Neighborhood-Based Stopping Criterion for Contrastive Divergence

Enrique Romero Merino, Ferran Mazzanti Castrillejo, and Jordi Delgado Pin

***Abstract*— Restricted Boltzmann Machines (RBMs) are general unsupervised learning devices to ascertain generative models of data distributions. RBMs are often trained using the Contrastive Divergence (CD) learning algorithm, an approximation to the gradient of the data log-likelihood (logL). A simple reconstruction error is often used as a stopping criterion for CD, although several authors have raised doubts concerning the feasibility of this procedure. In many cases, the evolution curve of the reconstruction error is monotonic, while the logL is not, thus indicating that the former is not a good estimator of the optimal stopping point for learning. However, not many alternatives to the reconstruction error have been discussed in the literature. An estimation of the logL of the training data based on annealed importance sampling is feasible but computationally very expensive. In this manuscript, we present a simple and cheap alternative, based on the inclusion of information contained in neighboring states to the training set, as a stopping criterion for CD learning.**

***Index Terms*— Machine learning, neural networks, recurrent neural networks, restricted Boltzmann machines, unsupervised learning.**

I. Introduction

LEARNING algorithms for deep multilayer neural networks have been known for a long time [1], though they usually could not outperform simpler shallow networks. In this way, deep multilayer networks were not widely used to solve large scale real-world problems until the last decade [2], [3]. In 2006, deep belief networks (DBNs) [4] came out as a real breakthrough in this field, since the learning algorithms proposed ended up being a feasible and practical method to train deep networks, with interesting results [5]–[9]. DBNs have restricted Boltzmann machines (RBMs) [10] as their building blocks.

RBMs are topologically constrained Boltzmann Machines (BMs) with two layers, one of hidden and another of visible neurons, and no intralayer connections. This property makes working with RBMs simpler than with regular BMs and, in particular, the stochastic computation of the logL gradient may be performed more efficiently by means of Gibbs sampling [2], [11].

In 2002, the *contrastive divergence* (CD) learning algorithm was proposed as an efficient training method for product-of-expert models, from which RBMs are a special case [12]. It was observed that using CD to train RBMs worked quite well in practice. This fact was important for deep learning since some authors suggested that a multilayer deep neural network is better trained when each layer is pre-trained separately as if it was a single RBM [5], [6], [13]. Thus, training RBMs with CD and stacking them up seems to be a good way to go when designing deep learning architectures.

However, the picture is not as nice as it looks, since CD is not a flawless training algorithm. Despite CD being an approximation of the true logL gradient [14], it is biased and it may not converge in some cases [15]–[17]. Moreover, it has been observed that CD and variants such as persistent CD (PCD) [18] or fast PCD [19] can lead to a steady decrease of the logL during learning [20]–[22]. Therefore, the risk of learning divergence imposes the requirement of a stopping criterion. There are two main methods used to decide when to stop the learning process. One is based on the monitorization of the *reconstruction error* [23]. The other is based on the estimation of the logL with *annealed importance sampling* (AIS) [24], [25]. The reconstruction error is easy to compute and it has been often used in practice, though its adequacy remains unclear because of monotonicity [21]. AIS seems to work better than the reconstruction error in most cases, though it is considerably more expensive to compute, and may also fail [20].

In this paper, we approach this problem from a different perspective. In general, CD_n tends to concentrate the probabilities in a small subset of the training data, leaving little probabilities to the rest of states. This is an undesired feature that prevents building a good model. In this paper, we propose a stopping criterion that tries to detect this before the likelihood starts to degenerate. Since in a Boltzmann distribution the probability of a given state is proportional to the exponential of its energy, states with similar energy also have similar probability. Based on the fact that the energy is a continuous and smooth function of its variables, the close neighborhood of the high-probability states is also expected to acquire a significant amount of probability. In this sense, we argue that the information contained in the neighborhood of the training data is valuable, and that it can be incorporated in the learning process of RBMs. We use the Hamming distance as a measure of how close different states are.

Manuscript received March 3, 2016; revised October 28, 2016 and February 6, 2017; accepted April 14, 2017. Date of publication May 17, 2017; date of current version June 21, 2018.

The work of Enrique Romero Merino was supported by the Spanish Research Project under Grant TIN2016-79576-R. The work of Ferran Mazzanti Castrillejo was supported by DGI, Spain, under Grant FIS2014-56257-C2-1-P. The work of Jordi Delgado Pin was supported in part by the Generalitat de Catalunya under Grant SGR2014-890 (MACDA) and in part by the MINECO project under Grant APCOM (TIN2014-57226-P). *(Corresponding author: Ferran Mazzanti Castrillejo.)*

E. R. Merino and J. D. Pin are with the Departament de Ciències de la Computació, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain (e-mail: eromero@cs.upc.edu; jdelgado@cs.upc.edu).

F. M. Castrillejo is with the Departament de Física i Enginyeria Nuclear, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain (e-mail: ferran.mazzanti@upc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2697455

The proposed stopping criterion depends on the information contained in the training set and its neighbors and can be used to detect changes in the curvature of the logL. In this sense, the criterion is local as it does not need to explore the whole space of states. Furthermore, and in order to make it computationally tractable, we build the stopping criterion in such a way that it becomes independent of the partition function of the model, which is computationally intractable in real-world large spaces. Moreover, the proposed quantity we monitor during learning is much cheaper to evaluate than the estimated logL using AIS. In the following sections, we define the neighborhood-based stopping criterion for CD_n and show its performance in several data sets.

II. Learning in Restricted Boltzmann Machines

A. Energy-Based Probabilistic Models

Energy-based probabilistic models define a probability distribution from an energy function, as follows:

$$P(\boldsymbol{x}, \boldsymbol{h}) = \frac{e^{-\text{Energy}(\boldsymbol{x}, \boldsymbol{h})}}{Z} \tag{1}$$

where $\boldsymbol{x}$ and $\boldsymbol{h}$ stand for (typically binary) visible and hidden variables, respectively. The normalization factor Z is called partition function and reads

$$Z = \sum_{\boldsymbol{x}, \boldsymbol{h}} e^{-\text{Energy}(\boldsymbol{x}, \boldsymbol{h})}. \tag{2}$$

Since only $\boldsymbol{x}$ is observed, one is interested in the marginal distribution

$$P(\boldsymbol{x}) = \frac{\sum_{\boldsymbol{h}} e^{-\text{Energy}(\boldsymbol{x}, \boldsymbol{h})}}{Z} \tag{3}$$

but the evaluation of the partition function Z is computationally prohibitive since it involves an exponentially large number of terms. In this way, one cannot measure directly $P(\boldsymbol{x})$.

The energy function depends on several parameters θ, which are adjusted at the learning stage. This is done by maximizing the likelihood of the data. In energy-based models, the derivative of the logL can be expressed as

$$\begin{aligned} -\frac{\partial \log P(\boldsymbol{x}; \theta)}{\partial \theta} = \; & E_{P(\boldsymbol{h}|\boldsymbol{x})}\left[\frac{\partial \text{Energy}(\boldsymbol{x}, \boldsymbol{h})}{\partial \theta}\right] \\ & - E_{P(\widetilde{\boldsymbol{x}})}\left[E_{P(\boldsymbol{h}|\widetilde{\boldsymbol{x}})}\left[\frac{\partial \text{Energy}(\widetilde{\boldsymbol{x}}, \boldsymbol{h})}{\partial \theta}\right]\right] \end{aligned} \tag{4}$$

where the first term is called the positive phase and the second term is called the negative phase. In this expression, $E_{P(\widetilde{\boldsymbol{x}})}$ stands for the expectation value over the probability of the visible states, and involves the evaluation of the partition function according to the definition $E_{P(\widetilde{\boldsymbol{x}})}[f(\widetilde{\boldsymbol{x}})] = \sum_{\widetilde{\boldsymbol{x}}} P(\widetilde{\boldsymbol{x}}) f(\widetilde{\boldsymbol{x}})$ with $P(\widetilde{\boldsymbol{x}})$ defined in (3). As it can be seen, the exact computation of the derivative of the logL is usually unfeasible because of the negative phase in (4), which comes from the derivative of the partition function.

B. Restricted Boltzmann Machines

RBMs are energy-based probabilistic models whose energy function is

$$\text{Energy}(\boldsymbol{x}, \boldsymbol{h}) = -\boldsymbol{b}^t \boldsymbol{x} - \boldsymbol{c}^t \boldsymbol{h} - \boldsymbol{h}^t \boldsymbol{W} \boldsymbol{x} \tag{5}$$

where $\boldsymbol{W}$ is the two-body weights connecting pairs of hidden and visible units and $\boldsymbol{b}$ and $\boldsymbol{c}$ are the corresponding bias terms. RBMs are at the core of DBNs [4] and other deep architectures that use RBMs for unsupervised pre-training previous to the supervised step [5], [6], [13].

The consequence of the particular form of the energy function is that in RBMs both $P(\boldsymbol{h}|\boldsymbol{x})$ and $P(\boldsymbol{x}|\boldsymbol{h})$ factorize. In this way, it is possible to compute $P(\boldsymbol{h}|\boldsymbol{x})$ and $P(\boldsymbol{x}|\boldsymbol{h})$ in one step, making it possible to perform Gibbs sampling efficiently, in contrast to more general models like Boltzmann machines [26].

C. Contrastive Divergence

The most common learning algorithm for RBMs uses an algorithm to estimate the derivative of the logL of a product of experts model. This algorithm is called CD [12].

CD_n estimates the derivative of the logL for a given point $\boldsymbol{x}$ as

$$\begin{aligned} -\frac{\partial \log P(\boldsymbol{x}; \theta)}{\partial \theta} \simeq \; & E_{P(\boldsymbol{h}|\boldsymbol{x})}\left[\frac{\partial \text{Energy}(\boldsymbol{x}, \boldsymbol{h})}{\partial \theta}\right] \\ & - E_{P(\boldsymbol{h}|\boldsymbol{x}_n)}\left[\frac{\partial \text{Energy}(\boldsymbol{x}_n, \boldsymbol{h})}{\partial \theta}\right] \end{aligned} \tag{6}$$

where $\boldsymbol{x}_n$ is the last sample from the Gibbs chain starting from $\boldsymbol{x}$ obtained after n steps:

1) $\boldsymbol{h}_1 \sim P(\boldsymbol{h}|\boldsymbol{x})$;
2) $\boldsymbol{x}_1 \sim P(\boldsymbol{x}|\boldsymbol{h}_1)$;
3) $\ldots$;
4) $\boldsymbol{h}_n \sim P(\boldsymbol{h}|\boldsymbol{x}_{n-1})$;
5) $\boldsymbol{x}_n \sim P(\boldsymbol{x}|\boldsymbol{h}_n)$.

Usually, $E_{P(\boldsymbol{h}|\boldsymbol{x})}[(\partial \text{Energy}(\boldsymbol{x}, \boldsymbol{h})/\partial \theta)]$ can be easily computed.

Several alternatives to CD_n are PCD [18], fast PCD [19], parallel tempering [22], dissimilar CD [27], average CD [28], or beyond mean field corrections [29].

D. Monitoring the Learning Process in RBMs

Learning in RBMs is a delicate procedure involving a lot of data processing that one seeks to perform at a reasonable speed in order to be able to handle large spaces with a huge amount of states. In doing so, drastic approximations that can only be understood in a statistically averaged sense are performed [30].

One of the most relevant points to consider at the learning stage is to find a good way to determine whether a good solution has been found or not, and so to decide when the learning process should stop. One of the most widely used criteria for stopping is based on the monitorization of the reconstruction error, which is a measure of the capability of the network to produce an output that is consistent with the data at input. Since RBMs are probabilistic models, the reconstruction error of a data point $\boldsymbol{x}^{(i)}$ is computed as the probability of $\boldsymbol{x}^{(i)}$ given the expected value of $\boldsymbol{h}$ for $\boldsymbol{x}^{(i)}$

$$R(\boldsymbol{x}^{(i)}) = -\log P(\boldsymbol{x}^{(i)} | E_{P(\boldsymbol{h}|\boldsymbol{x}^{(i)})}[\boldsymbol{h}]) \tag{7}$$

which is a probabilistic extension of the sum-of-squares reconstruction error for deterministic networks

$$\epsilon(\boldsymbol{x}^{(i)}) = ||\boldsymbol{x}^{(i)} - \boldsymbol{x}_n^{(i)}||^2. \tag{8}$$

In this expression, $\boldsymbol{x}_n^{(i)}$ stands for the nth reconstruction, in the Gibbs chain mentioned above, of the ith member of the training set. In practice, (7) is computed analytically. One first evaluates the expectation value of the hidden units for a given input, and then the conditional probability of the visible units given that.

Schulz *et al.* [20] and Fischer and Igel [21] have shown that, in some cases, learning induces an undesirable decrease in likelihood that goes undetected by the reconstruction error (both $R(\boldsymbol{x}^{(i)})$ and $\epsilon(\boldsymbol{x}^{(i)})$ usually decrease monotonically). Sinc°e no increase in the reconstruction error takes place during training, there is no apparent way to detect the change in behavior of the logL for CD$_n$.

Alternatively, one could evaluate an estimation of the likelihood of the training data by means of the AIS algorithm. While this is theoretically possible, it can be very expensive from a computational point of view when the system size is large, and in some cases, it is not even clear how well it performs [20].

III. Proposed Stopping Criterion

The proposed stopping criterion is based on the monitorization of the ratio of two quantities: the geometric average of the probabilities of the training set and the sum of probabilities of points in a given neighborhood of the training set. More formally, what we monitor is

$$\xi_d = \frac{\left[\prod_{i=1}^{N} P(\boldsymbol{x}^{(i)})\right]^{1/N}}{\sum_{\boldsymbol{y}\in D} P(\boldsymbol{y})} \tag{9}$$

where D is a subset of points at a Hamming distance less or equal than d from the training set. As usual, the distance between a given point and a data set is taken as the minimum distance from the given point to any element of the set. Notice that using points not in the training set to improve learning is also present in other works, as in [27].

The idea behind the definition is that the evolution of ξ_d at the learning stage is expected to capture the main trends of the logL for certain values of d and D. Notice that there are two interesting limiting cases. On one hand, when D spans the whole space (thus d being equal to the maximal possible Hamming distance), ξ_d is exactly the likelihood of the data since the denominator in (9) adds up to 1. On the other hand, only the data in the training set is involved in the calculation when $d = 0$. The choice of D and d is problem-dependent, but in any case one should make sensible choices, taking d small enough to have a local estimator and D of a reasonable size in order to make ξ_d computationally feasible.

In this paper, we propose to stop CD$_n$ learning at the maximum of ξ_d, which we expect to be close to the one shown by the logL of the data. This holds for suitable values of d and D, as shown by the experiments in the next sections.

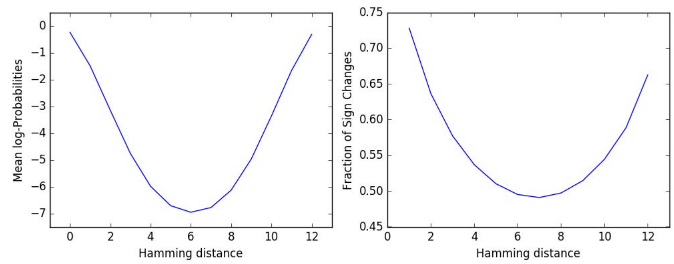


Fig. 1. Average logarithm of the probabilities (left) and fraction of sign changes for small weight changes (right) of the most probable states as a function of the Hamming distance for one thousand runs of an RBM with 12 visible units.

The motivation for the analytic form of ξ_d in (9) is twofold. On one hand, the numerator and the denominator monitor different things. The numerator, which is essentially the likelihood of the data, is sensitive to the accumulation of most of the probability mass by a reduced subset of the training data, a typical feature of CD$_n$. For continuity reasons, the denominator is strongly correlated with the sum of probabilities of the training data. Once the problem has been learned, the probabilities in a close neighborhood of the training set will be high. The value of ξ_d results from a delicate equilibrium between these two quantities (see Section IV). On the other hand, notice that as the partition function is the most expensive quantity to evaluate, we explicitly build ξ_d as a Z-independent quantity. This is a necessary condition we impose in the design of the quantity being monitorized. In this way, due to the structure of ξ_d, the partition functions Z involved in both the numerator and the denominator cancels out. In other words, the computation of ξ_d can be equivalently defined as

$$\xi_d = \frac{\left[\prod_{i=1}^{N} \sum_{\boldsymbol{h}} e^{-\text{Energy}(\boldsymbol{x}^{(i)},\boldsymbol{h})}\right]^{1/N}}{\sum_{\boldsymbol{y}\in D} \sum_{\boldsymbol{h}} e^{-\text{Energy}(\boldsymbol{y},\boldsymbol{h})}}. \tag{10}$$

The particular topology of RBMs allows to compute $\sum_{\boldsymbol{h}} e^{-\text{Energy}(\boldsymbol{z},\boldsymbol{h})}$ efficiently [2]. This fact dramatically decreases the computational cost involved in the calculation, which would otherwise become unfeasible in most real-world problems where RBMs could be successfully applied.

Defining the probabilistic neighborhood of each training sample is in general problematic because it clearly depends on the value of the weights and bias of the network, and can be computationally very expensive. The choice of the Hamming distance as a measure of probability proximity can be justified in a statistical sense, since the energy function is the sum of many terms involving a single bit from the visible units, one expects that changes in the total energy will be smaller the fewer bits are changed, at least in a small range of Hamming distances. Moreover, one also expects that changes in the probabilities of nearby states follow the same direction. In order to illustrate these points, we have conducted a series of synthetic experiments with randomly chosen Gaussian weights, such that a small fraction of the whole space acquires a significant amount of probability mass. In this way, our goal is to reproduce what it is usually found in learning problems, where the training set is small compared with the whole space. Fig. 1 shows the results of the average over one thousand runs on an RBM with 12 and 18 visible and hidden units, respectively. Parameters have been adjusted such that approximately a 5% of the total number of states exhausts approximately 0.8 of the total probability. The left panel shows the average probability of neighboring states to the most probable ones, as a function of the Hamming distance. As it can be seen from the plot, on average the probability is a smooth function of the Hamming distance that shows a monotonic behavior up to a certain point. The right panel shows the fraction of sign changes in the averaged probabilities when a small variation of less than a 1% in the weights is performed (which would account for a small update in a learning epoch). As it can be seen, most of the neighboring states follow the same sign changes as the original state, thus reinforcing the idea of continuity in probability space. In this way, the idea of using the Hamming distance as a measure of probabilistic similarity is supported in a statistical sense. Furthermore, it is one of the simplest and cheapest metrics to evaluate. It is clear that the Hamming distance may fail in some cases, but our criterion is based on the hypothesis that this is not the dominant case. In this way, other non-trivial metrics as the ones proposed in [31] and [32] could be used.

While the numerator in ξ_d is directly evaluated from the data in the training set, the problem of finding suitable values for $\boldsymbol{y} \in D$ still remains. Indeed, the set of points at a given Hamming distance d from the training set is independent of the weights and bias of the network. In this way, it can be built once at the very beginning of the process and be used as required during learning. Therefore, two issues have to be sorted out before the criterion can be applied. The first one is to decide a suitable value of d. Experiments with different problems show that this is indeed problem dependent, as is illustrated in the experimental section below. The second one is the choice of the subset D, which strongly depends on the size of the space being explored. For small spaces, one can safely use the complete set of points at a distance less than or equal to d, but that can be forbiddingly large in real world problems. For this reason, we explore two possibilities: one including all points and another including only a random subset of the same size as the training set, which is only as expensive as dealing with the training set. This is an arbitrary decision that can be changed at will, keeping in mind that one always needs a large enough set of points that does not, however, increase the computational cost significantly.

IV. Experiments

We performed several experiments to explore the aforementioned criterion defined in Section III and study the behavior of ξ_d in comparison with the logL and the reconstruction error of the data in several problems. For an exact analysis, we have explored problems of a size such that the logL can be exactly evaluated and compared with the proposed ξ_d parameter. Moreover, we have also included the results for large benchmarking data sets, where the calculation of the exact logL is unfeasible and has been approximated with the AIS algorithm [25].

A. Small Problems

The first small problem, denoted *bars and stripes* (BS), tries to identify vertical and horizontal lines in 4×4 pixel images. The training set consists in the whole set of images containing all possible horizontal or vertical lines (but not both), ranging from no lines (blank image) to completely filled images (black image), thus producing $2 \times 2^4 - 2 = 30$ different images (avoiding the repetition of fully back and fully white images) out of the space of 2^{16} possible images with black or white pixels. The second small problem, named *labeled shifter ensemble* (LSE), consists in learning 19-b states formed as follows: given an initial 8-b pattern, generate three new states concatenating to it the bit sequences 001, 010, or 100. The final 8-b pattern of the state is the original one shifting one bit to the left if the intermediate code is 001, copying it unchanged if the code is 010, or shifting it one bit to the right if the code is 100. One thus generates the training set using all possible $2^8 \times 3 = 768$ states that can be created in this form, while the system space consists of all possible 2^{19} different states one can build with 19 b. These two problems have already been explored in [21] and are adequate in the current context since, while still large, the dimensionality of space allows for a direct monitorization of the partition function and the logL during learning. For the sake of completeness, we have also tested the proposed criterion on randomly generated problems with different space dimensions, where the number of states to be learned is significantly smaller than the size of the space. In particular, we have generated four different data sets (RAN10, RAN12, RAN14, and RAN16) consisting of $N_v = 10, 12, 14, 16$ binary input units and $2^{N_v/2}$ examples to be learned, as suggested in [33].

In the following, we discuss the learning processes of these problems with binary RBMs, employing the CD algorithm CD_n with $n = 1$ and $n = 10$ as described in Section II-C. In the BS case, the RBM had 16 visible and 8 hidden units, while in the LSE problem these numbers were 19 and 10, respectively. For the random data sets, we have used 10 hidden units in each case.

Every simulation was carried out for a total of 50 000 epochs, with measures being taken every 50 epochs. Moreover, every point in the subsequent plots is the average of ten different simulations starting from different random values of the weights and bias. No weight decay was used, and momentum was set to 0.8. The learning rates (LRs) were chosen in order to make sure that the logL degenerates, in such a way that it presents a clear maximum that should be detected by ξ_d.

In the following sections, we present results for two series of experiments. In the first one (Section IV-A1), we analyze the case where all states in D are included for a given d. In the second one (Section IV-A2), we relax the computational cost of the evaluation of ξ_d by selecting only a small subset of all the states in D.

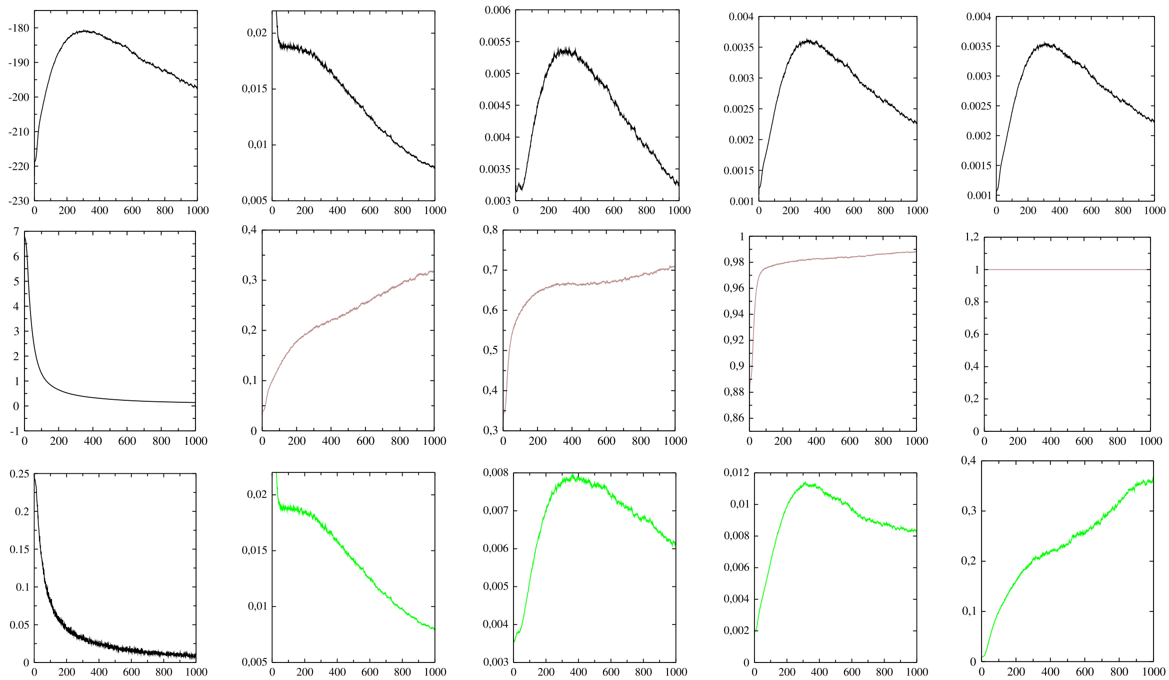


Fig. 2. Results for the RAN10 problem. The first column shows the logL (top) and the reconstruction errors (7) and (8) (center and bottom). The other columns in the first, second, and third rows depict ξ_d for $D = D_A$ (black curves), the sum of probabilities in the denominator of ξ_d for $D = D_A$ (brown curves) and ξ_d for $D = D_S$ (green curves) for $d = 0, 1, 2, 3$, respectively. The x-axis is the number of epochs along the simulation divided by 50 in all plots. All data in the y-axis are in arbitrary units.

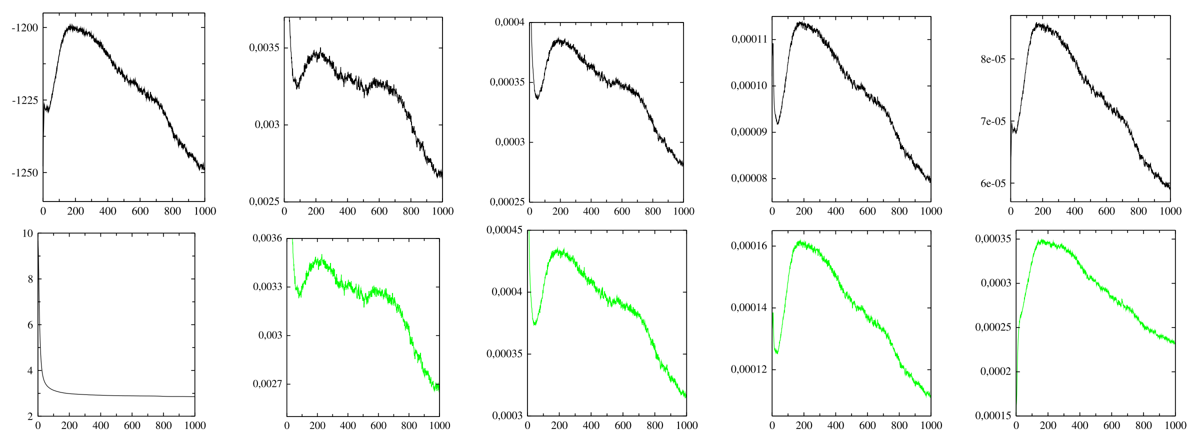


Fig. 3. Results for the RAN14 problem. The first column shows the logL and the reconstruction error (7) (top and bottom). The other columns in the upper and lower rows show ξ_d for $D = D_A$ and ξ_d for $D = D_S$ for $d = 0, 1, 2, 3$, respectively.

1) Complete Neighborhoods: We present the results for the problems at hand, showing for each analyzed instance different plots corresponding to the actual logL of the problem and ξ_d for different values of d, among other things. In order to identify the contributions to ξ_d from the different neighborhoods of the training set, we define two different sets: D_A containing all states at a distance less than or equal to d, and D_S accounting for those states at a distance exactly equal to d. We have computed ξ_d for $D = D_A$ and $D = D_S$ in all our experiments that are commented in the following.

Fig. 2 shows our results for the RAN10 data set. The upper left panel shows the logL of the data during training. As it can be seen, there is a clear maximum that should be identified as the stopping point. The panels below show the reconstruction errors (7) and (8), which clearly fail to identify the desired extremum. The rest of the columns show results for distances $d = 0, 1, 2$, and 3. The first row depicts ξ_d for D_A, where all states at the required distances are taken into account. As it can be seen, starting at $d = 1$, the criterion is robust and consistently detects the maximum of the logL at the right place, thus reinforcing the idea that the neighborhood of the data contains valuable information. The second row shows the denominator of ξ_d corresponding to the first row, that is, the sum of probabilities of the states included in each case. Notice that for $d = 3$, this sum equals one and ξ_d is exactly equal to the likelihood of the data. More interestingly, even when the sum is still far away from one, as it happens for $d = 1$, ξ_d consistently finds the desired point. This behavior is also

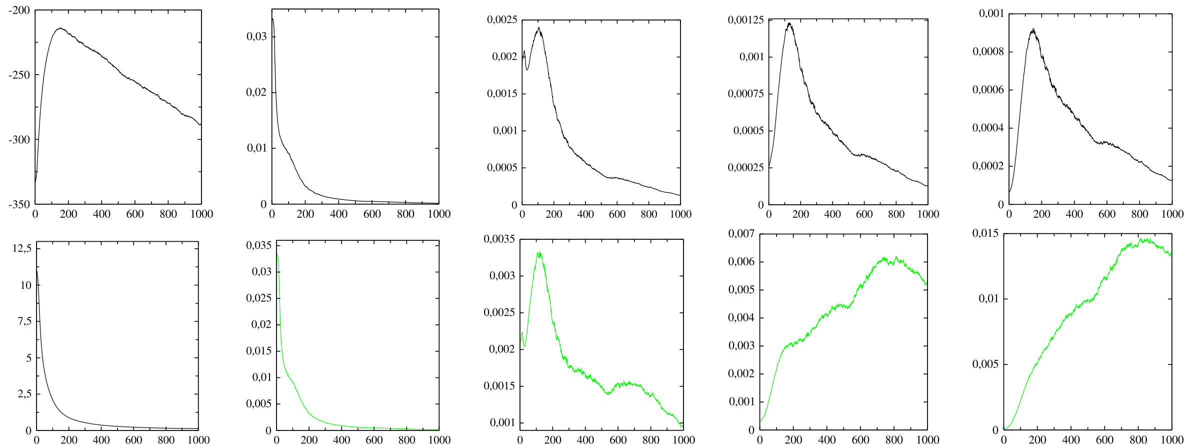


Fig. 4. Same as in Fig. 3 for the BS data set.

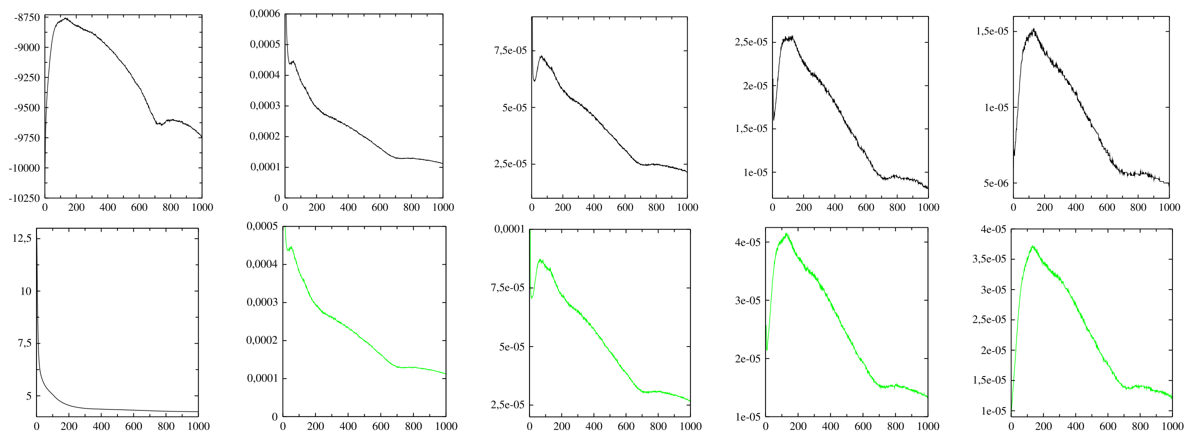


Fig. 5. Same as in Fig. 3 for the LSE data set.

TABLE I

NUMBER OF NEIGHBORS AT DIFFERENT HAMMING DISTANCES FOR THE BS AND LSE DATA SETS

Data Set	Hamming Distance									
	1	2	3	4	5	6	7	8	9	10
Bars and Stripes	480	3216	11360	20744	19296	8688	1632	90	-	-
Labeled Shifter Ensemble	8434	41160	110326	165088	132976	54160	10368	966	40	2

observed in the rest of the data sets analyzed. Finally, the third row shows ξ_d for D_S, thus showing the behavior of the criterion applied to different shells. For $d = 1$ and 2, the criterion detects reasonably well the maximum of the logL and can be used to identify the desired stopping point. Notice, though, that the data alone, entirely contained at $d = 0$, are not capable to reproduce this behavior. Moreover, for d larger than 2, the criterion also fails, as it is expected that starting at a certain distance the information regarding the model is lost. Please notice that the initial transitory behavior of some of the plots above is meaningless and can be omitted so it has been cut.

Equivalent results for the RAN14 case are shown in Fig. 3. The logL and the probabilistic reconstruction error in (7) are depicted in the upper and lower panels in the first column, respectively. The other panels show ξ_d for D_A and D_S, with $d = 0, 1, 2, 3$ (top and bottom rows, second to fifth columns). As in the previous case, the reconstruction error fails to detect the maximum of the likelihood, thus not being very useful in the present context. On the contrary, a stopping point obtained from ξ_d selects a near-optimal model. Notice that the criterion is robust along all distances explored, as desired. Similar results are found for the RAN12 and RAN16 cases.

The same plots for the BS and LSE problems are found in Figs. 4 and 5. Once again, the reconstruction error decreases monotonously and is therefore useless in the present context. For the LSE problem, ξ_d for d larger than 1 successfully does the task for $D = D_A$ and $D = D_S$. However, in the BS case, it works for $D = D_A$ but not for $D = D_S$ and $d > 1$. As it can be inferred from these results, the optimal value of d can not be fixed beforehand and is problem-dependent.

2) Incomplete Neighborhoods: Despite the success of the criterion built for $D = D_A$, it is clear that for large spaces it can be unpractical if the number of states in the neighborhood of the training set is very large. For that reason, we have tested

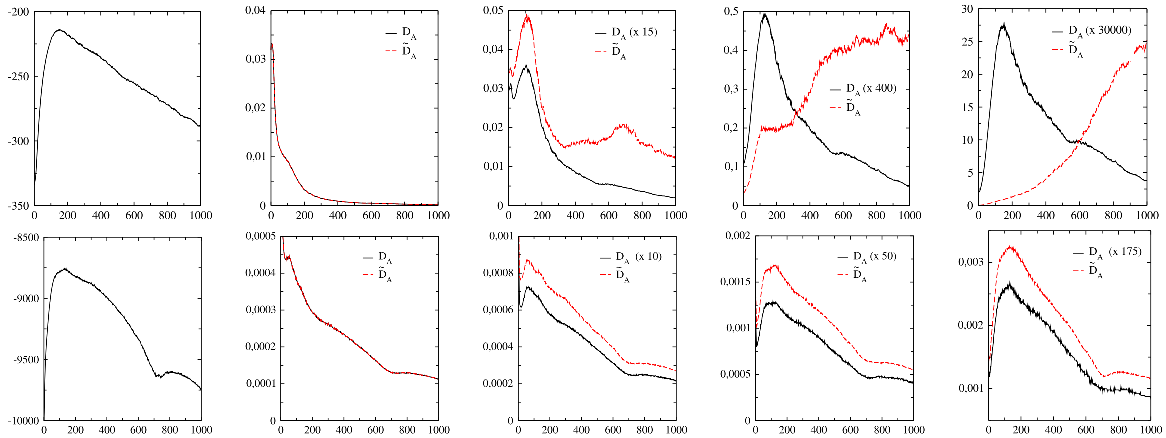


Fig. 6. Comparison between ξ_d (black curves) and $\tilde{\xi}_d$ (red curves) for the BS and LSE data sets (upper and lower rows). Notice that since the magnitude of these parameters is irrelevant, some curves have been scaled for the sake of clarity. The first column plots the logL of the data along the simulation.

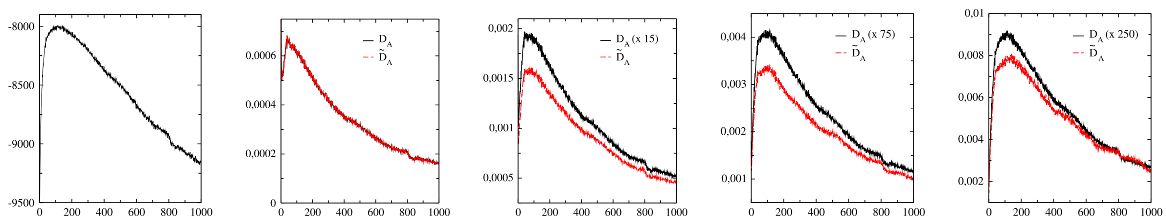


Fig. 7. Same as in Fig. 6 for the LSE problem in CD_{10}.

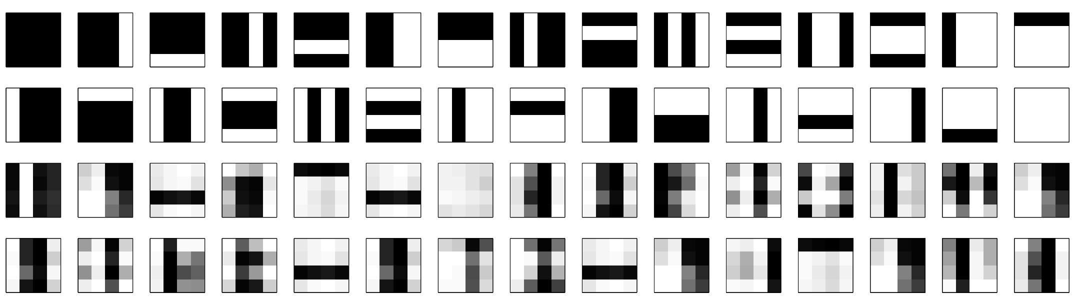


Fig. 8. Training data (two upper rows) and generated samples (two lower rows) for the BS problems with the weights and bias obtained at the stopping point detected by $\tilde{\xi}_d$ with $d = 1$.

the criterion on randomly selected subsets $\tilde{D}_A \subset D_A$ of the same size as the training set, which is always computationally tractable. In this sense, we denote by $\tilde{\xi}_d$, the evaluation of ξ_d on $\tilde{D}_A$. Fig. 6 shows $\tilde{\xi}_d$ compared with ξ_d from the previous figures for the BS (first row) and LSE (second row) problems. More precisely, the first column shows the logL of the data along the training process, while the rest of the columns plot both $\tilde{\xi}_d$ and ξ_d for $d = 0, 1, 2$, and 3. Notice that the absolute scales of ξ_d and $\tilde{\xi}_d$ may vary mainly due to the value of the sum of probabilities in the denominators. However, since the precise value of these quantities is irrelevant, we have decided to scale them properly for the sake of comparison. Although $\tilde{\xi}_d$ is built from a much smaller set than ξ_d, in most cases it captures the significant features of ξ_d and can therefore be used instead of it. In this sense, $\tilde{\xi}_d$ provides a good stopping criterion for CD_1, although it is not as robust as ξ_d due to the strong reduction of states contributing to $\tilde{\xi}_d$ as compared with those entering in ξ_d. This reduction is illustrated in Table I, where we show the number of neighboring states to the data set at different distances for the BS and LSE problems. By increasing the number of states included in $\tilde{\xi}_d$, convergence to ξ_d is expected at the expense of an increase in computational cost. However, the present results indicate that, at least for the problems at hand, a number of examples similar to that of the training set in the evaluation of $\tilde{\xi}_d$ are enough to detect the maximum of the logL of the data.

All the results presented up to this point show the goodness of the proposed stopping criterion for learning in CD_1. However, the underlying idea can be applied to different learning algorithms that try to maximize the logL of the data. In this way, we have repeated all the previous experiments for CD_{10} with very similar results to the ones above. As an example, Fig. 7 shows the logL, ξ_d, and $\tilde{\xi}_d$ with $d = 0, 1, 2, 3$ and CD_{10} for the LSE data set. As it is clearly seen, the quality

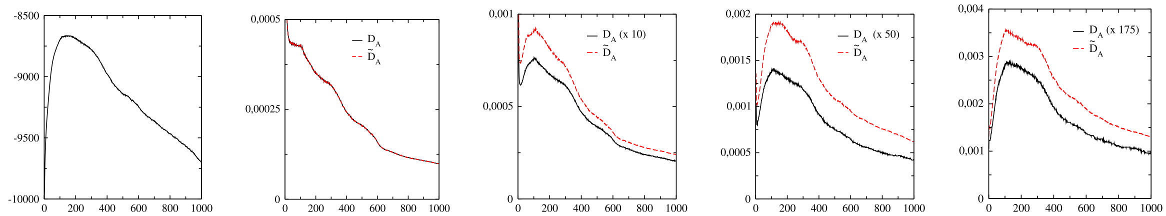


Fig. 9. Same as in Fig. 6 for the LSE problem in PCD.

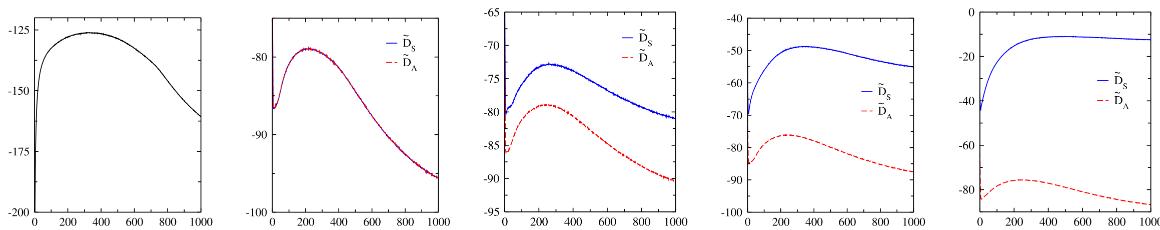


Fig. 10. Comparison between $\tilde{\xi}_d$ for $D = \tilde{D}_A$ (red curves) and $D = \tilde{D}_S$ (blue curves) for the MNIST problem. The first column shows the AIS-estimated logL of the data, while the rest of the columns show $\tilde{\xi}_d$ for $d = 0, 5, 10$ and 20, respectively.

TABLE II

Optimal AIS-Estimated Stopping Point, and $D = \tilde{D}_S$ and $D = \tilde{D}_A$ Predictions As a Function of the Distance d, for Several Large-Sized Problems Also Used in [35]–[37]. LR and logL Stand for Learning Rate and Log-Likelihood, Respectively. Epochs and Log-Likelihood of the Optimal Stopping Point Are Reported. Last Row Includes Results for the MNIST Problem

Dataset	LR	Optimal AIS		$\tilde{D}_S$ $d=0$		$\tilde{D}_S$ $d=5$		$\tilde{D}_S$ $d=10$		$\tilde{D}_S$ $d=20$	
		Epoch	logL	Epoch	logL	Epoch	logL	Epoch	logL	Epoch	logL
Adult-a5a	0.01	976	-13.67	1000	-13.88	880	-13.90	976	-13.67	998	-14.07
Caltech101	0.0005	158	-157.91	260	-188.36	354	-209.09	995	-333.74	1000	-328.05
Connect-4	0.001	968	-13.77	356	-14.61	992	-13.93	992	-13.93	1000	-13.86
DNA	0.01	998	-62.32	056	-80.70	987	-62.34	992	-62.48	991	-62.48
Mushrooms	0.001	997	-13.41	999	-13.71	1000	-13.58	1000	-13.58	1000	-13.58
NIPS-0-12	0.05	568	-83.95	094	-128.68	184	-98.96	325	-88.17	993	-85.87
OCR-Letter	0.01	086	-41.80	051	-42.01	185	-42.73	492	-44.25	913	-46.06
RCV1	0.01	059	-52.21	050	-52.82	053	-52.77	051	-52.85	097	-54.64
Web-w6a	0.001	945	-28.07	967	-28.52	971	-28.47	997	-28.32	998	-28.60
MNIST	0.0001	357	-125.73	201	-127.48	266	-126.26	319	-125.85	424	-126.44
Dataset	LR	Optimal AIS		$\tilde{D}_A$ $d=0$		$\tilde{D}_A$ $d=5$		$\tilde{D}_A$ $d=10$		$\tilde{D}_A$ $d=20$	
		Epoch	logL	Epoch	logL	Epoch	logL	Epoch	logL	Epoch	logL
Adult-a5a	0.01	976	-13.67	1000	-13.88	1000	-13.88	1000	-13.88	1000	-13.88
Caltech101	0.0005	158	-157.91	260	-188.36	262	-189.31	256	-187.92	316	-200.47
Connect-4	0.001	968	-13.77	356	-14.61	388	-14.50	356	-14.61	335	-14.70
DNA	0.01	998	-62.32	056	-80.70	062	-79.83	056	-80.70	051	-81.50
Mushrooms	0.001	997	-13.41	999	-13.71	999	-13.71	999	-13.71	999	-13.71
NIPS-0-12	0.05	568	-83.95	094	-128.68	130	-112.30	111	-119.38	104	-123.05
OCR-Letter	0.01	086	-41.80	051	-42.01	051	-42.01	064	-41.87	061	-42.11
RCV1	0.01	059	-52.21	050	-52.82	055	-52.60	053	-52.77	054	-52.69
Web-w6a	0.001	945	-28.07	967	-28.52	970	-28.44	969	-28.48	972	-28.42
MNIST	0.0001	357	-125.73	201	-127.48	201	-127.48	242	-126.60	242	-126.60

Optimal AIS-estimated stopping point, and $D = \tilde{D}_S$ and $D = \tilde{D}_A$ predictions as a function of the distance d, for several large-sized problems also used in [35]–[37]. LR and logL stand for learning rate and log-likelihood, respectively. Epochs and log-likelihood of the optimal stopping point are reported. Last row includes results for the MNIST problem.

of the results is very similar to the CD_1 case, thus stressing the robustness of the criterion.

As a final remark, we note that for the BS problem the trained RBM stopped using the proposed criterion is able to qualitatively generate samples similar to those in the training set. We show in Fig. 8 the complete training set (two upper rows) and the same number of generated samples (two lower rows) obtained from the RBM trained with CD_1 and stopped after 5000 epochs, around the maximum shown by $\tilde{\xi}_{d=1}$, which approximately coincides with the optimal value of the logL. It is important to realize that, ultimately, the quality of the model is a direct measure of the quality of CD_1 learning, and that the model used to generate the plots is the one with largest $\tilde{\xi}_d$, which is quite close to the one with largest likelihood.

B. Persistent CD

PCD is a well known and cheap alternative to plain CD that helps improving learning [18], [19]. We have tested our stopping criteria in the same setting of the previous sections using PCD, leading to similar results. This can be justified from the fact that it is known that under certain conditions PCD also degenerates [20], [21] as much as CD does, due to probability concentration in a handful of states. Therefore, a measure that qualitatively captures the logL behavior for CD is expected to work also for PCD.

This is illustrated in Fig. 9, where $\tilde{\xi}_d$ and ξ_d are shown for the LSE problem learned with PCD. As in the previous cases, the evolution of the proposed estimators along the simulation qualitatively resembles that of the ground truth, and thus the stopping criteria detect a reasonably good stopping point.

C. MNIST Data set

The MNIST data set is a well known benchmark problem corresponding to 28×28 binarized images of hand-written digits from a huge space of 2^{764} possible states.[1] In this case, an RBM with 764 visible and 500 hidden units has been employed. The calculation of the reference logL of the training set has been approximated with the AIS technique, for a total of 100 running chains of 1000 β_k each [25]. These values have been chosen for efficiency reasons and have been checked to provide reasonable estimations of the likelihood compared with results obtained with larger values. The RBM was run for a total of 1000 epochs, and the LR and momentum chosen for the following figures are 0.0001 and 0.8, respectively. No weight decay has been used, though exploration with nonzero values showed very little influence on the final results.

The left panel in Fig. 10 shows the AIS-estimated logL of the training set. The other plots depict $\tilde{\xi}_d$ for $d = 0, 5, 10, 20$ corresponding to $D = \tilde{D}_A$ and $D = \tilde{D}_S$. Notice that only the incomplete neighborhood estimator has been evaluated as the total amount of neighbors of the training set at a given distance is exceedingly large to be of practical use. Remarkably, our measure works equally well in all these cases, thus showing that the proposed estimator is in principle able to capture the leading features of the likelihood even in large problems. Notice that in this case already $d = 0$ provides a good estimation of the stopping point.

One could think that the AIS estimated likelihood would provide a equally good stopping point. While this is true, it is worth noticing that, with the standard parameters used in real calculations based on AIS, the computational costs would increase by a few orders of magnitude. For example, with the parameters in [34] where a total of 5000 running chains with 10^5 β_k's, the computational cost would be approximately 10^4 times larger.

Additionally, and in order to compare with exact results as in [25], we have tested our stopping criterion on the MNIST problem with 25 hidden units. Notice that in this case the exact partition function is evaluated, not estimated using AIS or any other approximation. Best results are achieved with a LR $\epsilon = 10^{-3}$, where the stopping point according to the exact likelihood is located at the epoch ~ 100. In contrast, our criteria for $D = \tilde{D}_S$ and $D = \tilde{D}_A$ give similar results and suggest to stop at epoch ~ 120.

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

D. Other Large Problems

We have extended our analysis to other large-sized problems of relatively high dimensionality: Adult-a5a, Connect-4, DNA, Mushrooms, NIPS-0-12, OCR-Letter, RCV1, Web-w6a (used in [35] and [36], for example) and the Caltech101 Silhouettes data set (used in [37], for example). The data sets can be downloaded from http://www.cs.toronto.edu/~larocheh/code/nade.tgz and http://www.cs.ubc.ca/~bmarlin/data. We have used the same topology as in the references. In each case, we have performed ten runs and averaged the resulting curves, as in the MNIST problem. Table II shows the results (stopping epoch and AIS-estimated logL at that epoch) obtained for $D = \tilde{D}_S$ and $D = \tilde{D}_A$ for several distances d. Notice that the results reported in the table are the most representative of the general behavior, obtained after many runs with different LRs. As it can be seen, both criteria work well in most cases. When the likelihood achieves a maximum, it is usually detected by both criteria, yielding a good estimation of the optimal likelihood. Still, in some cases the criterion fails to detect a good stopping point, as happens with the Caltech101 and the NIPS-0-12. However, even in these cases, valuable information is recovered, as both criteria detect that the likelihood achieves a maximum at some point and afterwards degenerates, which suggests to start the learning process again with a lower LR. When the best likelihood is achieved around the last epoch of the training, our criteria usually indicate that one should stop near the end, though in some cases $D = \tilde{D}_S$ performs better (DNA, Connect-4). Overall, our criteria successfully detect a good stopping point that can be taken as the end of the learning process.

V. Conclusion

In this paper, we have introduced the contribution of neighboring points to the training set to build a stopping criterion for learning in CD. We have shown that not only the training set but also the neighboring states contain valuable information that can be used to follow the evolution of the network along training.

Based on the fact that learning tries to increase the contribution of the relevant states while decreasing the contribution of the rest, continuity, and smoothness of the energy function assigns more probability to states close to the training data. This is the key idea behind the proposed stopping criterion. In fact, two different but related estimators (depending on the number of states used to compute them) have been proposed and tested experimentally. The first one includes all states close to the training set, while the second one takes only a fraction of these states as small as the size of the training set. The first estimator is robust but may require the use of a forbiddingly large amount of states, while the second one is always tractable and captures most of the features of the first one, thus providing a suitable stopping learning criterion. This second estimator has been shown to work equally well in the MNIST and other large data sets, where an exact computation of the logL is not possible. Additionally, the main idea of proximity to the training set will be explored in other aspects related to learning in the future work. Furthermore, we could try different metrics to measure proximity between neighboring states.

References

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362.

[2] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[3] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[4] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[5] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[6] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," *J. Mach. Learn. Res.*, vol. 10, no. 1, pp. 1–40, Jan. 2009.

[7] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 609–616.

[8] Q. V. Le *et al.*, "Building high-level features using large scale unsupervised learning," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 81–88.

[9] R. Sarikaya, G. E. Hinton, and A. Deoras, "Application of deep belief networks for natural language understanding," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 4, pp. 778–784, Apr. 2014.

[10] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA, USA: MIT Press, 1986, pp. 194–281.

[11] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 6, pp. 721–741, Nov. 1984.

[12] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, 2002.

[13] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing*, vol. 19. Cambridge, MA, USA: MIT Press, 2007, pp. 153–160.

[14] Y. Bengio and O. Delalleau, "Justifying and generalizing contrastive divergence," *Neural Comput.*, vol. 21, no. 6, pp. 1601–1621, May 2009.

[15] M. A. Carreira-Perpinan and G. E. Hinton, "On contrastive divergence learning," in *Proc. Int. Workshop Artif. Intell. Statist.*, 2005, pp. 33–40.

[16] A. Yuille, "The convergence of contrastive divergence," in *Advances in Neural Information Processing Systems*, vol. 17. Cambridge, MA, USA: MIT Press, 2005, pp. 1593–1600.

[17] D. J. C. MacKay, "Failures of the one-step learning algorithm," Cavendish Lab., Cambridge, U.K., Tech. Rep., 2001. [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/gbm.pdf

[18] T. Tieleman, "Training restricted Boltzmann machines using approximations to the likelihood gradient," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1064–1071.

[19] T. Tieleman and G. E. Hinton, "Using fast weights to improve persistent contrastive divergence," in *Proc. 26th Int. Conf. Mach. Learn.*, 2009, pp. 1033–1040.

[20] H. Schulz, A. Müller, and S. Behnke, "Investigating convergence of restricted boltzmann machine learning," in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2010.

[21] A. Fischer and C. Igel, "Empirical analysis of the divergence of Gibbs sampling based learning algorithms for restricted Boltzmann machines," in *Proc. Int. Conf. Artif. Neural Netw. (ICANN)*, vol. 3. 2010, pp. 208–217.

[22] G. Desjardins, A. Courville, Y. Bengio, P. Vincent, and O. Delalleau, "Parallel tempering for training of restricted Boltzmann machines," in *Proc. 13th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2010, pp. 145–152.

[23] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks Trade*. Berlin, Germany: Springer, 2012, pp. 599–619.

[24] R. M. Neal, "Annealed importance sampling," Dept. Statist., Univ. Toronto, Toronto, ON, Canada, Tech. Rep. 9805, 1998.

[25] R. Salakhutdinov and I. Murray, "On the quantitative analysis of deep belief networks," in *Proc. Int. Conf. Mach. Learn.*, 2008, pp. 872–879.

[26] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines. A Stochastic Approach to Combinatorial Optimization and Neural Computing*. Hoboken, NJ, USA: Wiley, 1990.

[27] A. R. Sankar and V. N. Balasubramanian, "Similarity-based contrastive divergence methods for energy-based deep learning models," in *Proc. JMLR, Workshop Conf.*, vol. 45. 2015, pp. 391–406.

[28] X. Ma and X. Wang, "Average contrastive divergence for training restricted Boltzmann machines," *Entropy*, vol. 18, no. 1, p. 35, 2016. [Online]. Available: http://www.mdpi.com/1099-4300/18/1/35

[29] M. Gabrie, E. W. Tramel, and F. Krzakala, "Training restricted Boltzmann machine via the thouless-anderson-palmer free energy," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 640–648.

[30] A. Fischer and C. Igel, "Training restricted Boltzmann machines: An introduction," *Pattern Recognit.*, vol. 47, no. 1, pp. 25–39, 2014.

[31] L. Li, J. Lv, and Z. Yi, "A non-negative representation learning algorithm for selecting neighbors," *Mach. Learn.*, vol. 102, no. 2, pp. 133–153, 2016.

[32] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the L2-graph for robust subspace learning and subspace clustering," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1053–1066, 2017.

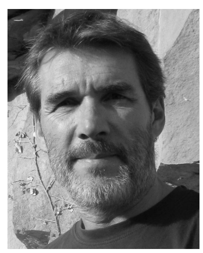
[33] P. Bühlmann and S. Van De Geer, *Statistics for High-dimensional Data: Methods, Theory and Applications*. Berlin, Germany: Springer, 2011.

[34] M.-A. Côté and H. Larochelle, "An infinite restricted Boltzmann machine," *Neural Comput.*, vol. 28, no. 7, pp. 1265–1288, 2016.

[35] H. Larochelle, Y. Bengio, and J. Turian, "Tractable multivariate binary density estimation and the restricted Boltzmann forest," *Neural Comput.*, vol. 22, no. 9, pp. 2285–2307, 2010.

[36] H. Larochelle and I. Murray, "The neural autoregressive distribution estimator," in *Proc. Int. Workshop Artif. Int. Statist.*, 2011, pp. 29–37.

[37] B. M. Marlin, K. Swersky, B. Chen, and N. de Freitas, "Inductive principles for restricted Boltzmann machine learning," in *Proc. Int. Workshop Artif. Int. Statist.*, 2010, pp. 509–516.

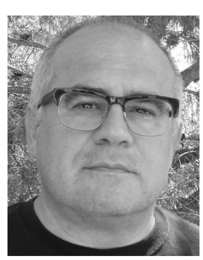


Enrique Romero Merino received the B.Sc. degree in mathematics from the Universitat Autònoma de Barcelona, Barcelona, Spain, in 1989, and the B.Sc. and Ph.D. degrees in computer science from the Universitat Politècnica de Catalunya (UPC), Barcelona, in 1994 and 2004, respectively.

In 1996, he joined the Department of Computer Science, UPC, where he is currently an Associate Professor. His current research interests include pattern recognition, support vector machines, neural networks, and deep learning.

Ferran Mazzanti Castrillejo received the B.Sc. and Ph.D. degrees in physics from the Universitat de Barcelona, Barcelona, Spain, in 1991 and 1997, respectively.

In 1992, he joined Electronics Department, Enginyeria I Arquitectura La Salle, Barcelona, as an Associate Professor. In 2006, he joined the Physics Department, Universitat Politècnica de Catalunya, Barcelona. His current research interests include quantum many body problems at zero and finite temperature, computational simulation of quantum systems, Boltzmann machine neural networks, and deep learning.



Jordi Delgado Pin received the B.Sc. and Ph.D. degrees in computer science from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 1992 and 1997, respectively.

He is currently an Associate Professor with Computer Science Department, UPC, where he teaches the programming project and advanced programming concepts courses at the Barcelona School of Informatics. His current research interests include neural networks, machine learning, and programming languages.