Learning strategies for the maximally stable diluted binary perceptron

D. Malzahn
Institut für Theoretische Physik, Otto-von-Guericke-Universität, Postfach 4120, D-39016 Magdeburg, Germany
(Received 25 October 1999)

I show analytically that an optimally chosen continuous precursor **J** in the hypercube is highly correlated to the maximally stable diluted binary perceptron which solves the same storage problem. **J** allows the construction of a diluted binary perceptron **D** by a simple rule. Performing simulations for perceptrons of size $N=100$ I demonstrate that **D** is highly stable and can be improved in an efficient manner by partial enumeration thereby incorporating information from the precursor components. The precursor highlights the vector components on which partial enumeration improves the stability of the vector most efficiently. Moreover, it discriminates for each vector component i at least one of the three possible values $D_i=\{-1,0,1\}$ as being extremely unlikely.

PACS number(s): 05.20.−y, 87.18.Sn

I. INTRODUCTION

Learning algorithms are needed to train neural networks for the tasks they have to perform. In many cases, learning can be formulated as an optimization problem [1,2] in which the minimum of a cost function defines the optimal synaptic weights. Several efficient minimization algorithms exist for networks with continuous weights and cost functions with a single minimum. In contrast, for networks with discrete weights, the same cost functions, defined on the discrete set display a huge number of local minima in which standard minimization methods will get trapped [3,4]. The only known method to find the global minimum is enumeration of all possible weight vectors. But even for the simplest network, the binary perceptron, enumeration can be done in a reasonable time only for a system with about 40 input units [5,6]. For larger systems, the amount of computer time becomes inordinate and a different approach is required.

A number of learning schemes have been proposed for the maximally stable binary perceptron (MSB). The MSB is represented by a binary vector **B** with N components. It stores a set of input-output mappings $\{\boldsymbol{\xi}^\nu \rightarrow \sigma^\nu\}$, $\nu=1,\ldots,\alpha N$, in a robust manner: $\sigma^\nu \Lambda^\nu \geqslant \kappa$ where κ becomes maximal and $\Lambda^\nu = \mathbf{B}\boldsymbol{\xi}^\nu/\sqrt{N}$ denotes the so-called local fields. Among the numerical methods that try to locate the global minimum of an appropriate cost function, the most successful so far are the genetic algorithm of Köhler and simulated annealing of Horner [3]. Their performance is quite good for perceptrons with up to 65 weights but rapidly deteriorates when the number of weights exceeds 100. An alternative and very attractive approach tries to draw some advantage from the fact that efficient algorithms exist for the learning problem of the continuous perceptron [7–9]. By continuous optimization one selects a unique perceptron vector **J** that solves the same storage problem as the MSB and is highly correlated to it. The continuous precursor **J** is used to predict an important fraction of binary components whereas all uncertain components of **B** must be enumerated. The approach tries to generate the optimal binary vector or a good approximation for it while simultaneously reducing the set of enumerated components.

As a general finding, strong precursor weights predict with a high probability the correct sign for the corresponding binary component. However, precursor weights of small absolute value give unreliable predictions. A principal difficulty is that in contrast to the continuous precursor, the MSB can not differentiate between weak and strong components but must match to any precursor component a strong (binary) weight.

In this paper, I will consider the learning problem for the maximally stable diluted binary perceptron (MSDB) in which weights can take on the three values -1, 0, or 1. The focus of the paper is to study the performance of continuous precursors for the prediction of MSDB weights. On first sight one can expect that the correlation to its respective optimal continuous precursor is much stronger for the MSDB than for the MSB. In particular, the addition of the zero weight offers a natural match to weak precursor components. Simultaneously, exact determination of the most stable vector is a much harder problem for the MSDB than for the MSB. The search space of combinatorially possible vectors is 3^N rather than 2^N with the consequence that full enumeration of all components becomes even more time consuming. It has been carried out so far only for perceptron sizes $N \leqslant 16$ [10]. The increased complexity of the problem underlines the value of a continuous precursor. Finally it should be noted, that the addition of the zero weight which merely eliminates some of the connections, brings about a substantial increase in storage capacity [11].

The present paper consists of two parts. The first is analytic theory. I calculate the conditional probability $p(\mathbf{D}|\mathbf{J})$. It is the key quantity for judging the predictive power of any precursor **J** with respect to the coupling vector **D** of the MSDB. I will consider the pattern entries to be statistically independent random numbers. Hence, $p(\mathbf{D}|\mathbf{J})$ reduces to $p(D|J)$ which compares corresponding vector components D and J. $p(D|J)$ allows me to set up rules for the construction of a diluted binary vector of high stability. It also provides suggestions for different partial enumeration schemes. From the set of saddle point equations that determine the order parameters one obtains the cosine of the angle between **J** and **D**. I use this order parameter to evaluate different precursors **J** with respect to their similarity in direction to **D** and show that a nearly optimal precursor can be obtained by convex minimization in the hypercube [9]. In a second purely nu-

merical part of the paper, I test the quality of different learning strategies for the MSDB by simulations for perceptrons of size $N=100$. The quality measure for the different learning strategies is the average stability of the generated diluted binary vectors or, equivalently, the obtained storage capacity. In the final Section I summarize my results.

II. THEORY: CORRELATIONS BETWEEN A CONTINUOUS PRECURSOR AND THE MAXIMALLY STABLE DILUTED BINARY PERCEPTRON

I consider the combined system of a diluted binary perceptron $\mathbf{D}=(D_1,D_2,\ldots,D_N)$ and a continuous perceptron $\mathbf{J}=(J_1,J_2,\ldots,J_N)$ which are both trained by their individual learning rules to store the same set of patterns $\{\boldsymbol{\xi}^\nu,\sigma^\nu\}$, $\nu=1,\ldots,\alpha N$. The components of the pattern vectors $\boldsymbol{\xi}^\nu$ are random Gaussian numbers with zero mean and unit variance. Without loss of generality one can set $\sigma^\nu=+1$. Following the general approach of Wong, Rau, and Sherrington [12] I consider the joint probability distribution

$$p(D,J)=\lim_{\kappa\to\max}\left\langle\left\langle\frac{1}{\mathcal{Z}}\int d\mu(\mathbf{J})\prod_{\nu=1}^{\alpha N}h(\lambda^\nu)\times\sum_{\mathbf{D}}\prod_{\nu=1}^{\alpha N}\Theta(\Lambda^\nu-\kappa)\,\delta_{D_1,D}\,\delta(J_1-J)\right\rangle\right\rangle. \quad (1)$$

Note, that all pairs of corresponding vector components (D_l,J_l), $l=1,\ldots,N$, have the same joint probability distribution Eq. (1). The average $\langle\langle\cdots\rangle\rangle$ is taken over the pattern set $\{\boldsymbol{\xi}^\nu\}$ and $\mathcal{Z}$ is the partition function

$$\mathcal{Z}=\int d\mu(\mathbf{J})\prod_{\nu=1}^{\alpha N}h(\lambda^\nu)\sum_{\mathbf{D}}\prod_{\nu}^{\alpha N}\Theta(\Lambda^\nu-\kappa) \quad (2)$$

in the joint weight space. I introduce the local fields λ^ν,Λ^ν of $\mathbf{J}$ and $\mathbf{D}$

$$\lambda^\nu=\frac{\mathbf{J}\boldsymbol{\xi}^\nu}{\sqrt{N}},\quad \Lambda^\nu=\frac{\mathbf{D}\boldsymbol{\xi}^\nu}{\sqrt{N}}. \quad (3)$$

Given any random but fixed pattern set, Eq. (1) specifies exactly two perceptrons $\mathbf{D}$ and $\mathbf{J}$ by learning rules. $\mathbf{D}$ is the diluted binary vector

$$D_i=-1,0,1 \quad (4)$$

of maximal stability κ^{db},

$$\Lambda^\nu\geqslant\kappa^{db}. \quad (5)$$

The restriction (4) is enforced in Eq. (1) by the summation $\Sigma_{\mathbf{D}}$. For the continuous perceptron I consider learning rules which have the measure $p(\mathbf{J}|\boldsymbol{\xi}^1,\ldots,\boldsymbol{\xi}^{\alpha N})=\Pi_\nu h(\lambda^\nu)$ and result in a single solution. The constraint on $\mathbf{J}$ is reflected in Eq. (1) by the integration measure $d\mu(\mathbf{J})$. The remainder of this section introduces the order parameters and summarizes the central results of the replica calculation. Full specifications of the precursor $\mathbf{J}$ as well as the result for the conditional probability for a nearly optimal precursor are given below in two subsections.

The pattern average in Eq. (1) can be performed using the replica trick $\mathcal{Z}^{-1}=\lim_{n\to0}\mathcal{Z}^{n-1}$ provided the local fields λ^ν, Λ^ν serve as independent integration variables. The natural order parameters are

$$Q_{ab}=\frac{\mathbf{D}^a\mathbf{D}^b}{N},\quad q_{ab}=\frac{\mathbf{J}^a\mathbf{J}^b}{N},\quad r_{ab}=\frac{\mathbf{D}^a\mathbf{J}^b}{N}, \quad (6)$$

where $a,b=1,\ldots,n$ are replica indices and $\mathbf{D}$ and $\mathbf{J}$ from Eq. (1) are denoted by $\mathbf{D}^1$ and $\mathbf{J}^1$, respectively. Introducing Q_{ab}, q_{ab}, r_{ab} as independent integration variables and using the Fourier representation of the δ function gives rise to their conjugate variables $\hat{Q}_{ab},\hat{q}_{ab},\hat{r}_{ab}$. The order parameters can be written in compact form as elements of the $n\times n$ matrices $\mathbf{Q},\mathbf{q},\mathbf{r}$ and $\hat{\mathbf{Q}},\hat{\mathbf{q}},\hat{\mathbf{r}}$ (see Appendix A for details). This yields

$$\begin{aligned}p(D,J)=&\lim_{\substack{n\to0\\k\to\max}}\int\frac{1}{2^n}\prod_{a\leqslant b}\frac{dq_{ab}\,d\hat{q}_{ab}}{2\pi/N}\\&\times\int\frac{1}{2^n}\prod_{a\leqslant b}\frac{dQ_{ab}d\hat{Q}_{ab}}{2\pi/N}\int\prod_{a,b}\frac{dr_{ab}d\hat{r}_{ab}}{2\pi/N}\\&\times\exp\left(N\left[\frac{1}{2}(\mathbf{q}\hat{\mathbf{q}}+\mathbf{Q}\hat{\mathbf{Q}})-\mathbf{r}\hat{\mathbf{r}}+\alpha\ln G_1(\mathbf{q},\mathbf{Q},\mathbf{r})\right.\right.\\&\left.\left.+\ln G_2(\hat{\mathbf{q}},\hat{\mathbf{Q}},\hat{\mathbf{r}})\right]\right)p(D,J|\hat{\mathbf{q}},\hat{\mathbf{Q}},\hat{\mathbf{r}}). \quad (7)\end{aligned}$$

Equation (7) integrates over all order parameters. The factor $\exp(N[\cdots])$ gives the order parameter density. In the limit $N\to\infty$ it becomes sharply peaked and is dominated by the saddle point values of the order parameters. G_1 and G_2 are given by Eqs. (A1),(A2). The density

$$\begin{aligned}&p(D,J|\hat{\mathbf{q}},\hat{\mathbf{Q}},\hat{\mathbf{r}})\\&=\frac{1}{G_2(\hat{\mathbf{q}},\hat{\mathbf{Q}},\hat{\mathbf{r}})}\times\int\prod_{a=1}^{n}d\mu(J_1^a)\sum_{D_1^a}\delta_{D_1^1,D}\,\delta(J_1^1-J)\\&\times\exp\left(-\frac{1}{2}(\vec{J}_1\hat{\mathbf{q}}\vec{J}_1+\vec{D}_1\hat{\mathbf{Q}}\vec{D}_1)+\vec{J}_1\hat{\mathbf{r}}\vec{D}_1\right), \quad (8)\end{aligned}$$

is the density $p(D,J)$ provided one inserts the correct saddle point values for $\hat{\mathbf{q}},\hat{\mathbf{Q}},\hat{\mathbf{r}}$. The notation $\vec{J}_1=(J_1^1,\ldots,J_1^n)$ and $\vec{D}_1=(D_1^1,\ldots,D_1^n)$ subsumes all n replicas of the first component of $\mathbf{J}$ and $\mathbf{D}$. I assume a replica symmetric saddle point

$$Q_{ab}\sim\begin{cases}Q_0 & a=b\\ Q & a\neq b,\end{cases}\quad q_{ab}\sim\begin{cases}q_0 & a=b\\ q & a\neq b,\end{cases}\quad r_{ab}\sim r. \quad (9)$$

Note, that $Q_0\leqslant1$ due to the dilution of $\mathbf{D}$. I will consider constraints on $\mathbf{J}$ that impose $q_0\leqslant1$ and focus on learning rules, which yield a unique solution, $q\to q_0$. For $\hat{Q}_{ab}$, $\hat{q}_{ab}$, $\hat{r}_{ab}$ holds the analog to Eq. (9). All saddle point equations that determine the order parameters are given in Appendix A.

Before I proceed some words on the characterization of the MSDB are in order. Iwanski *et al.* [10] and Krauth and

Mézard [13] showed that the correct solution can be either obtained by a one step replica symmetry broken ansatz or, alternatively, by the replica symmetric saddle point equations supplemented with the additional constraint of a vanishing replica symmetric entropy. The zero entropy condition determines the maximum stability $\kappa^{db}(\alpha)$. Solving the saddle point equations for the MSDB reveals a peculiarity: For all values $\alpha>0$ one finds $Q<Q_0$. This indicates that many different weight vectors satisfy the conditions (5) even at maximum stability. Since it is impossible to distinguish the individual weight vectors, all theoretical results relate to the average $\langle \mathbf{D} \rangle$ over this ensemble of maximally stable diluted binary vectors.

Remarkably, the saddle point equations (A12) and (A13) contain the quantities

$$\gamma = \frac{r}{\sqrt{Qq}}, \quad \hat{\gamma} = \frac{\hat{r}}{\sqrt{\hat{Q}\hat{q}}} \tag{10}$$

and $|\gamma|, |\hat{\gamma}| \leqslant 1$. γ can be interpreted geometrically: Correcting the overlap r for the reduced lengths of $\langle \mathbf{D} \rangle$ and $\mathbf{J}$ yields the cosine of the angle between both vectors. The order parameter $\hat{\gamma}$ is crucial for the distribution function Eq. (8): Within replica symmetry, the integrand of Eq. (8) factorizes into two contributions. They depend solely on the saddle point values of the order parameters of $\mathbf{J}$ and $\mathbf{D}$, respectively. However, both contributions are coupled by $\hat{\gamma}$.

A. A nearly optimal continuous precursor

In this subsection, I will quantify the impact of the constraint and of the learning rule on the quality of the precursor. To the MSDB only a limited set of directions are available. As Bouten *et al.* [9] pointed out, it is advantageous to incorporate this information in the constraint on $\mathbf{J}$ while simultaneously the convexity of the defined vector space shall be preserved. The latter is important to ensure that the considered learning rules yield a unique solution. I consider the usual spherical constraint $\mathbf{J}^2 = N$, which is isotropic, in contrast to a hypercube constraint

$$-1 \leqslant J_i \leqslant 1 \tag{11}$$

which favors binary directions. Figure 1 characterizes four perceptrons which obey the condition of maximal stability under different constraints: the MSB with pure binary weights (thin curve), the MSDB with binary weights and dilution (bold curve), the MSC with continuous weights in the hypercube (dot-dashed curve), the MSN with continuous weights on the hypersphere (dashed curve). The comparison between MSB and MSDB is given to demonstrate briefly the impact of the dilution. Figure 1 shows the stability $\kappa = \min_\nu(\mathbf{V}\boldsymbol{\xi}^\nu/\sqrt{\mathbf{N}})$ as a function of the pattern load α where the coupling vector of the respective perceptron is represented by the symbolic vector $\mathbf{V}$. κ passes through zero when the storage capacity is reached that is $\alpha_c = 2$ for the continuous perceptrons MSN and MSC, $\alpha_c^{db} = 1.17$ for the MSDB and $\alpha_c^b = 0.83$ for the MSB. The stability κ is normalized by the respective reduced length of the coupling vector, for example for the MSDB one has $\kappa_n = \kappa^{db}/\sqrt{Q_0}$.

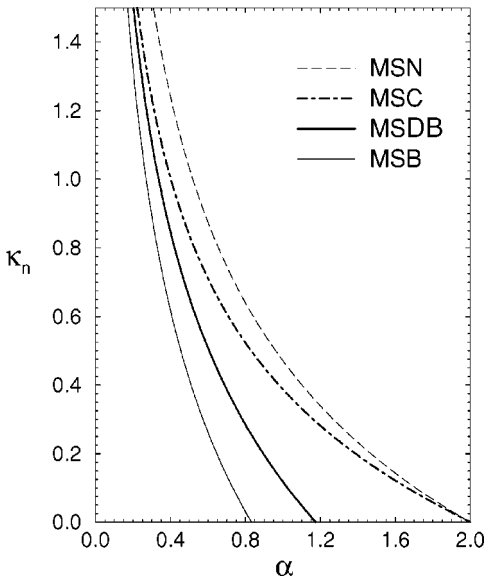

FIG. 1. Maximal stability κ as a function of the pattern load α for four perceptrons with different constraints. From left to right, MSB: binary weights ($\kappa_n = \kappa$), MSDB: binary weights and dilution ($\kappa_n = \kappa/\sqrt{Q_0}$), MSC: continuous weights in the hypercube ($\kappa_n = \kappa/\sqrt{q_0}$), MSN: continuous weights on the hypersphere ($\kappa_n = \kappa$). κ was normalized by the reduced perceptron length.

Due to this normalization, a difference in κ_n characterizes a difference in direction and one can conclude that the MSC and the MSDB are much closer related than the MSDB and the MSN.

Figure 2 shows γ, the cosine of the angle between $\mathbf{J}$ and $\langle \mathbf{D} \rangle$, for different continuous precursors $\mathbf{J}$ as a function of the pattern load α (bold curves). The value of γ is obtained from the saddle point equations (A12),(A13). $\gamma \to 1$ indicates

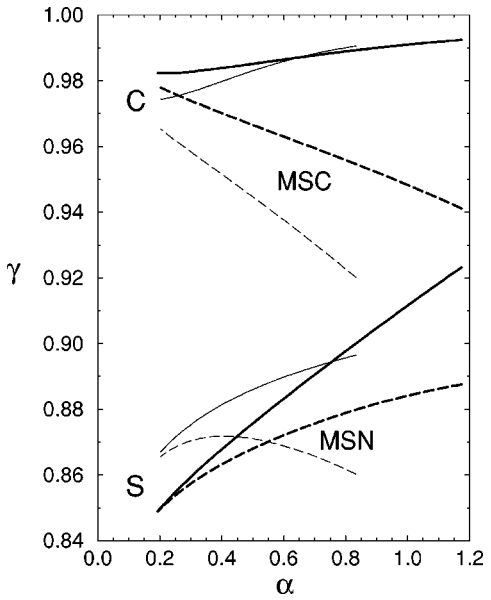

FIG. 2. Similarity in direction $\gamma = \cos(\langle \mathbf{D} \rangle, \mathbf{J})$ as a function of the pattern load α. Different precursors $\mathbf{J}$ compared to the MSDB (bold lines) and to the MSB (thin lines). Different constraint for $\mathbf{J}$: hypercube (C) versus hypersphere (S). Different learning rule for $\mathbf{J}$: learning with potential Eq. (12) (solid lines) versus maximum stability condition (dashed lines).

that the precursor becomes optimal and the directions of **J** and $\langle \mathbf{D} \rangle$ coincide. The presented precursors **J** differ with respect to the applied constraint (C = cubic, S = spheric) and the learning rule. The learning rule was either to maximize the stability (MSC and MSN) or to minimize a cost function $E(\mathbf{J}) = \Sigma_\nu V(\lambda^\nu)$ defined by the convex potential

$$V(\lambda^\nu) = \begin{cases} 1/(\lambda^\nu - \kappa^{db}) & \text{if } \lambda^\nu > \kappa^{db} \\ \infty & \text{otherwise.} \end{cases} \tag{12}$$

The potential (12) is constructed in analogy to Ref. [8].

As expected, Fig. 2 demonstrates a strong dependence of the quality of the precursor on the applied constraint. The hypercube restricted MSC is superior to the hypersphere restricted MSN. The explanation is simple. In the hypercube, vector **J** may differ in length and a loss in quality of the direction of **J** can be compensated by a gain in length of **J** by shifting it slightly towards the nearest cube edge. Thus, the hypercube favors those orientations that constitute the binary subspace. A somewhat smaller but nevertheless still considerable improvement is brought about by the usage of learning rule Eq. (12) (solid lines) in comparison to maximal stability learning (dashed lines). The MSC realizes a higher stability than the MSDB and is located in the Gardner volume [14] of continuous, hypercube restricted vectors **J** with stability $\kappa \geqslant \kappa^{db}$ were κ^{db} denotes the theoretical stability of the MSDB. The sole information available about the position of the MSDB is that it lies for all pattern loads α at the boundary of this Gardner volume. Given this information, the optimal precursor to the MSDB would be the center of mass of the Gardner volume [16]. The potential Eq. (12) is characterized by a strong repulsion away from the boundary of the Gardner volume with stability $\kappa \geqslant \kappa^{db}$ and pushes the minimizing vector of the cost function $E(\mathbf{J}) = \Sigma_\nu V(\lambda^\nu)$ toward its center of mass [15]. Within the class of precursors obtainable by minimization of a cost function of the type $E(\mathbf{J}) = \Sigma_\nu V(\lambda^\nu)$ the simple potential Eq. (12) provides a quasioptimal solution and gives, as Fig. 2 shows, a nearly optimal precursor to the MSDB.

So far I discussed only the bold curves in Fig. 2. They compare different precursors **J** to the MSDB. To relate my results to previous work [8,9], thin curves show the performance of the four precursors on the MSB. Note that learning rule Eq. (12) operates in this case with the stability κ^b of the MSB as an input parameter. Bold and thin curves end at the storage capacity of the MSDB and the MSB, respectively. According to Fig. 2, hypercube precursors approximate the MSDB better than the MSB in a wide range of α values. However, the difference in quality becomes rather small for the nearly optimal precursor. Figure 2 shows that for the nearly optimal hypercube precursor γ is almost independent of α.

B. The conditional probability $p(D|J)$ for hypercube precursors

In the previous subsection I have shown that a continuous precursor in the hypercube which minimizes a cost function $E(\mathbf{J}) = \Sigma_\nu V(\lambda^\nu)$ with the convex potential Eq. (12) on average almost coincides in direction with the coupling vector of the MSDB. It is very instructive to compare the distribution of components $p(D)$ and $p(J)$ of the MSDB and the nearly optimal hypercube precursor. **D** is a diluted binary vector, hence

TABLE I. Theory: Percentage of binary components in the nearly optimal hypercube precursor **J** and the maximally stable diluted binary perceptron **D** for different pattern loads α and infinite perceptron size N.

α	**J**	**D**
0	100%	100%
0.20	81.7%	88.1%
0.30	74.5%	84.6%
0.50	62.4%	78.7%
0.80	48.0%	71.3%
1.16	34.0%	63.4%

$$p(D) = (1 - Q_0)\,\delta(D) + \frac{Q_0}{2}[\,\delta(D+1) + \delta(D-1)]. \tag{13}$$

In the limit $\alpha \to 0$ one finds $Q_0 = 1$ and the MSDB becomes a purely binary vector. At the storage capacity $\alpha_c^{db} = 1.17$ one has $Q_0 = 0.63$ and about 37% of the MSDB weights are diluted. **J** has an α dependent fraction of binary components due to the cubic constraint [9]

$$p(J) = \frac{e^{-J^2/2s^2}}{s\sqrt{2\pi}}\Theta(1 - |J|) + H\left(\frac{1}{s}\right)[\,\delta(J-1) + \delta(J+1)], \tag{14}$$

where $H(u) = \int_u^\infty dz\, \exp(-z^2/2)/\sqrt{2\pi}$. The order parameter s starts at $\alpha = 0$ with $s = \infty$ and drops to $s = 0$ at $\alpha = 2$. Consequently, the distribution of **J** components changes gradually from all being binary to Gaussian. At the storage capacity of the MSDB $\alpha_c = 1.17$ I find $s \sim 1.04$ and about 33.4% of components of **J** take on the value ± 1. A short summary is given by Table I. It lists the percentage of binary components for **D** and **J** at different pattern loads α.

For hypercube precursors, calculation of $p(D,J)$ yields an expression which factorizes out the separate terms of the distribution $p(J)$ of the hypercube weights Eq. (14). This makes it simple to read off the conditional probabilities $p(D|J)$:

$$p(D|J) = \int_{-\infty}^{\infty} Du\; f\left(u\sqrt{1-\hat{\gamma}^2} + \hat{\gamma}\frac{J}{s}, D\right) \text{ for } -1 < J < 1, \tag{15}$$

$$p(D|J) = \int_{-\infty}^{\infty} Du\; \frac{f(u,D)}{H(1/s)} H\left(\frac{1/s - \hat{\gamma}uJ}{\sqrt{1-\hat{\gamma}^2}}\right) \text{ for } J = \pm 1 \tag{16}$$

with $Du = \exp(-u^2/2)du/\sqrt{2\pi}$ and $f(u,D)$ as a shorthand notation for

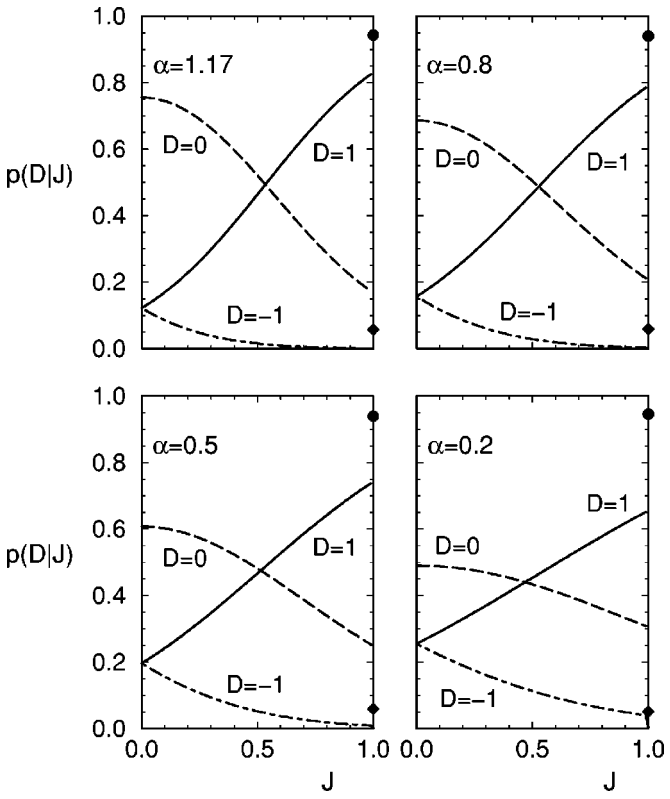

FIG. 3. The conditional probability $p(D|J)$ compares the nearly optimal hypercube precursor **J** to the coupling vector **D** of the MSDB by the value of corresponding components. Shown are $p(D=1|J)$ (solid line and circle), $p(D=0|J)$ (dashed line and diamond), and $p(D=-1|J)$ (dot-dashed line) at four different pattern loads α.

$$f(u,D)=\frac{\exp\left[-\frac{1}{2}(\hat{Q}_0+\hat{Q})D^2+uD\sqrt{\hat{Q}}\right]}{1+2\exp\left[-\frac{1}{2}(\hat{Q}_0+\hat{Q})\right]\cosh(u\sqrt{\hat{Q}})}. \quad (17)$$

These conditional probabilities are a direct measure for gauging the ability of the precursor components to predict the weights of the MSDB correctly. Figure 3 shows $p(D|J)$ for the nearly optimal hypercube precursor as a function of J for different values of α. The symmetry $p(D|J)=p(-D|-J)$ allows me to restrict the displayed range of J to the positive interval $[0,1]$. To a particular value $J>0$ corresponds the value $D=-1$ with probability $p(-1|J)$ (dot dashed), the value $D=0$ with probability $p(0|J)$ (dashed) and the value $D=1$ with probability $p(1|J)$ (solid). A binary $J=1$ corresponds either to $D=1$ (circle) or to a diluted component $D=0$ (diamond). The probability $p(-1|1)$ is very close to zero and therefore not shown in the figures.

Figure 3 shows that binary precursor components are highly reliable. They give a correct prediction in about 94% of all cases. The prediction certainty of binary precursor components varies only slightly in the displayed range of α values whereas the percentage of binary components in **J** depends strongly on alpha (see Table I). For very small values of α, precursor components of magnitude $|J_i|<1$ are rare and almost equally distributed. Consequently, as Fig. 3 shows, it is hard to decide which of them must be diluted. The error in sign prediction of binary **D** components is small but not negligible. The overall probability for precursor components that are different from ± 1 increases with increasing pattern load α. Their distribution function evolves into a pronounced Gaussian shape. This behavior of $p(J)$ is reflected in $p(D|J)$. Errors in sign prediction for binary D components are unlikely for $\alpha>0.6$. The distinction between weak (to be diluted) and strong (to be binary) components improves.

III. SIMULATION: NUMERICAL TEST OF DIFFERENT LEARNING RULES

Several interesting features can be noted from Fig. 3 that are useful for setting up a learning strategy for the MSDB. When $|J|$ exceeds a particular value $J_c(\alpha)$, the most probable value for D is $\mathrm{sgn}(J)$ while for smaller values of J, the most probable value is zero. This result suggests the following very simple learning rule:

$$D_i=0 \text{ if } |J_i|\leqslant J_c,$$
$$D_i=\mathrm{sgn}(J_i) \text{ if } |J_i|>J_c. \quad (18)$$

The crossing point J_c of the curves $D=0$ and $D=1$ lies close to $J=0.5$ for all values of α. I have carried out simulations for perceptrons of size $N=100$ using the simple learning rule Eq. (18) on the nearly optimal hypercube precursor and approximating J_c by 0.5 for all values of α. Results for the stability $\kappa(\alpha)$ of the generated diluted binary vectors are shown in Fig. 4 as a function of the pattern load α (squares). The solid curve displays the analytical results for the averaged MSDB of infinite size. All simulation data represent averages over at least 100 samples and the error bars are smaller than the diameter of the small circles in Fig. 4. The learned input patterns were drawn at random from a Gaussian distribution with zero mean and unit variance.

To improve on the simple learning rule Eq. (18), I performed enumerations on a subset of N_e components of the clipped vector **D** while keeping its remaining $(N-N_e)$ components fixed. To lower the numerical effort, partial enumeration was done using a branch and bound algorithm. Details on the bound conditions can be found in Appendix B. Theory predicts a high reliability of binary precursor components. Consequently, I accepted them always as ''correct'' predictions and excluded the corresponding components in the clipped vector from partial enumeration. The lower bound for the number of fixed components $N-N_e$ is hence given by the actual number N_b of binary precursor components. An estimate can be read of Table I which lists N_b/N for different α and the limit $N\to\infty$. For each particular enumeration scheme, I will keep the number N_e of enumerated components constant over the whole range of α values as long as I find enough precursor components J_i with $|J_i|<1$. Partial enumeration will be performed on all N_e vector components i where $|J_i|$ is closest to a particular value J^0. The number of non-binary precursor components decreases with decreasing α but nevertheless N_e shall be kept constant. Consequently, the values of precursor components $|J_i|$ which correspond to enumerated vector components D_i will vary

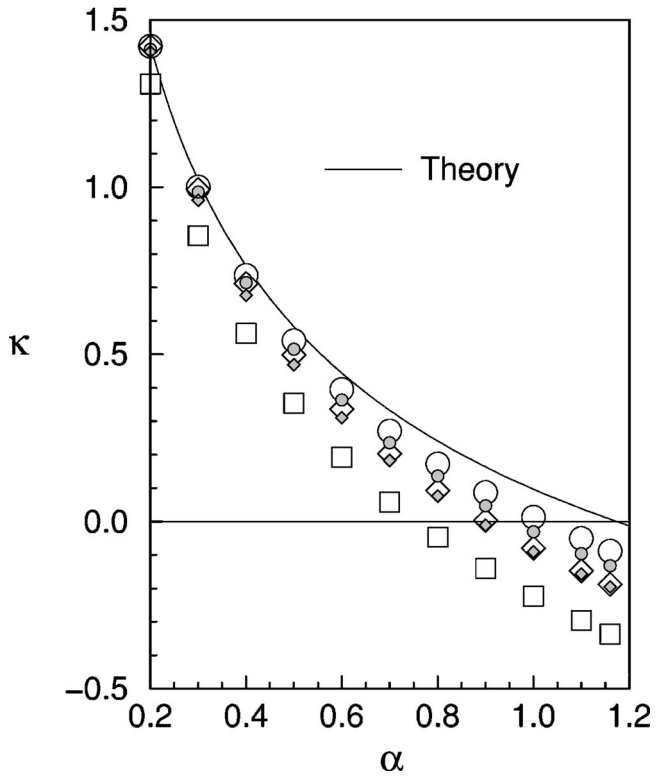

FIG. 4. Effect of different learning rules on the stability κ of diluted binary perceptrons of size $N=100$ for different pattern loads α. The stability κ of the clipped vector $\mathbf{D}^{\mathrm{clip}}$ is represented by squares. All other symbols refer to situations where $\mathbf{D}^{\mathrm{clip}}$ got improved by partial enumeration (a) $D_i=\{0,\mathrm{sgn}(J_i)\}$ for 16 components corresponding to $|J_i|\sim 0$ (large diamond) or to $|J_i|\sim 0.5$ (large circle), (b) $D_i=\{-1,0,1\}$ for 10 components corresponding to $|J_i|\sim 0$ (small diamond), or to $|J_i|\sim 0.5$ (small circle). The size of the respective enumeration space was almost equal.

for small α over a broader range of values than for larger α. This should be advantageous given the shape of $p(D|J)$.

Partial enumeration considers for each vacant vector component D_i all three possibilities $D_i=\{-1,0,1\}$. However, according to Fig. 3, two of these three possibilities are almost equally likely. The third possibility, an error in sign prediction, is expected to be rare.

The small symbols in Fig. 4 show the average stability of an improved variant of the clipped vector. The clipped vector is improved with respect to its stability by partial enumeration on a subset of ten components thereby considering the values $D_i=\{-1,0,1\}$. The subset of vector components D_i selected for enumeration did correspond either to precursor values $|J_i|\sim 0$ (small diamond) or to precursor values $|J_i|\sim 0.5$ (small circle). In a next step I neglected errors in sign prediction and considered only the two most likely values $\mathbf{D}_i=\{0,\mathrm{sgn}(J_i)\}$ while performing partial enumerations. Enumeration on the same subsets as above implies a considerable reduction of the size of the enumeration space since I am left with 2^{10} possibilities instead of 3^{10}. Naturally, the resulting stabilities would be lower and the reduction in stability gives an indication of the practical relevance of the influence of errors in sign prediction. Partial enumeration on the subset of 10 components corresponding to $|J_i|\sim 0$ (small diamonds) yields the following results: The average gain in stability to the clipped vector (squares) amounts for $\alpha=0.2$ to $\Delta\kappa(0.2)\sim 0.097$ while for $\alpha=1.16$ it takes on the value $\Delta\kappa(1.16)\sim 0.141$. The stability loss which would be induced by enumeration on the same subset of components with two values $D_i=\{0,\mathrm{sgn}(J_i)\}$ only would shift the small diamond down by $\Delta\kappa(0.2)\sim 0.006$ or by $\Delta\kappa(1.16)\sim 0.034$, respectively. I conclude that the influence of errors in sign prediction in the vicinity $|J_i|\sim 0$ is small in proportion to wrong decisions regarding the dilution of a vector component. However, it is not negligible. In contrast, for the subset of ten components that correspond to precursor components with $|J_i|\sim 0.5$ (small circle) errors in sign prediction have negligible influence on the resulting stability. The difference in stability $\Delta\kappa(\alpha)$ to the clipped vector ranges between $\Delta\kappa(0.2)\sim 0.103$ and $\Delta\kappa(1.16)\sim 0.204$. Neglecting errors in sign and enumerating the same subset of ten components would shift the small circle down by $\Delta\kappa(0.2)\sim 0.001$ or $\Delta\kappa(1.16)\sim 0.002$, respectively.

TABLE II. Simulation: Storage capacity α_c of different diluted binary perceptrons of size $N=100$. The clipped vector got improved by partial enumeration $D_i=\{0,\mathrm{sgn}(J_i)\}$ on a subset of vector components where $|J_i|\sim 0.5$.

Percentage of partial enumeration	Storage capacity α_c	Symbol	
0%	0.76	Square	(Fig. 4)
16%	1.02	Large circle	(Figs. 4,5)
30%	1.09	Triangle	(Fig. 5)

Alternatively, preserving the size of the enumeration space while taking into account only two values $\mathbf{D}_i=\{0,\mathrm{sgn}(J_i)\}$ allows to consider roughly 16 components instead of just 10 since $3^{10}\sim 2^{15.8}$. The result is indicated in Fig. 4 by large symbols (diamond and circle). The subset of 16 enumerated vector components D_i did correspond either to precursor values $|J_i|\sim 0$ (large diamond) or to precursor values $|J_i|\sim 0.5$ (large circle). Note, that for a vector size $N\geqslant 100$ and pattern loads $\alpha\geqslant 0.5$ I find two completely disjunct subsets of about 16 components with $|J_i|\sim 0.5$ or $|J_i|\sim 0$, respectively. By performing partial enumerations on the respective subsets of $\mathbf{D}$, I improve the clipped vector in completely different subspaces and can monitor the efficiency of partial enumeration on the magnitude of J_i.

The two types of diamonds and circles were obtained by the same numerical enumeration effort. The simulation clearly shows that partial enumeration is highly effective and superior to any other strategy in the vicinity of the crossing point $|J_c|\sim 0.5$ of the theoretical curves $p(0|J)$ and $p[\mathrm{sgn}(J)|J]$ while simultaneously considering only the two most likely values $D_i=\{0,\mathrm{sgn}(J_i)\}$.

The size of the enumerated subspace was chosen to keep the enumeration effort on a moderate level while it simultaneously allowed to obtain an informative picture on the efficiency of the different strategies. Table II shows for the most successful enumeration strategy the influence of the enumeration effort on the storage capacity of the obtained vectors. Partial enumeration $D_i=\{0,\mathrm{sgn}(J_i)\}$ was performed for 16 and 30% of the clipped vector on vector components being related to $|J_i|\sim 0.5$. This is visualized in Fig. 5.

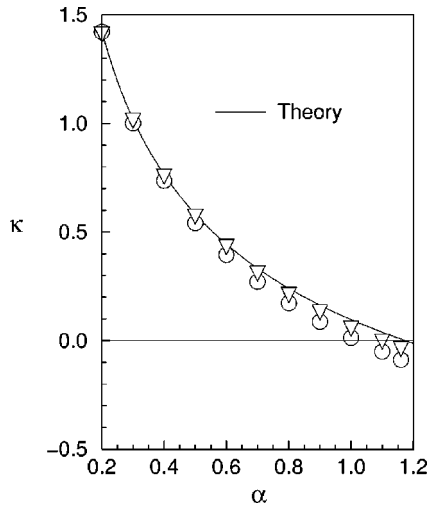

FIG. 5. Stability κ as a function of the pattern load α (perceptron size $N=100$). The clipped vector got improved by partial enumeration $D_i=\{0,\mathrm{sgn}(J_i)\}$ on a subset of vector components that are related to precursor values $|J_i|\sim 0.5$. Percentage of enumerated vector components: 16% (circle), 30% (triangle). The circles are identical to Fig. 4.

IV. SUMMARY

I showed analytically as well as by simulations that the coupling vector **D** of the MSDB is highly correlated to a continuous precursor **J** which is obtained by convex minimization with hypercube constraint. The predictive power of **J** can be appreciated from the conditional probability $p(D|J)$. It suggests a very simple learning rule for a diluted binary perceptron of high stability: All components of **J** that have magnitude greater than 0.5 are clipped whereas all weaker components of **J** get diluted. The nearly optimal hypercube precursor **J** has a considerable fraction of binary components which give a correct prediction for **D** with a probability of about 94%. Precursor components of magnitude $|J_i|<1$ give a highly reliable prediction of the sign whereas the decision to dilute the corresponding **D** component is less obvious. To test the efficiency of different learning rules with respect to the resulting stabilities I performed simulations for perceptrons of size $N=100$. The simple clipped vector

$$D_i=0 \quad \text{if } |J_i|\leqslant 0.5, \qquad (19)$$

$$D_i=\mathrm{sgn}(J_i) \quad \text{otherwise},$$

was found to be highly stable realizing on average a storage capacity $\alpha_c\sim 0.76$. Using a branch and bound algorithm, I improved the clipped vector by partial enumeration on its least reliable components while keeping its remaining components fixed. The best results were obtained by partial enumeration with $D_i=\{0,\mathrm{sgn}(J_i)\}$ on a subset of components corresponding to precursor components $|J_i|$ close to 0.5. Following this strategy and enumerating 30% of the vector results in a storage capacity of $\alpha_c\sim 1.09$. The obtained stability $\kappa(\alpha)$ gives a close approximation to its theoretical upper limit over the whole range of $\alpha\leqslant 1.17$.

ACKNOWLEDGMENTS

It is a pleasure to thank M. Bouten for drawing my attention to the problem, for stimulating discussions and for a critical reading of the manuscript. I thank B. Van Rompaey for valuable discussions on numerical aspects in the early stages of this work.

APPENDIX A: SADDLE POINT EQUATIONS

The central result of the replica calculation is Eq. (7). The distribution of the order parameters is regulated by the terms

$$G_1(\mathbf{q},\mathbf{Q},\mathbf{r})=\int \frac{d\vec{\lambda}\,d\vec{x}}{(2\pi)^n}\int \frac{d\vec{\Lambda}\,d\vec{X}}{(2\pi)^n} \times\exp\left(i\vec{x}\vec{\lambda}+i\vec{X}\vec{\Lambda}-\frac{1}{2}(\vec{x}\mathbf{q}\vec{x}+\vec{X}\mathbf{Q}\vec{X})-\vec{x}\mathbf{r}\vec{X}\right) \times\prod_{a=1}^{n} h(\lambda^a)\Theta(\Lambda^a-\kappa), \qquad (A1)$$

$$G_2(\hat{\mathbf{q}},\hat{\mathbf{Q}},\hat{\mathbf{r}})=\int\prod_{a=1}^{n} d\mu(J_1^a)\sum_{D_1^a}\exp\left(-\frac{1}{2}(\vec{J}_1\hat{\mathbf{q}}\vec{J}_1+\vec{D}_1\hat{\mathbf{Q}}\vec{D}_1) +\vec{J}_1\hat{\mathbf{r}}\vec{D}_1\right). \qquad (A2)$$

Vector notations run over the replica index, for example $\vec{\lambda}$ and $\vec{\Lambda}$ contain the replicas of the local field of an arbitrary pattern. I write the order parameters Eq. (6) as elements of the matrices $\mathbf{Q},\mathbf{q},\mathbf{r}$, their conjugate variables define the matrices $\hat{\mathbf{Q}},\hat{\mathbf{q}}$, and $\hat{\mathbf{r}}$

$$\hat{\mathbf{Q}}=i\begin{pmatrix}\hat{Q}_{11} & & -\hat{Q}_{ab}\\ & \ddots & \\ -\hat{Q}_{ab} & & \hat{Q}_{nn}\end{pmatrix},$$

$$\hat{\mathbf{q}}=i\begin{pmatrix}\hat{q}_{11} & & -\hat{q}_{ab}\\ & \ddots & \\ -\hat{q}_{ab} & & \hat{q}_{nn}\end{pmatrix}, \qquad (A3)$$

$$\hat{\mathbf{r}}=i(\hat{r}_{ab}).$$

Assuming replica symmetry, the order parameter density takes on the form

$$\exp(N[\cdots])=\exp(N[ns_{db}(\mathbf{Q},\hat{\mathbf{Q}})+ns_c(\mathbf{q},\hat{\mathbf{q}})+\mathcal{O}(n^2)]). \qquad (A4)$$

$s_{db}(\mathbf{Q},\hat{\mathbf{Q}})$ denotes the entropy of the diluted binary perceptron **D** [see Eq. (A5)] and $s_c(\mathbf{q},\hat{\mathbf{q}})$ is either the entropy (for maximum stability learning) or the free energy (for learning by minimization of a cost function) of the continuous precursor **J**. In the limit $n\to 0$, the leading order term in the exponent (A4) is independent of the correlation order parameters r_{ab} and $\hat{r}_{ab}$. Hence, $r_{ab},\hat{r}_{ab}$ can not be determined after taking the limit $n\to 0$. Rather, they are given by the saddle

point equations before the limit $n\to 0$ and follow from the general form of Eq. (7), Ref. [12]. In the replica symmetric ansatz, the saddle point equations for r and $\hat{r}$ reduce to Eqs. (A12), (A13). They depend on the correct saddle point values for the order parameters of the two individual perceptrons.

1. Saddle point equation for D

The order parameters Q_0, Q and $\hat{Q}_0, \hat{Q}$ must extremize the entropy s_{db}

$$s_{db}=\frac{Q_0\hat{Q}_0}{2}+\frac{Q\hat{Q}}{2}+\int_{-\infty}^{+\infty}Du\left\{\alpha\ln\left[H\left(\frac{\kappa-u\sqrt{Q}}{\sqrt{Q_0-Q}}\right)\right]+\ln\left[2\cosh(u\sqrt{\hat{Q}})\exp\left(-\frac{(\hat{Q}_0+\hat{Q})}{2}\right)+1\right]\right\}. \quad \text{(A5)}$$

The value for the maximal stability κ^{db} is determined simultaneously by the zero entropy condition $s_{db}=0$ [10,13].

2. Saddle point equation for J

The considered learning rules result in a unique solution, hence I focus on the limit $q\to q_0$. A hypercube restricted vector **J** that minimizes a cost function $E(\mathbf{J})=\Sigma_\nu V(\lambda^\nu)$ is described by a set of four saddle point equations [9]

$$y=1-2H\left(\frac{1}{s}\right), \quad \text{(A6)}$$

$$q_0=s^2y+2H\left(\frac{1}{s}\right)-\sqrt{\frac{2}{\pi}}se^{-1/2s^2}, \quad \text{(A7)}$$

$$y=-\frac{\alpha}{\sqrt{q_0}}\int_{-\infty}^{\infty}Du\ (\lambda_0-u\sqrt{q_0})\ u, \quad \text{(A8)}$$

$$y^2s^2=\alpha\int_{-\infty}^{\infty}Du\ (\lambda_0-u\sqrt{q_0})^2. \quad \text{(A9)}$$

The order parameters y and s are defined by

$$y=(\hat{q}_0+\hat{q})(q_0-q),\quad s=\frac{\sqrt{\hat{q}}}{\hat{q}_0+\hat{q}}. \quad \text{(A10)}$$

For $(q_0-q)\to 0$, $\hat{q}_0$ and $\hat{q}$ tend to infinity while y and s remain finite. Comparing Eqs. (A6) and (14) reveals that $1-y$ represents the fraction of binary precursor components. s regulates the shape of the distribution function $p(J)$. Equations (A8), (A9) depend on the choice of the potential $V(\lambda)$ via the function λ_0 defined as

$$\lambda_0=\mathrm{Arg}\min_{\lambda}\left[V(\lambda)+\frac{(\lambda-u\sqrt{q_0})^2}{2\mathrm{x}}\right]. \quad \text{(A11)}$$

The new variable $\mathrm{x}=\beta(q_0-q)$ is finite in the limit $q\to q_0$ which is driven by the inverse temperature $\beta\to\infty$.

For the vector that maximizes the stability κ_c under hypercube constraint, Eqs. (A6),(A7) remain valid while Eqs. (A8),(A9) must be modified: λ_0 is to be replaced by κ_c and the u integration is restricted to $\kappa_c-u\sqrt{q_0}\geqslant 0$.

For perceptrons on the hypersphere ($q_0=1$) the description is much simpler [2,17]. For learning by minimization of a cost function it suffices to know the value of saddle point variable x. For maximum stability learning κ_c is to be determined.

3. Coupling between J and D

In the limit $q\to q_0$, $\hat{r}$ becomes infinite while $\hat{r}_0=\hat{r}(q_0-q)$ remains finite. Using the saddle point values for the order parameters of the individual perceptrons, the order parameters $r,\hat{r}_0$ (or, respectively, $\gamma,\hat{\gamma}$) are determined by

$$r=(Q_0-Q)\frac{\hat{r}_0}{y}-2s\int Du\left[(\hat{\gamma}u+s^{-1})H\left(\frac{\hat{\gamma}u+s^{-1}}{\sqrt{1-\hat{\gamma}^2}}\right)-\frac{\sqrt{1-\hat{\gamma}^2}}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{(\hat{\gamma}u+s^{-1})^2}{1-\hat{\gamma}^2}\right)\right]\times\frac{2\sinh(u\sqrt{\hat{Q}})}{\exp((\hat{Q}_0+\hat{Q})/2)+2\cosh(u\sqrt{\hat{Q}})}, \quad \text{(A12)}$$

$$\hat{r}_0=\alpha\int\int D_\gamma(u,w)\frac{(\lambda_0-u\sqrt{q_0})}{\sqrt{2\pi(Q_0-Q)}}\times\exp\left(-\frac{1}{2}\frac{(\kappa^{db}-w\sqrt{Q})^2}{Q_0-Q}\right)H^{-1}\left(\frac{\kappa^{db}-w\sqrt{Q}}{\sqrt{Q_0-Q}}\right). \quad \text{(A13)}$$

$D_\gamma(u,w)$ denotes a two-dimensional Gaussian with zero mean and variance γ. Equation (A12) is valid for a hypercube constraint. For a spherical constraint ($q_0=1$), it is to be replaced by the simple identity $\hat{r}_0=r(Q_0-Q)^{-1}$. Equation (A13) is valid for any vector **J** that minimizes a cost function $E(\mathbf{J})=\Sigma_\nu V(\lambda^\nu)$ where λ_0 is defined by Eq. (A11). For a vector **J** that maximizes the stability κ_c, Eq. (A13) must be modified: λ_0 is to be replaced by κ_c and the u integration is restricted to $\kappa_c-u\sqrt{q_0}\geqslant 0$.

APPENDIX B: PARTIAL ENUMERATION WITH A BRANCH AND BOUND ALGORITHM

In this appendix I give some details on the enumeration algorithm. For a particular storage problem $\{\boldsymbol{\xi}^\nu\}$, $\nu=1,\ldots,\alpha N$, one has to determine the precursor **J**, the corresponding clipped vector $\mathbf{D}^{\mathrm{clip}}$ as well as the index field $i[k]$ which points to precursor components $J_i\neq\pm 1$ ranking them by their absolute distance to a value J^0

$$|J^0-|J_{i[k-1]}||\geqslant|J^0-|J_{i[k]}||. \quad \text{(B1)}$$

The positions of the N_b binary components of **J** are listed in arbitrary order by $i[1],\ldots,i[N_b]$. Partial enumeration starts on a diluted binary vector **D** where $N_f\geqslant N_b$ components

$$\mathbf{D}_{i[k]}=\mathbf{D}^{\mathrm{clip}}_{i[k]};\ k=1,\ldots,N_f \tag{B2}$$

are given by the clipped vector and kept fixed. For the remaining subset of N_e components all 3^{N_e} or 2^{N_e} combinatorial possibilities must be evaluated with respect to the minimal stability of the resulting **D** vector. The search space of 3^{N_e} or 2^{N_e} possibilities can be listed in a treelike structure. From a node of the order $m\in(0,N_e-1)$ separate three or two new branches. They copy $k=1,\ldots,(N_f+m)$ vector components $D_{i[k]}$ but differ with respect to the (N_f+m+1)-th component $D=\{-1,0,1\}$ or $D=\{0,\mathrm{sgn}(J_{i[N_f+m+1]})\}$, respectively.

The choice $D_{i[k]}=\mathrm{sgn}(\xi^{\nu}_{i[k]})$ for all free components $k=N_f+1,\ldots,N$ maximizes the local field

$$\Lambda^{\nu}(0)=\frac{1}{\sqrt{N}}\left(\sum_{k=1}^{N_f}D^{\mathrm{clip}}_{i[k]}\xi^{\nu}_{i[k]}+\sum_{k=N_f+1}^{N}|\xi^{\nu}_{i[k]}|\right) \tag{B3}$$

of a pattern $\boldsymbol{\xi}^{\nu}$. Any other choice results in a correction

$$\Lambda^{\nu}(m)=\Lambda^{\nu}(m-1)-\frac{2}{\sqrt{N}}|\xi^{\nu}_{i[k]}|\Theta(-\xi^{\nu}_{i[k]}D_{i[k]}),$$

$$k=N_f+m \tag{B4}$$

with $\Theta(0):=0.5$ and $m\geqslant 1$. The maximum value $\kappa=\max_{\mathbf{D}}[\min_{\nu}\mathbf{D}\boldsymbol{\xi}^{\nu}]/\sqrt{N}$ which is obtainable by the class of vectors **D** which coincide up to the mth node is hence bounded

$$\kappa\leqslant\min_{\nu}\Lambda^{\nu}(m) \tag{B5}$$

with a decreasing upper bound for increasing node number $m\leqslant N_e$. A branch can be cut off at node m when the value of the upper bound $\min_{\nu}\Lambda^{\nu}(m)$ drops below a reference value κ^{opt}. κ^{opt} gets initialized by the stability of the clipped vector D^{clip} and updated whenever a more stable **D** vector is found.

If partial enumeration is ignoring the possibility of an error in sign $D_{i[k]}=\{0,\mathrm{sgn}(J_{i[k]})\}$ for $k=N_f+1,\ldots,N$, the potential sign of $D_{i[k]}$ is known. This allows me to improve the upper bound Eqs. (B3),(B4). In order to maximize the local field $\Lambda^{\nu}(0)$ of a particular pattern $\boldsymbol{\xi}^{\nu}$, it is best to dilute $D_{i[k]}$ whenever $\mathrm{sgn}(J_{i[k]})\neq\mathrm{sgn}(\xi^{\nu}_{i[k]})$

$$\Lambda^{\nu}(0)=\frac{1}{\sqrt{N}}\min_{\nu}\left(\sum_{k=1}^{N_f}D^{\mathrm{clip}}_{i[k]}\boldsymbol{\xi}^{\nu}_{i[k]}+\sum_{k=N_f+1}^{N}|\boldsymbol{\xi}^{\nu}_{i[k]}|\Theta(\boldsymbol{\xi}^{\nu}_{i[k]}J_{i[k]})\right). \tag{B6}$$

Any other choice results in the correction

$$\Lambda^{\nu}(m)=\Lambda^{\nu}(m-1)-\frac{1}{\sqrt{N}}|\boldsymbol{\xi}^{\nu}_{i[k]}|(\Theta(\boldsymbol{\xi}^{\nu}_{i[k]}J_{i[k]})\delta_{0,D(i[k])}+\Theta(-\boldsymbol{\xi}^{\nu}_{i[k]}J_{i[k]})\delta_{\mathrm{sgn}\{J(i[k])\},D(i[k])}),$$

$$k=N_{\mathrm{f}}+m. \tag{B7}$$

The second sum in Eqs. (B3), (B6) is an overestimation giving an easy, robust but moderate bound condition. However, it is balanced by the first sum which is exact and known from the beginning since the major part of vector components is kept fixed $N_f>N_e$.

[1] K. Y. M. Wong and D. Sherrington, J. Phys. A **23**, 4659 (1990).
[2] M. Griniasti and H. Gutfreund, J. Phys. A **24**, 715 (1991).
[3] H. M. Koehler, J. Phys. A **23**, L1265 (1990); H. Horner, Z. Phys. B **86**, 291 (1992).
[4] C. J. P. Vicente, J. Carrabina, and E. Valderrama, Network **3**, 165 (1992).
[5] G. Milde and S. Kobe, J. Phys. A **30**, 2349 (1997).
[6] M. Schröder and R. Urbanczik, Phys. Rev. Lett. **80**, 4109 (1998); W. Nadler and W. Fink, *ibid.* **78**, 555 (1997).
[7] R. W. Penney and D. Sherrington, J. Phys. A **26**, 6173 (1993).
[8] L. Reimers, M. Bouten, and B. Van Rompaey, J. Phys. A **29**, 6247 (1996).
[9] M. Bouten, L. Reimers, and B. Van Rompaey, Phys. Rev. E **58**, 2378 (1998).
[10] J. Iwanski, J. Schietse, and M. Bouten, Phys. Rev. E **52**, 888 (1995).
[11] M. Bouten, A. Komoda, and R. Serneels, J. Phys. A **23**, 2605 (1990).
[12] K. Y. M. Wong, A. Rau, and D. Sherrington, Europhys. Lett. **19**, 559 (1992).
[13] W. Krauth and M. Mézard, J. Phys. (France) **50**, 3057 (1989).
[14] E. Gardner, J. Phys. A **21**, 257 (1988).
[15] M. Bouten, J. Schietse, and C. Van den Broeck, Phys. Rev. E **52**, 1958 (1995).
[16] T. L. H. Watkin, Europhys. Lett. **21**, 871 (1993).
[17] E. Gardner and B. Derrida, J. Phys. A **21**, 271 (1988).