. This result may indicate that the P&GEO indeed produces good latent representation. Nevertheless, this result may follow from limiting capabilities of the t-SNE method and in high dimensions the latent representation are grouped properly.

**Table 5** Average test log-likelihood with one standard deviation calculated using AIS on 20Newsgroups

	$K = 1$	$K = 5$	$K = 10$
CD	$-13.72 \pm 0.02$	$-13.73 \pm 0.03$	$-13.70 \pm 0.02$
P&GEO	$-18.19 \pm 0.19$	$-13.70 \pm 0.01$	$-13.72 \pm 0.02$
P&CD	$-18.38 \pm 0.18$	$-13.71 \pm 0.02$	$-13.72 \pm 0.02$

**Table 6** The average precision score (AvgPrec), the classification accuracy (ClassAcc) and the normalized mutual information score (NMI) for the three considered learning algorithms on 20Newsgroups

Method	AvgPrec	ClassAcc	NMI
CD	0.497	0.799	0.485
P&GEO	0.503	0.801	0.490
P&CD	0.502	0.801	0.491

Presented results are averaged over 5 runs. All standard deviations were much smaller than 0.001 and that is why are omitted

## 4.3 Text Dataset

### 4.3.1 Unsupervised Evaluation

*Setting* In this experiment, we evaluate the proposed approach on the 20-newsgroups dataset [13], 20Newsgroups<sup>9</sup> for short, that contains 8,500 training, 1245 validation, and 6,497 test text documents of blog posts. Each text document is represented as a binary vector of 100 most frequent words among all documents. The problem is the text analysis and the document classification to one of four newsgroup meta-topics (classes).

In the learning procedure we trained the RBM with 50 hidden units, and we used mini-batches of size 100. We used sampling/optimization steps equal  $K = \{1, 5, 10\}$ . Detailed results of the considered learning techniques are presented in Table 5. All experiments were run 3 times with a random parameter initialization and kept fixed across all methods.

*Discussion* In Table 5 we notice that there is no significant difference between the CD and the Perturb-and-MAP approach. However, interestingly for the CD method there was almost no difference in the generative performance with respect to a varying number sampling steps while for the PM method it mattered a lot. Nevertheless, in this experiments the CD and the PM performed on a par.

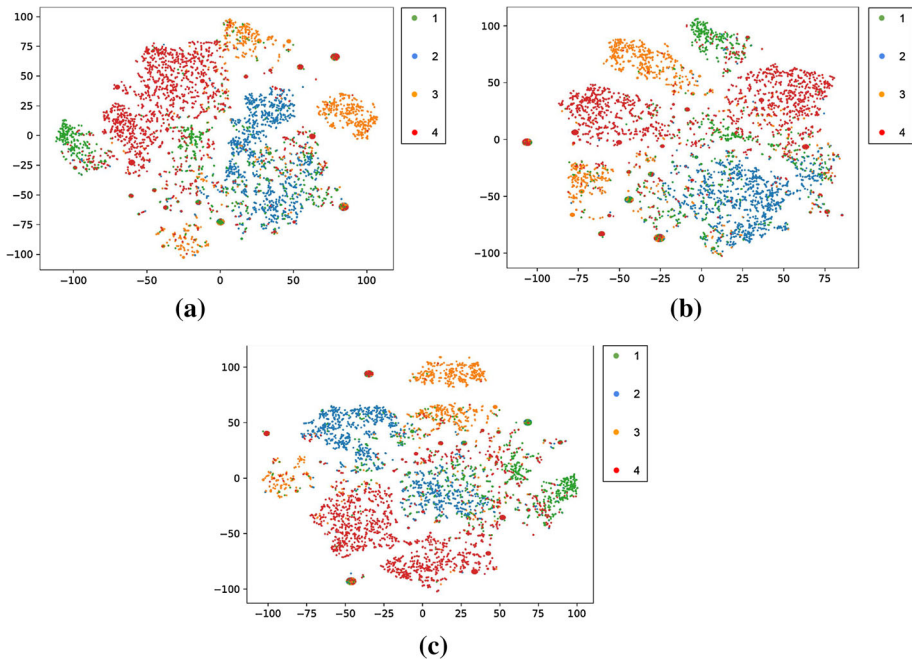
### 4.3.2 Supervised Evaluation

*Setting* Similarly to the image datasets, we aim at evaluating latent representation given by  $p(\mathbf{h}|\mathbf{x})$  using labeled data. Again, we used  $k$ -nearest neighbor classifier ( $k$ -NN) with  $k \in \{1, 3, 5, 7, 9\}$  and the Euclidean metric. In order to evaluate learned latent representation we used the average precision (AvgPrec), the classification accuracy (ClassAcc) and the normalized mutual information score (NMI). Eventually, we performed t-SNE to obtain two-dimensional embedding for visualizing latent representations.

*Discussion* Contrary to the unsupervised evaluation, we notice that the PM approach allows to obtain better values of all performance metrics than the CD method, see Table 6. Moreover, the P&GEO method slightly outperforms P&CD in terms of the average precision but achieves

<sup>9</sup> <http://www.cs.nyu.edu/~roweis/data.html>.





**Fig. 6** Two-dimensional visualization of the latent representations given by t-SNE method

a slightly worse result in terms of the normalized mutual information. These results suggest that the PM method allows to train better discriminative representation while maintaining similar generative capabilities as the CD method.

In Fig. 6 we present a two-dimensional visualization of the latent representations using t-SNE. The CD method tend to learn a representation that in some regions seems to be hard to assign a single class label. There are consistent clusters of a single class, however, in many regions classes are completely mixed. The PM approach, on the other hand, tends to group objects of the same class into more coherent clusters.

We also notice that there is an interesting effect of aggregating many latent representations in very small regions, see Fig. 6. A possible explanation for this phenomenon is that multiple documents are expressed by the same or almost the same sparse binary vectors. Therefore, they start formulating extremely dense clusters of points.

### 5 Conclusion

The current work focuses on the low-dimensional perturbations in the *Perturb Step*. Although our results indicate that this approach is efficient, it is worth considering higher-order perturbations. Naturally, while computational complexity grows exponentially with the order of the perturbations, this future direction might lead to much better results. We are currently investigating other optimization methods to find the ground state of the perturbed energy that may fit to the considered framework for learning RBMs. In the near future we would like to utilize our approach to deep models, where the much more complex form of the energy function poses a greater challenge.

In this paper, we introduced a novel application of the Perturb-and-MAP approach in the context of learning parameters of RBM. Since theoretical considerations indicate that samples obtained within the low-dimensional PM framework are approximate samples from the RBM (see Theorem 2), our method works by perturbing unary potentials (i.e., bias terms) and further finding configurations of visible variables that minimize the perturbed energy. For this purpose we proposed a greedy optimization technique to find MAP solutions of the perturbed energy. Using both toy datasets, real image benchmark datasets and a text analysis dataset, we showed empirically that our method is competitive to the well-known Contrastive Divergence algorithm. Moreover, we indicated some advantages of the proposed greedy method over the coordinate descent.

**Acknowledgements** The work conducted by J.M. Tomczak was partially co-financed within the *Project: Young Staff 2015*.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

- Bertasius G, Liu Q, Torresani L, Shi J (2016) Local Perturb-and-MAP for structured prediction. arXiv preprint [arXiv:1605.07686](https://arxiv.org/abs/1605.07686)
- Desjardins G, Courville A, Bengio Y, Vincent P, Delalleau O (2010) Tempered markov chain monte carlo for training of restricted boltzmann machines. In: AISTATS, pp 145–152
- Gane A, Hazan T, Jaakkola T (2014) Learning with maximum a-posteriori perturbation models. In: AISTATS, pp 247–256
- Gumbel E (1954) Statistical theory of extreme values and some practical applications: a series of lectures. Applied mathematics series. U. S. Govt. Print Office, Tinker Air Force Base
- Hazan T, Jaakkola T (2012) On the partition function and random maximum a-posteriori perturbations. In: ICML
- Hazan T, Maji S, Jaakkola T (2013) On sampling from the gibbs distribution with random maximum a-posteriori perturbations. In: NIPS, pp 1268–1276
- Hinton G (2002) Training products of experts by minimizing contrastive divergence. *Neural Comput* 14(8):1771–1800
- Kappes JH, Swoboda P, Savchynskyy B, Hazan T, Schnörr C (2016) Multicuts and perturb & MAP for probabilistic graph clustering. *J Math Imaging Vis* 56:1–17
- Kochenberger G, Hao J-K, Glover F, Lewis M, Lü Z, Wang H, Wang Y (2014) The unconstrained binary quadratic programming problem: a survey. *J Comb Optim* 28(1):58–81
- Koller D, Friedman N (2009) Probabilistic graphical models: principles and techniques. MIT Press, Cambridge
- Kolmogorov V, Zabini R (2004) What energy functions can be minimized via graph cuts? *IEEE Trans Pattern Anal Mach Intell* 26(2):147–159
- Lake BM, Salakhutdinov R, Tenenbaum JB (2015) Human-level concept learning through probabilistic program induction. *Science* 350(6266):1332–1338
- Lang K (1995) Newsweeder: learning to filter netnews. In: Machine learning proceedings, pp 331–339
- LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324
- Li K, Swersky K, Zemel R (2013) Efficient feature learning using Perturb-and-MAP. In: NIPS 2013 workshop on perturbations
- Lvd Maaten, Hinton G (2008) Visualizing data using t-SNE. *J Mach Learn Res* 9:2579–2605
- Maddison C, Tarlow D, Minka T (2014) A\* Sampling. In: NIPS, pp 3086–3094
- Maji S, Hazan T, Jaakkola T (2014) Active boundary annotation using random map perturbations. In: AISTATS, pp 604–613
- Marlin B, Swersky K, Chen B, Freitas N.De (2010) Inductive principles for restricted boltzmann machine learning. In: AISTATS, pp 509–516

20. Meier R, Knecht U, Jungo A, Wiest R, Reyes M (2017) Perturb-and-MPM: quantifying segmentation uncertainty in dense multi-label CRFs. arXiv preprint [arXiv:1703.00312](https://arxiv.org/abs/1703.00312)
21. Merz P, Freisleben B (2002) Greedy and local search heuristics for unconstrained binary quadratic programming. *J Heuristics* 8(2):197–213
22. Nadarajah S (2008) Exact distribution of the linear combination of  $p$  Gumbel random variables. *Int J Comput Math* 85(9):1355–1362
23. Papandreou G, Yuille A (2011) Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In: ICCV, pp 193–200
24. Papandreou G, Yuille A (2014) Perturb-and-MAP random fields: reducing random sampling to optimization, with applications in computer vision. In: Nowozin S, Gehler P, Jancsary J, Lampert C (eds) *Advanced structured prediction*. MIT Press, Cambridge, pp 159–185
25. Radford A, Metz L, Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint [arXiv:1511.06434](https://arxiv.org/abs/1511.06434)
26. Ravanbakhsh S, Greiner R, Frey B (2014) Training restricted boltzmann machine by perturbation. arXiv preprint [arXiv:1405.1436](https://arxiv.org/abs/1405.1436)
27. Salakhutdinov R, Murray I (2008) On the quantitative analysis of deep belief networks. In: ICML, pp 872–879
28. Shimony S (1994) Finding MAPs for belief networks is NP-hard. *Artif Intell* 68(2):399–410
29. Smolensky P (1986) Information processing in dynamical systems: foundations of harmony theory. In: Rumelhart DE, McClelland JL (eds) *Parallel Distributed processing: explorations in the microstructure of cognition*, vol. 1, pp 194–281. MIT Press
30. Tarlow D, Adams R, Zemel R (2012) Randomized optimum models for structured prediction. In: AIS-TATS, pp 1221–1229
31. Taskar B, Guestrin C, Koller D (2003) Max-margin Markov networks. *Adv Neural Inf Process Syst* 16:25
32. Tieleman T (2008) Training restricted Boltzmann machines using approximations to the likelihood gradient. In: ICML, pp 1064–1071
33. Tomczak J (2016) On some properties of the low-dimensional Gumbel perturbations in the Perturb-and-MAP model. *Stat Probab Lett*. <https://doi.org/10.1016/j.spl.2016.03.019>
34. Van der Maaten L (2009) A new benchmark dataset for handwritten character recognition. Tilburg University, Tilburg, pp 2–5

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.