On Contrastive Divergence Learning

Miguel Á. Carreira-Perpiñán[*] **Geoffrey E. Hinton**
Dept. of Computer Science, University of Toronto
6 King's College Road. Toronto, ON M5S 3H5, Canada
Email: {miguel,hinton}@cs.toronto.edu

Abstract

Maximum-likelihood (ML) learning of Markov random fields is challenging because it requires estimates of averages that have an exponential number of terms. Markov chain Monte Carlo methods typically take a long time to converge on unbiased estimates, but Hinton (2002) showed that if the Markov chain is only run for a few steps, the learning can still work well and it approximately minimizes a different function called "contrastive divergence" (CD). CD learning has been successfully applied to various types of random fields. Here, we study the properties of CD learning and show that it provides biased estimates in general, but that the bias is typically very small. Fast CD learning can therefore be used to get close to an ML solution and slow ML learning can then be used to fine-tune the CD solution.

Consider a probability distribution over a vector $\mathbf{x}$ (assumed discrete w.l.o.g.) and with parameters $\mathbf{W}$

$$p(\mathbf{x};\mathbf{W}) = \frac{1}{Z(\mathbf{W})} e^{-E(\mathbf{x};\mathbf{W})} \tag{1}$$

where $Z(\mathbf{W}) = \sum_{\mathbf{x}} e^{-E(\mathbf{x};\mathbf{W})}$ is a normalisation constant and $E(\mathbf{x};\mathbf{W})$ is an energy function. This class of random-field distributions has found many practical applications (Li, 2001; Winkler, 2002; Teh et al., 2003; He et al., 2004). Maximum-likelihood (ML) learning of the parameters $\mathbf{W}$ given an iid sample $\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$ can be done by gradient ascent:

$$\mathbf{W}^{(\tau+1)} = \mathbf{W}^{(\tau)} + \eta \left.\frac{\partial L(\mathbf{W};\mathcal{X})}{\partial \mathbf{W}}\right|_{\mathbf{W}^{(\tau)}}$$

where the learning rate η need not be constant. The average log-likelihood is:

$$\begin{aligned} L(\mathbf{W};\mathcal{X}) &= \tfrac{1}{N}\textstyle\sum_{n=1}^N \log p(\mathbf{x}_n;\mathbf{W}) = \langle \log p(\mathbf{x};\mathbf{W})\rangle_0 \\ &= -\langle E(\mathbf{x};\mathbf{W})\rangle_0 - \log Z(\mathbf{W}) \end{aligned}$$

where $\langle\cdot\rangle_0$ denotes an average w.r.t. the data distribution $p_0(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^N \delta(\mathbf{x}-\mathbf{x}_n)$. A well-known difficulty arises in the computation of the gradient

$$\frac{\partial L(\mathbf{W};\mathcal{X})}{\partial \mathbf{W}} = -\left\langle \frac{\partial E(\mathbf{x};\mathbf{W})}{\partial \mathbf{W}} \right\rangle_0 + \left\langle \frac{\partial E(\mathbf{x};\mathbf{W})}{\partial \mathbf{W}} \right\rangle_\infty$$

where $\langle\cdot\rangle_\infty$ denotes an average with respect to the model distribution $p_\infty(\mathbf{x};\mathbf{W}) = p(\mathbf{x};\mathbf{W})$. The average $\langle\cdot\rangle_0$ is readily computed using the sample data $\mathcal{X}$, but the average $\langle\cdot\rangle_\infty$ involves the normalisation constant $Z(\mathbf{W})$, which cannot generally be computed efficiently (being a sum of an exponential number of terms). The standard approach is to approximate the average over the distribution with an average over a sample from $p(\mathbf{x};\mathbf{W})$, obtained by setting up a Markov chain that converges to $p(\mathbf{x};\mathbf{W})$ and running the chain to equilibrium (for reviews, see Neal, 1993; Gilks et al., 1996). This Markov chain Monte Carlo (MCMC) approach has the advantage of being readily applicable to many classes of distribution $p(\mathbf{x};\mathbf{W})$. However, it is typically very slow, since running the Markov chain to equilibrium can require a very large number of steps, and no foolproof method exists to determine whether equilibrium has been reached. A further disadvantage is the large variance of the estimated gradient.

To avoid the difficulty in computing the log-likelihood gradient, Hinton (2002) proposed the contrastive divergence (CD) method which approximately follows the gradient of a different function. ML learning minimises the Kullback-Leibler divergence

$$\mathrm{KL}\left(p_0 \| p_\infty\right) = \sum_{\mathbf{x}} p_0(\mathbf{x}) \log \frac{p_0(\mathbf{x})}{p(\mathbf{x};\mathbf{W})}.$$

CD learning approximately follows the gradient of the

[*] Current address: Dept. of Computer Science & Electrical Eng., OGI School of Science & Engineering, Oregon Health & Science University. Email: miguel@cse.ogi.edu.

difference of two divergences (Hinton, 2002):

$$\mathrm{CD}_n = \mathrm{KL}\left(p_0 \| p_\infty\right) - \mathrm{KL}\left(p_n \| p_\infty\right).$$

In CD learning, we start the Markov chain at the data distribution p_0 and run the chain for a small number n of steps (e.g. $n = 1$). This greatly reduces both the computation per gradient step and the variance of the estimated gradient, and experiments show that it results in good parameter estimates (Hinton, 2002). CD has been applied effectively to various problems (Chen and Murray, 2003; Teh et al., 2003; He et al., 2004), using Gibbs sampling or hybrid Monte Carlo as the transition operator for the Markov chain. However, it is hard to know how good the parameter estimates really are, since no comparison was done with the real ML estimates (which are impractical to compute). There has been a little theoretical investigation of the properties of contrastive divergence (MacKay, 2001; Williams and Agakov, 2002; Yuille, 2004), but important questions remain unanswered: Does it converge? If so how fast, and how are its convergence points related to the true ML estimates?

In this paper we provide some theoretical and empirical evidence that contrastive divergence can, in fact, be the basis for a very effective approach for learning random fields. We concentrate on Boltzmann machines, though our results should be more generally valid. First, we show that CD provides biased estimates in general: for almost all data distributions, the fixed points of CD are not fixed points of ML, and vice versa (section 2). We then show, by comparing CD and ML in empirical tests, that this bias is small (sections 3–4) and that an effective approach is to use CD to perform most of the learning followed by a short run of ML to clean up the solution (section 5).

To eliminate sampling noise from our investigations, we use fairly small models (with e.g. 48 parameters) for which we can compute the exact model distribution and the exact distribution at each step of the Markov chain at each stage of learning. Throughout, we take ML learning to mean exact ML learning (i.e., with $n \to \infty$ in the Markov chain) and CD_n learning to mean learning using the exact distribution of the Markov chain after n steps. The sampling noise in real MCMC estimates would create an additional large advantage that favours CD over ML learning, because CD has much lower variance in its gradient estimates.

1 ML and CD learning for two types of Boltzmann machine

We will concentrate on two types of Boltzmann machine, which are a particular case of the model of eq. (1). In Boltzmann machines, there are v visible units $\mathbf{x} = (x_1, \dots, x_v)^T$ that encode a data vector, and h hidden units $\mathbf{y} = (y_1, \dots, y_h)^T$; all units are binary and take values in $\{0, 1\}$.

Fully visible Boltzmann machines have $h = 0$ and the visible units are connected to each other. The energy is then $E(\mathbf{x}; \mathbf{W}) = -\frac{1}{2}\mathbf{x}^T\mathbf{W}\mathbf{x}$, where $\mathbf{W} = (w_{ij})$ is a symmetric $v \times v$ matrix of real-valued weights (for simplicity, we do not consider biases). We denote such a machine by $\mathrm{VBM}(v)$. In VBMs, the log-likelihood has a unique optimum because its Hessian is negative definite. Since $\partial E/\partial w_{ij} = -x_i x_j$, ML learning takes the form

$$w_{ij}^{(\tau+1)} = w_{ij}^{(\tau)} + \eta\left(\langle x_i x_j\rangle_0 - \langle x_i x_j\rangle_\infty\right)$$

while CD_n learning takes the form

$$w_{ij}^{(\tau+1)} = w_{ij}^{(\tau)} + \eta\left(\langle x_i x_j\rangle_0 - \langle x_i x_j\rangle_n\right).$$

Here, $p_n(\mathbf{x}; \mathbf{W}) = \mathbf{p}_n = \mathbf{T}^n\mathbf{p}_0$ is the nth-step distribution of the Markov chain with transition matrix[1] $\mathbf{T}$ started at the data distribution $\mathbf{p}_0$.

Restricted Boltzmann machines (Smolensky, 1986; Freund and Haussler, 1992) have connections only between a hidden and a visible unit, i.e., they form a bipartite graph. The energy is then $E(\mathbf{x}, \mathbf{y}; \mathbf{W}) = -\mathbf{y}^T\mathbf{W}\mathbf{x}$, where we have v visible units and h hidden units, and $\mathbf{W} = (w_{ij})$ is an $h \times v$ matrix of real-valued weights. We denote such a machine by $\mathrm{RBM}(v, h)$. By making h large, an RBM can be given far more representational power than a VBM, but the log-likelihood can have multiple maxima. The learning is simpler than in a general Boltzmann machine because the visible units are conditionally independent given the hidden units, and the hidden units are conditionally independent given the visible units. One step of Gibbs sampling can therefore be carried out in two half-steps: the first updates all the hidden units and the second updates all the visible units. Equivalently, we can write $\mathbf{T} = \mathbf{T}_\mathbf{x}\mathbf{T}_\mathbf{y}$ where $t_{\mathbf{x};j,i} = p(\mathbf{x} = j|\mathbf{y} = i; \mathbf{W})$ and $t_{\mathbf{y};i,j} = p(\mathbf{y} = i|\mathbf{x} = j; \mathbf{W})$.

Since $\partial E/\partial w_{ij} = -y_i x_j$, ML learning takes the form

$$w_{ij}^{(\tau+1)} = w_{ij}^{(\tau)} + \eta\left(\left\langle\langle y_i x_j\rangle_{p(\mathbf{y}|\mathbf{x};\mathbf{W})}\right\rangle_0 - \langle y_i x_j\rangle_\infty\right)$$

while CD_n learning takes the form

$$w_{ij}^{(\tau+1)} = w_{ij}^{(\tau)} + \eta\left(\left\langle\langle y_i x_j\rangle_{p(\mathbf{y}|\mathbf{x};\mathbf{W})}\right\rangle_0 - \langle y_i x_j\rangle_n\right).$$

2 Analysis of the fixed points

A probability distribution over v units is a vector of 2^v real-valued components (from 0 to $2^v - 1$ in bi-

[1] Here and elsewhere we omit the dependence of $\mathbf{T}$ on the parameter values $\mathbf{W}$ to simplify the notation.

nary notation) that lives in the 2^v–dimensional simplex $\Delta_{2^v} = \{\mathbf{x} \in \mathbb{R}^{2^v} :\ x_i \geq 0,\ \sum_{i=1}^{2^v} x_i = 1\}$. Each coordinate axis of $\mathbb{R}^{2^v}$ corresponds to a state (binary vector). We write such a distribution as $p(\cdot)$, when emphasising that it is a function, or as a vector $\mathbf{p}$, when emphasising that it is a point in the simplex. We define a Markov chain through its transition operator, which is a stochastic $2^v \times 2^v$ matrix $\mathbf{T}$. We use the Gibbs sampler as the transition operator because of its simplicity and its wide applicability to many distributions. For Boltzmann machines with finite weights, the Gibbs sampler converges to a stationary distribution which is the model distribution $p(\mathbf{x}; \mathbf{W})$. For an initial distribution $\mathbf{p}$ we then have $\mathbf{p}_n = \mathbf{T}^n \mathbf{p}$ for $n = 1, 2, \ldots$; and $p(\mathbf{x}; \mathbf{W}) = \mathbf{p}_\infty = \mathbf{T}^\infty \mathbf{p}$ for any $\mathbf{p} \in \Delta_{2^v}$, i.e., $\mathbf{T}^\infty = \mathbf{p}_\infty \mathbf{1}^T$ where $\mathbf{1}$ is the vector of ones.

VBMs or RBMs define a manifold $\mathcal{M}$ within the simplex, parameterised by $\mathbf{W}$, with $w_{ij} \in \mathbb{R}$ (we ignore the case of infinite weights, corresponding to distributions in the intersection of $\mathcal{M}$ and the simplex boundary). The learning (ML or CD) starts at a point in $\mathcal{M}$ and follows the (approximate) gradient of the log-likelihood, thus tracing a trajectory within $\mathcal{M}$.

For ML with gradient learning, the fixed points are the zero-gradient points (maxima, minima and saddles), which satisfy $\langle \mathbf{g} \rangle_0 = \langle \mathbf{g} \rangle_\infty$ where $\mathbf{g} = \partial E / \partial \mathbf{W}$. For n-step CD, the fixed points satisfy $\langle \mathbf{g} \rangle_0 = \langle \mathbf{g} \rangle_n$. In this section we address the theoretical question of whether the fixed points of ML are fixed points of CD and vice versa. We show that, in general,

$$\begin{aligned} &\exists \mathbf{p}_0 :\ \langle \mathbf{g} \rangle_0 = \langle \mathbf{g} \rangle_\infty \neq \langle \mathbf{g} \rangle_1 \\ &\exists \mathbf{p}_0 :\ \langle \mathbf{g} \rangle_0 = \langle \mathbf{g} \rangle_1 \neq \langle \mathbf{g} \rangle_\infty . \end{aligned}$$

We give a brief explanation of a framework for analysing the fixed points of ML and CD; full details appear in Carreira-Perpiñán and Hinton (2004). The idea is to fix a value of the weights and so a value of the moments (defined below), determine which data distributions $\mathbf{p}_0$ have such moments (i.e., the opposite of the learning problem) and then determine under what conditions ML and CD agree over those distributions. Call $\mathbf{G}$ the $|\mathbf{W}| \times 2^v$ matrix of energy derivatives, defined by

$$G_{i\mathbf{x}} = -\frac{\partial E}{\partial w_i}(\mathbf{x}; \mathbf{W})$$

where we consider $\mathbf{W}$ as a column vector with $|\mathbf{W}|$ elements and the state $\mathbf{x}$ takes values $0, 1, \ldots, 2^v - 1$ in the case of v binary variables. We can then write the moments $\mathbf{s} = \left\langle -\frac{\partial E}{\partial \mathbf{W}} \right\rangle_{\mathbf{p}} = -\langle \mathbf{g} \rangle_{\mathbf{p}}$ of a distribution $\mathbf{p}$ as $\mathbf{s} = \mathbf{G}\mathbf{p}$, i.e., $\mathbf{s}$ is a linear function of $\mathbf{p}$. Call $\mathbf{T}$ the transition matrix for the sampling operator with stationary distribution $\mathbf{p}_\infty$ (so we have $\mathbf{p}_\infty = \mathbf{T}\mathbf{p}_\infty$). In general, both $\mathbf{G}$ and $\mathbf{T}$ are functions of $\mathbf{W}$.

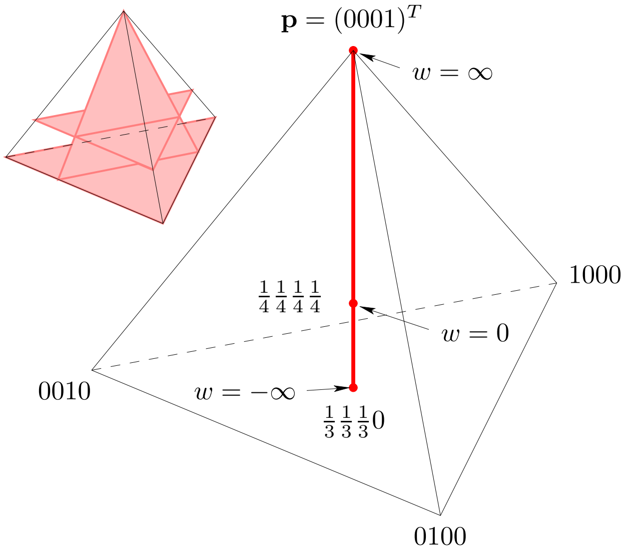
$\mathbf{p} = (0001)^T$
$w = \infty$
1000
$\frac{1}{4}\frac{1}{4}\frac{1}{4}\frac{1}{4}$
$w = 0$
0010
$w = -\infty$
$\frac{1}{3}\frac{1}{3}\frac{1}{3}0$
0100

Figure 1: The simplex in 4 dimensions of the VBM(2). The model has a single parameter $\mathbf{W} = w \in (-\infty, \infty)$. The tetrahedron represents the simplex, i.e., the set $\{\mathbf{p} \in \mathbb{R}^4 :\ \mathbf{1}^T\mathbf{p} = 1,\ \mathbf{p} \geq 0\}$. The tetrahedron corners correspond to the pure states, i.e., the distributions that assign all the probability to a single state. The red vertical segment is the manifold of VBMs, i.e., the distributions reachable by a VBM(2) for $w \in (-\infty, \infty)$. The ML estimate of a data distribution is its orthogonal projection on the model manifold. The CD estimate only agrees with the ML one for data distributions in the 3 shaded planes in the inset.

Now consider a fixed value of $\mathbf{W}$ and let $\mathbf{p}_\infty$ be its associated model distribution. Thus its moments are $\mathbf{s}_\infty = \mathbf{G}\mathbf{p}_\infty$. We define two sets $\mathcal{P}_0$ and $\mathcal{P}_1$ that depend on $\mathbf{s}_\infty$. First, the set of data distributions $\mathbf{p}_0$ that have those same moments $\mathbf{s}_\infty$ is:

$$\mathcal{P}_0 = \left\{ \mathbf{p}_0 \in \mathbb{R}^{2^v} :\quad \begin{array}{c} \mathbf{G}\mathbf{p}_0 = \mathbf{s}_\infty \\ \mathbf{1}^T\mathbf{p}_0 = 1 \\ \mathbf{p}_0 \geq 0 \end{array} \right\}.$$

Each distribution in $\mathcal{P}_0$ gives a fixed point of ML. Likewise, define the set of data distributions $\mathbf{p}_0$ whose distribution $\mathbf{p}_1 = \mathbf{T}\mathbf{p}_0$ (first step in the Markov chain, i.e., what CD$_1$ uses instead of $\mathbf{p}_\infty$) has the same moments as $\mathbf{s}_\infty$:

$$\mathcal{P}_1 = \left\{ \mathbf{p}_0 \in \mathbb{R}^{2^v} :\quad \begin{array}{c} \mathbf{G}\mathbf{T}\mathbf{p}_0 = \mathbf{s}_\infty \\ \mathbf{1}^T\mathbf{p}_0 = 1 \\ \mathbf{p}_0 \geq 0 \end{array} \right\}.$$

Both $\mathcal{P}_0$ and $\mathcal{P}_1$ are nonempty since $\mathbf{p}_\infty$ is in both. Now we can reformulate the problem in terms of the sets $\mathcal{P}_0$ and $\mathcal{P}_1$. For example, a distribution with $\mathbf{p}_0 \in \mathcal{P}_0$ and $\mathbf{p}_0 \notin \mathcal{P}_1$ satisfies $\langle \mathbf{g} \rangle_0 = \langle \mathbf{g} \rangle_\infty \neq \langle \mathbf{g} \rangle_1$ and thus gives a fixed point of ML but not of CD (that is, the CD learning rule would move away from such a $\mathbf{p}_\infty$).

In general (and ignoring technical details regarding the inequality $\mathbf{p}_0 \geq 0$), $\mathcal{P}_0$ and $\mathcal{P}_1$ are linear subspaces of the same dimension because $\mathbf{G}$ is full rank (the moments are l.i.) and $\mathbf{T}$ is generally full rank. Thus we cannot *generally* expect $\mathcal{P}_0 = \mathcal{P}_1$, so points with CD bias are the rule; points in $\mathcal{P}_0 \cap \mathcal{P}_1$ have no CD bias but are the exception (the intersection being a lower-dimensional subspace). We can make the statement precise for a given model. For example, for VBM(2) with Gibbs sampling we have $\mathbf{G} = (0\ 0\ 0\ 1)$ ($\mathbf{G}$ happens to be independent of $\mathbf{W}$ for VBMs), we can compute $\mathbf{T}$ and we can work out the set $\mathcal{P}_0 \cap \mathcal{P}_1$ for every $\mathbf{s}_\infty$ value. The resulting set, which contains all the data distributions for which ML and CD have the same fixed points (i.e., no bias), is the union (intersected with the simplex) of the 3 planes: $p_{11} = 0$; $p_{11} = \frac{1}{4}$; $3p_{01} + p_{11} = 1$, where we write a distribution as a 4-dimensional vector $\mathbf{p} = (p_{00}\ p_{01}\ p_{10}\ p_{11})^T$, corresponding to the probabilities of the 4 states $00, \dots, 11$ (see fig. 1). This set has measure zero in the simplex, so CD is biased for almost every data distribution.

A reachable distribution $\mathbf{p}_0 \in \mathcal{M}$ is a fixed point for both CD and ML, as it is invariant under $\mathbf{T}$ (Hinton, 2002). This is consistent with the above argument, as $\mathbf{p}_0 = \mathbf{p}_\infty \in \mathcal{P}_0 \cap \mathcal{P}_1$. The distributions of practical interest are typically unreachable because real data are nearly always more complicated than our computationally tractable model of it.

In summary, we expect that for almost every data distribution $\mathbf{p}_0$, the fixed points of ML are not fixed points of CD and vice versa. This means that, in general, CD is a biased learning algorithm. Our argument can be applied to models other than Boltzmann machines, transition operators other than Gibbs sampling, and to $n > 1$ (writing $\mathbf{T}^n$ instead of $\mathbf{T}$). What determines whether CD is biased are the hyperplanes defined by the matrices $\mathbf{G}$ and $\mathbf{GT}$. However, nontrivial models (i.e., defining a lower-dimensional manifold) may exist for which CD is not biased; an example is Gaussian Boltzmann machines (Williams and Agakov, 2002) and Gaussian distributions, at least in 2D (Carreira-Perpiñán and Hinton, 2004).

This analysis does not imply that CD learning converges (to a stable fixed point); at present, we do not have a proof for this. But if CD does converge, as it appears to in practice and in all our experiments, it can only converge to a fixed point. Naturally, ML does converge to its stable fixed points (maxima) from almost everywhere, since it follows the exact gradient of an objective function; in the noisy sampling case that is used in practice, it also converges provided the learning rate η follows a Robbins-Monro schedule (Benveniste et al., 1990), since the rule performs stochastic gradient learning.

3 Experiments with fully visible BMs

Since CD is biased with respect to ML for almost all data distributions, we now investigate empirically the magnitude of the bias. In all experiments in the paper, ML and CD are tested under exactly the same conditions (unless otherwise stated). Both ML and CD learning use the same initial weight vectors, the same constant learning rate $\eta = 0.9$ and the same maximum of 10 000 iterations (which is rarely reached for VBMs), stopping when $\|\mathbf{e}\|_\infty < 10^{-7}$ (where $\mathbf{e} = \langle x_i x_j \rangle_0 - \langle x_i x_j \rangle_\infty$ is the gradient vector for ML, and $\mathbf{e} = \langle x_i x_j \rangle_0 - \langle x_i x_j \rangle_1$ is the approximate gradient for CD$_1$). All the experiments use $n = 1$ step of Gibbs sampling with fixed ordering of the variables for CD learning, because this should produce the greatest bias (since CD $\xrightarrow[n\to\infty]{}$ ML). Although each of our simulated models is necessarily small, our empirical results hold for a range of model sizes and conditions, which suggests they may be more generally valid.

In this section we consider fully visible Boltzmann machines, denoted VBM(v), which have a single ML optimum. It appears that CD has a single convergence point too: for $v = 2$ we can prove this, and for $v \in \{3, \dots, 10\}$ we checked empirically by running CD from many different initial weight vectors that it always converged to the same point (up to a small numerical error). Thus, we assume that CD has a unique convergence point for VBM(v). This allows us to characterise the bias for this model class by sampling many data distributions and computing the convergence point of ML and of CD.

For a given value of v we sampled a number (as large as computationally feasible) of data distributions uniformly distributed in the simplex in 2^v variables (see Carreira-Perpiñán and Hinton, 2004 for details of how to generate these samples). Then we ran ML and CD starting with $\mathbf{W} = \mathbf{0}$ because small weights give faster convergence on average. The results for experiments for $v \in \{2, \dots, 10\}$ were qualitatively similar. For $v = 2$ it was feasible to sample 10^4 data distributions and the results are summarised in figures 2–5.

The histograms in figs. 2–3 show the bias is very small for most distributions. Fig. 4 shows that the KL error (for both ML and CD) is small for data distributions near the simplex centre. For less vague data distributions there is more variability, with some distributions having a low error and some having a much higher one. Generally speaking, the distributions having the highest KL error for ML (i.e., the distributions that are modelled worst by the VBM) are also the ones that have the highest bias. Most of these lie near the boundaries of the simplex, particularly near the corners. However, not all corners and boundaries are far

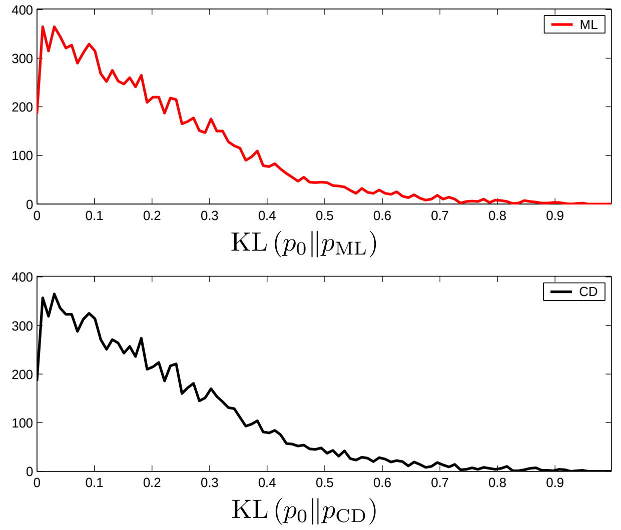

Figure 2: Histograms of KL $(p_0 \| p_{\mathrm{ML}})$ and KL $(p_0 \| p_{\mathrm{CD}})$ for VBM(2) after learning on 10^4 data distributions, where p_{ML} and p_{CD} are the convergence points for ML and CD, respectively. The performance of CD is very close to ML on average.

from the model manifold; this depends on the geometry of the model. In fig. 4 (for $v = 2$) we can discern the geometry of the simplex in fig. 1. The discontinuity in the slope at a Euclidean distance $\|p_0 - u\|$ just less than 0.3 corresponds to the radius of the inscribed sphere. The branch in the scatterplot which has low error corresponds to the direction passing through the centre and the simplex corner corresponding to the delta distribution of the $(1, 1)$ state (i.e., along the VBM manifold). The other branch which has high error and more data points corresponds to the directions passing through the centre and any of the other 3 corners (i.e., away from the VBM manifold).

As v increases, most of the volume of the simplex concentrates at a distance intermediate between the corners and the centre, close to the radius of the inscribed hypersphere. Consequently, a finite uniform sample contains essentially no points near the boundaries of the simplex, which produce the highest bias. For large v, CD has very small bias for nearly all randomly chosen data distributions. Only those rare distributions near the simplex boundaries produce a significant bias, but these are important in practice: real-world distributions are often near the boundaries (though not as far as the corners) because large parts of the data space have negligible probability.

Fig. 5 shows some typical learning curves. Both CD and ML decrease in a similar way, converging at the same rate (first-order), taking the same number of iterations to converge to a given tolerance. CD yields a higher KL error. In the lower example, the CD curve increases slightly at the end, suggesting it came close to the ML optimum but then moved away.

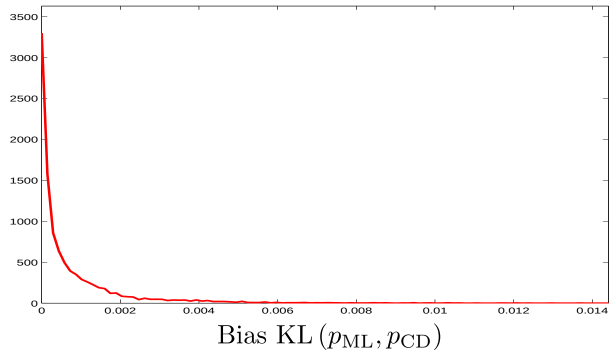

Figure 3: Histogram of the symmetrised KL divergence for VBM(2) between the model distributions found by ML and CD for all 10^4 data distributions. This shows that the bias of CD is almost always very small ($< 5\%$ of the KL error obtained by ML for the same distribution; data not shown). However, data distributions do exist that have a relatively large bias.

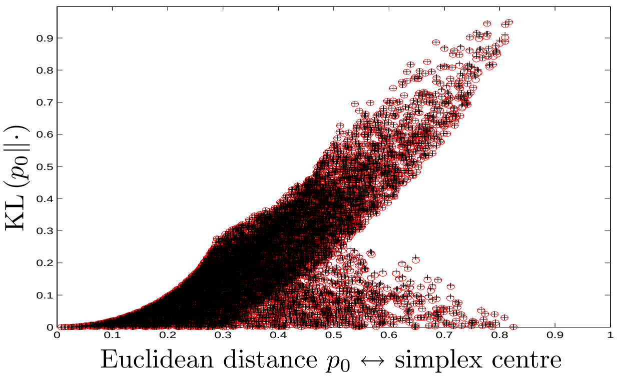

Figure 4: KL error for ML KL $(p_0 \| p_{\mathrm{ML}})$ (red $\circ$) and for CD KL $(p_0 \| p_{\mathrm{CD}})$ (black $+$) vs Euclidean distance $\|p_0 - u\|$ between the data distribution and the uniform distribution (centre of the simplex). This Euclidean distance gives a linear ordering of the data distributions (lowest Euclidean distance: p_0 is the uniform distribution, $\|p_0 - u\| = 0$; highest: p_0 is one of the corners of the simplex, $\|p_0 - u\| = \sqrt{1 - 2^{-v}}$). For clarity, not all 10^4 distributions are plotted.

In summary, we find the CD bias to be very small for most distributions and to be highest (but still small) with real-world distributions (near the simplex boundary). This bias is small in relative terms (compared to the KL error for ML) and absolute terms (compared to the simplex dimensions). CD and ML converge at about the same rate, but an ML iteration costs much

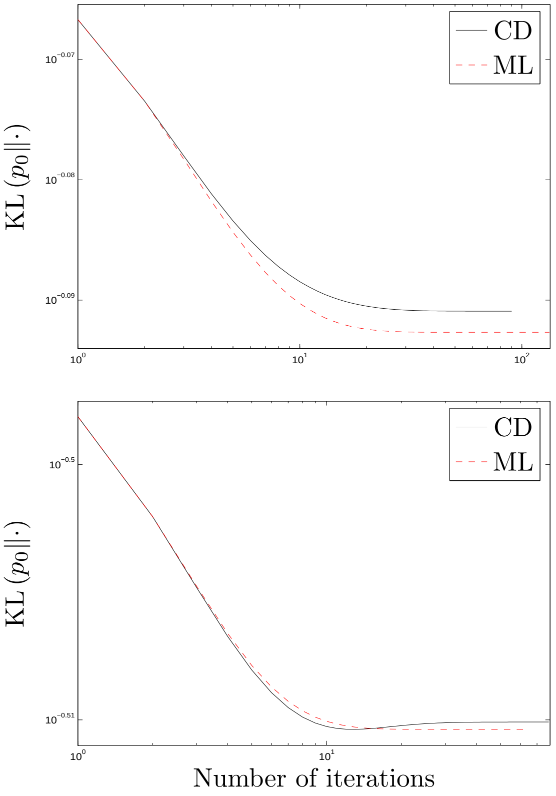

Figure 5: Learning curves for ML and CD for 2 randomly chosen data distributions. Axes are in log scale.

more than a CD one in a MCMC implementation.

4 Experiments with restricted BMs

RBMs are practically more interesting than VBMs, since they have a higher representational power. They also introduce a new element that complicates our study: the existence of multiple local optima of ML and CD. This prevents the characterisation of the bias over a large number of data distributions. Instead, we can only afford to select one data distribution (or a few) and try to characterise the set of all optima of ML and CD.

Given a data distribution, we generate a collection of 60 random initial weight vectors $\mathbf{W}$ and compute all the optima of ML and CD that are reachable from any of the initial weight vectors or from the optima found by the other learning method. This requires iterating over the current set of optima with ML and CD, until no new optima are found. The result is a bipartite, self-consistent convergence graph where an arrow $A \rightarrow B$ indicates that ML optimum A converges to CD optimum B under CD, or CD optimum A converges to ML optimum B under ML. Using many different initial weight vectors should give a representative collection of optima and a Good-Turing estimator (Good, 1953) can be used as a coarse indicator of how many optima we missed. The graph depends on how we decide whether two very similar optima are really the same. The threshold and number of parameter updates have to be carefully chosen so that truly different optima are not confused but two discoveries of the same optimum are not considered different. We found that using the symmetrised KL distance with a threshold of 0.01 worked well with 10^5 parameter updates.

We ran experiments for various values of v and h and various data distributions. Fig. 6 summarises the results for one representative case, corresponding to: $v = 6$, $h = 4$. The data distribution was generated from a data set of 4 binary vectors by adding an extra count of 0.1 to each possible binary vector and renormalising (thus it is close to the simplex boundary). We used 10^5 iterations and 60 different initial weight vectors: 20 random $\mathcal{N}(0, \sigma = 0.1)$, 20 random $\mathcal{N}(0, \sigma = 1)$ and 20 random $\mathcal{N}(0, \sigma = 10)$. We found 27 ML optima and 28 CD optima, and missed about 3 and 4, respectively (Good-Turing estimate).

Panel **A** shows a 2D visualisation of ML optima (red $\circ$) and CD optima (black $+$), i.e., the visible-unit distributions p_{ML} and p_{CD}, and their convergence relations. The blue ☆ is the data distribution p_0 (of the visible variables). To avoid cluttering the plot, pairs of arrows $A \leftrightarrows B$ are drawn as a single line without arrowheads $A — B$. Note that many such lines are too short to be distinguished. The 2D view was obtained with SNE (Hinton and Roweis, 2003) which tries to preserve the local distances. Using a perplexity of 3 to determine the local neighborhood size, SNE gives a better visualisation than projecting onto the first 2 principal components.

This panel shows an important and robust phenomenon: ML and CD optima typically come in pairs that converge to each other. The CD optimum always has a greater or equal KL error than its associated ML optimum but the difference is small. These pairs are to be expected for CD$_n$ when n is large because CD becomes ML as $n \rightarrow \infty$. However they occur very often even for $n = 1$, as shown. Panels **B**–**C** show that the choice of initial weights has a much larger effect on the KL error than the CD bias.

5 Using CD to initialise ML

The previous experiments show that CD takes us close to an ML optimum, but that a small bias remains. An obvious way to eliminate this bias is to use increasing values of n as training progresses. In this section we explore a crude version of this strategy: run CD until it is close to convergence then use a short run of ML

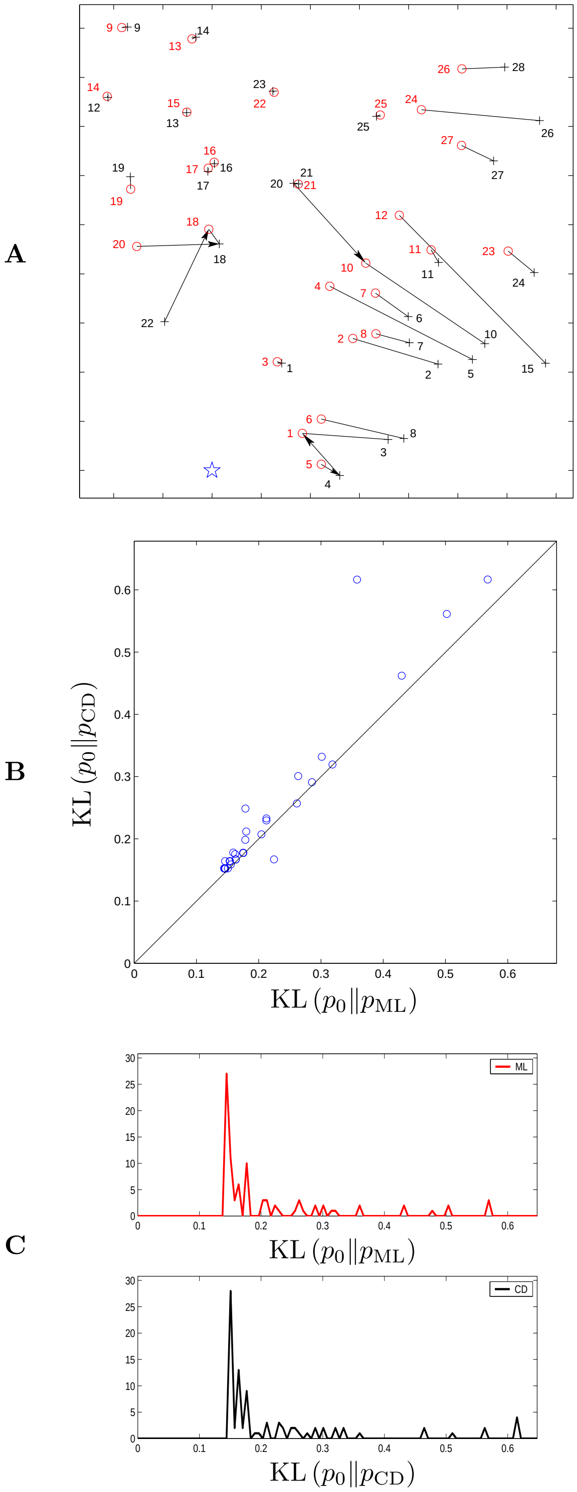

Figure 6: Empirical study of the convergence points of ML and CD for RBM(6, 4) with a single data distribution. **A**: 2D SNE visualization of the points (ML: red ○; CD: black +), and convergence relations among them with ML and CD (a line without an arrowhead stands for two arrows ⇆, to avoid clutter). **B**: KL error of CD vs ML from the same initial weight vectors, for 60 random initial weight vectors. **C**: histograms of the KL error of ML and CD.

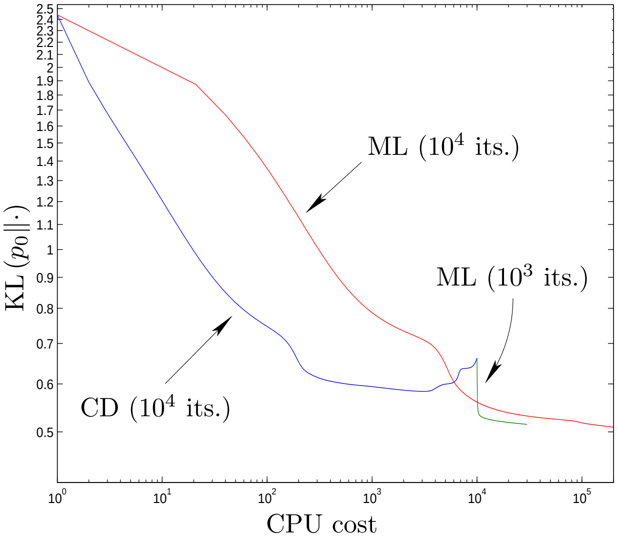

Figure 7: Learning curves for CD-ML and ML, where each ML iteration is scaled to cost as much as 20 CD iterations. The axes have a log scale, so CD-ML is an order of magnitude faster for about the same final KL.

to reduce the bias. We call this strategy CD-ML. We use a data distribution which is more representative of a real problem. It is located on the simplex boundary and derived from the statistics of all 3×3 patches in 11000 16×16 images of handwritten digits from the USPS dataset. The 256 intensity levels are thresholded at 150 to produce 9-dimensional binary vectors (thus $v = 9$). p_0 is the normalised counts of each of these binary vectors in the 2 156 000 patches.

We used 60 different initial weight vectors: 20 random $\mathcal{N}(0, \sigma = \frac{1}{3})$, 20 random $\mathcal{N}(0, \sigma = 1)$ and 20 random $\mathcal{N}(0, \sigma = 3)$. For each starting condition, two types of learning were used: ML learning for 10^4 iterations; and CD learning for 10^4 iterations, followed by a shorter run of ML learning. We ran experiments for $h \in \{1, \ldots, 8\}$ and found that there is a unique ML optimum and several CD optima of varying degrees of bias. If CD learning was followed by 1 000 iterations of ML, all the CD optima converged to the ML optimum.

Fig. 7 shows the learning curves (i.e., the error KL $(p_0\|\cdot)$ as a function of estimated CPU time) for the different methods with $h = 8$: CD (blue line), the short ML run (1 000 iterations) following CD (green line) and ML (red line), for a selected starting condition. We assume that each ML iteration costs 20 times as much as each CD iteration (a reasonable estimate for the size of this RBM). CD-ML reaches the same error as ML but at a small fraction of the cost. Note how sharply the CD-ML curve drops when we switch to ML, suggesting good performance can be achieved with very few of the expensive ML iterations.

6 Conclusion

Our first result is negative: for two types of Boltzmann machine we have shown that, in general, the fixed points of CD differ from those of ML, and thus CD is a biased algorithm. This might suggest that CD is not a competitive method for ML estimation of random fields. Our remaining, empirical results show otherwise: the bias is generally very small, at least for Gibbs sampling, since CD typically converges very near an ML optimum. And this small bias can be eliminated by running ML for a few iterations after CD, i.e., using CD as an initialisation strategy for ML, with a total computation time that is much smaller than that of full-fledged ML (which will also have slight bias because the Markov chain cannot be run forever).

The theoretical analysis of CD is difficult because of the complicated form that the p_1 (or p_n) distribution takes; p_1 is a moving target that changes with $\mathbf{W}$ in a complicated way, and depends on the sampling scheme used (e.g. Gibbs sampling). As a result, very few theoretical results about CD exist. MacKay (2001) gave some examples of CD bias, but these used unusual sampling operators. Our analysis applies to any model and operator (through the $\mathbf{G}$ and $\mathbf{T}$ matrices), in particular generally applicable operators such as Gibbs sampling. Williams and Agakov (2002) showed that, for 2D Gaussian Boltzmann machines, CD is unbiased and typically decreases the variance of the estimates. Yuille (2004) gives a condition for CD to be unbiased, though this condition is difficult to apply in practice.

One open theoretical problem is whether the exact version of CD converges (we believe that it does). Assuming we can prove convergence for the exact case, the right tools to use to prove it in the noisy case are probably those of stochastic approximation (Benveniste et al., 1990; Yuille, 2004).

Acknowledgements

This research was funded by NSERC and CFI. GEH is a fellow of CIAR and holds a CRC chair.

References

A. Benveniste, M. Métivier, and P. Priouret. *Adaptive Algorithms and Stochastic Approximations.* Springer-Verlag, 1990.

M. Á. Carreira-Perpiñán and G. E. Hinton. On contrastive divergence (CD) learning. Technical report, Dept. of Computer Science, University of Toronto, 2004. In preparation.

H. Chen and A. F. Murray. Continuous restricted Boltzmann machine with an implementable training algorithm. *IEE Proceedings: Vision, Image and Signal Processing*, 150(3):153–158, June 20 2003.

Y. Freund and D. Haussler. Unsupervised learning of distributions on binary vectors using 2-layer networks. In *NIPS*, pages 912–919, 1992.

W. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice.* Chapman & Hall, 1996.

I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3/4):237–264, Dec. 1953.

X. He, R. S. Zemel, and M. Á. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *CVPR*, pages 695–702, 2004.

G. Hinton and S. T. Roweis. Stochastic neighbor embedding. In *NIPS*, pages 857–864, 2003.

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

S. Z. Li. *Markov Random Field Modeling in Image Analysis.* Springer-Verlag, 2001.

D. J. C. MacKay. Failures of the one-step learning algorithm. Available online at `http://www.inference.phy.cam.ac.uk/mackay/abstracts/gbm.html`, 2001.

R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG–TR–93–1, Dept. of Computer Science, University of Toronto, Sept. 1993. Available online at `ftp://ftp.cs.toronto.edu/pub/radford/review.ps.Z`.

P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart and J. L. MacClelland, editors, *Parallel Distributed Computing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations*, chapter 6. MIT Press, 1986.

Y. W. Teh, M. Welling, S. Osindero, and G. E. Hinton. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4:1235–1260, Dec. 2003.

C. K. I. Williams and F. V. Agakov. An analysis of contrastive divergence learning in Gaussian Boltzmann machines. Technical Report EDI–INF–RR–0120, Division of Informatics, University of Edinburgh, May 2002.

G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods.* Springer-Verlag, second edition, 2002.

A. Yuille. The convergence of contrastive divergences. To appear in NIPS, 2004.