

# Compression by Replication

Roberto C. Alamino, Juan P. Neirotti and David Saad

*Non-linearity and Complexity Research Group, Aston University, Birmingham B4 7ET, UK*

A recently introduced inference method based on system replication and an on-line message passing algorithm is employed to complete a previously suggested compression scheme based on a non-linear perceptron. The algorithm is shown to approach the information theoretical bounds for compression as the number of replicated systems increases, offering superior performance compared to basic message passing algorithms. In addition, the suggested method does not require fine-tuning of parameters or other complementing heuristic techniques, such as the introduction of inertia terms, to improve convergence rates to non-trivial results.

## I. INTRODUCTION

The successful application of techniques developed in statistical mechanics of disordered systems to a wide range of problems in information theory has benefitted both fields by exchanging methods and ideas, providing new insights and algorithmic tools [1]. The statistical mechanics methodology has complemented the mathematical rigor of traditional information theory techniques by providing exact analytical results for typical properties in the limit of very large systems - the thermodynamic limit. Conversely, the importance of obtaining *specific* microscopic states in practical information theory problems, in contrast to the usual goal of characterizing macroscopic states in statistical mechanics, has contributed to a renewed interest in inference methods to obtain ground state solutions of the corresponding Hamiltonians when the energy landscape is complex.

In problems related to communication systems, microscopic states are associated with specific transmitted messages and one is interested in recovering those messages rather than analyzing general macroscopic properties of an ensemble of them. The correct message refers to the ground state of the corresponding Hamiltonian with the level of noise in the channel being represented by the temperature used. Common examples are error correcting codes [2], where one wishes to recover the original message, after it has been encoded and corrupted upon transmission through a noisy channel.

The problem of finding a ground state, or equivalently the global minimum of a Hamiltonian, can only be solved analytically in very simple situations. For disordered systems, specially in the spin-glass phase, the energy landscape is so complex that the use of approximate computational techniques is unavoidable. The ruggedness of energy landscapes which characterize such systems poses a challenge for these techniques. For instance, gradient descent-based methods get trapped in local minima and more sophisticated Monte Carlo algorithms, such as Parallel Tempering [3, 4], are computationally slow.

An alternative family of algorithms which provide computationally efficient approximations to the exact but computationally hard full Bayesian inference is that of message passing (MP) algorithms, a.k.a. belief propagation [5]. These methods have been able to achieve

good performance in many complex problems and are considered a promising alternative to tackling inference problems in a range of fields such as information theory [1], hard combinatorial problems [6], statistical mechanics models and complex systems in general.

The information theoretical problem we address here is that of source coding or lossy compression. The problem is of great importance practically and is highly challenging theoretically; computationally efficient solutions for this problem have been sought after for over 50 years. Shannon was the first to study lossless and lossy [7, 8] compression and to establish theoretical bounds to the achievable performance under a given information loss. However, Shannon's results are not constructive, leaving open the challenge of finding a computationally feasible scheme that saturates the theoretical bounds.

The main difficulty in finding such schemes is the associated computational complexity. Some schemes can saturate the theoretical bounds, for instance by an exhaustive search, but are impractical due to the computational cost involved which scales exponentially with the size of the message. Other approaches provided good approximations [9–14] that still fall short of the theoretical limits for certain loss rates. The search for efficient schemes, those which are at least polynomial in message size, is what drives research in the field even today. Notable among these schemes are recent approximate Bayesian methods based on MP algorithms.

A radically different approach based on the non-linear perceptron has been introduced by Hosaka et al [15]. By using the replica method it has been shown that a non-linear perceptron can be used as part of a compression scheme, which can achieve close to optimal performance, both in terms of the theoretical compression-distortion limits [15] and the related error-exponents [16], depending on the parameters of the message generation and activation function of the perceptron. The analytical results are obtained for the typical case and were numerically verified only by exhaustive search methods, which are clearly exponentially slow as the size of the message is increased. In a follow-up work [17] an MP algorithm has been suggested for the compression of the messages showing good performance as long as some heuristic modifications were added; but performance was bounded due to inherent limitations of the inference method. While

this compression method is highly promising it still requires an efficient inference technique to bring performance closer to the theoretical limits.

The aim of the current paper is to do exactly that; namely, to adapt a recently introduced MP algorithm [18] to bring the performance of the non-linear perceptron compression scheme closer to the theoretical limits. This algorithm, named the replicated on-line message passing algorithm (rOnMP), is based on improved inference in hard computational problems by averaging over results obtained from different solutions of replicated variable systems. It has been previously applied to solve the binary Ising perceptron capacity problem as a benchmark case.

We will employ a similar replicated on-line message passing algorithm to obtain solutions as part of the compression method and show that it can achieve an increasingly better performance as the number of replica increases. The rOnMP algorithm exhibits several advantages over basic MP methods: (i) it does not require any heuristic additions to suppress convergence to non-trivial solutions due to symmetry properties of the problem; (ii) it does not depend on training parameters that may need fine-tuning; (iii) and most importantly, it provides increasingly improved performance as the number of replica increases.

The rest of this paper is organised as follows. In section II we give the theoretical formulation of the compression method using a non-linear perceptron. We then proceed to introduce in section III the usual MP equations corresponding to this method. The usual MP equations are online learning equations which use the whole dataset at once. In section IV we argue in favour of a modification of these equations which turns them into offline equations. These modifications improve the performance of the algorithm, but we still need another step to complete the proposed scheme. This next step is the replication of the algorithm and it is explained in section V. Once the rOnMP has been fully derived in its modified version for the compression problem, we analyse the issue of its computational complexity in section VI. The results of numerical simulations are presented on section VII and a summary with our conclusions and some final considerations is finally given on section VIII.

## II. COMPRESSION BY A NON-LINEAR PERCEPTRON

The compression problem consists in encoding an  $N$ -dimensional binary message  $\mathbf{y} = (y_1, \dots, y_N) \in \{\pm 1\}^N$  into a  $K$ -dimensional binary vector  $\mathbf{b} = (b_1, \dots, b_K) \in \{\pm 1\}^K$ , where  $K < N$ , such that the compressed message can be later recovered by a decompressing algorithm with zero or minimal loss. The compression rate  $R = K/N$  indicates the level of compression; it is desirable to minimize  $R$  while minimizing distortion losses. When zero information loss is possible in the recovered message we term

the problem lossless compression, while when allowing for some deterioration after recovery it is called lossy compression.

Given that we will usually work with finite compression rates during this work, we will refer to both  $N$  and  $K$  interchangeably as the *system size*. The thermodynamic limit will then be taken by sending both  $N$  and  $K$  to infinity while keeping  $R$  fixed to a finite value.

Shannon's source-code theorem [7] shows that lossless compression is possible when the rate  $R$  is less than the entropy per bit of the source  $\mathbf{y}$  in the thermodynamic limit. Higher compression rates can be achieved if one allows for information loss, with the precise meaning that a non-vanishing average error per bit will be expected in the retrieved message. The error per bit, also called the *distortion rate*, is measured by the average Hamming distance between the original message  $\mathbf{y}$  and its inferred version  $\hat{\mathbf{y}}$  as

$$D = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{\mu=1}^N \delta(y_\mu, \hat{y}_\mu), \quad (1)$$

where  $\delta$  represents the Kroenecker delta.

Perceptrons represent simple non-linear maps and have been extensively studied in statistical mechanics [19]. As such, they are promising candidates for compression schemes; one such specific scheme was proposed in [15]. The perceptron used corresponds to the mapping

$$y_\mu = \text{sgn}(\Delta - |\xi_\mu|), \quad (2)$$

with the so called *synaptic field* given by

$$\xi_\mu = \frac{1}{\sqrt{K}} \sum_{k=1}^K b_k s_{\mu k}, \quad (3)$$

where the *a priori* given vectors  $\{\mathbf{s}_\mu\}$  are fixed at each instance of the problem. The non-linear activation function used is visualized in Fig. 1, where it can be seen that the constant threshold  $\Delta$  defines the width of the square bump.

In the suggested compression scheme, a fixed set of  $N$  randomly sampled  $K$ -dimensional vectors  $\mathbf{s}_\mu \in \mathbb{R}^K$  is given, playing the role of fixed parameters that characterize the compression scheme and facilitate decompression. The dataset  $\mathcal{D}$  composed of pairs  $(y_\mu, \mathbf{s}_\mu)$  is used to estimate the parameters of the perceptron, which represent the compressed codeword corresponding to the original the data vector  $\mathbf{y}$ . These parameters are encoded by the vector  $\mathbf{b}$ , known in the literature as the *synaptic vector*. Decompression consists in presenting the input vectors  $\mathbf{s}_\mu$  to the perceptron to obtain the decompressed message  $\hat{\mathbf{y}}$  using Eq. (2).

Typical case analysis of the achievable compression rate for a given distortion (error rate)  $D$  was carried out using the replica method, with replica symmetry being sufficient in this case [15]. The data bits  $y_\mu \in \{\pm 1\}$  of the original message were randomly sampled from a biased distribution  $p = \mathcal{P}(y_\mu = 1) = 1 - \mathcal{P}(y_\mu = -1)$ ; the

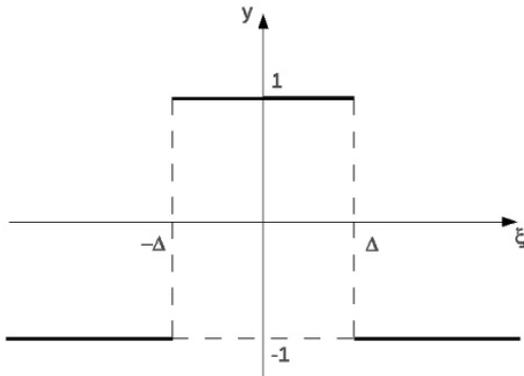


FIG. 1. Activation function for the non-linear perceptron used in the present compression scheme.

parameter vectors  $s_{\mu k}$  were sampled from a Gaussian distribution of zero mean and unit variance. Theoretically, the compression scheme saturates Shannon's limit, with a compression rate  $R(D)$

$$R = H_2(p) - H_2(D) \quad (4)$$

$$\text{and} \quad \text{erf}\left(\frac{\Delta}{\sqrt{2}}\right) = \frac{p - D}{1 - 2D}, \quad (5)$$

where  $H_2(x) = -x \log_2 x - (1-x) \log_2 (1-x)$ .

Equation (4) represents the optimal compression rates for a given fixed  $D$  for any message bias, while Eq. (5) gives the optimal value of  $\Delta$  in terms of the bias and either  $D$  or  $R$ . We will use these results to determine  $\Delta$  in our scheme and compare the performance of our algorithm with optimal compression.

### III. MESSAGE PASSING

In [15], the validity of the replica solution for the suggested compression scheme was tested using an exhaustive search, which is very slow and practically infeasible as it scales exponentially with the size of the system. A MP version of the proposed algorithm to implement the compression scheme was then suggested by two of the same authors of that study in [17]. Our objective in this paper is to suggest an alternative efficient compression algorithm, where 'efficient' is applied with that meaning that the algorithm's complexity should scale polynomially with the system size.

The algorithm we propose here is based on a modified version introduced in [18] of the usual MP algorithm. The introduced modifications had the objective of circumventing one particularly serious recurring problem in optimization tasks for perceptron learning in general - the complexity of the energy landscape.

Two key modifications of usual MP algorithms were fundamental to improve the obtained solutions:

1. **Making MP an online algorithm.** The usual MP algorithm, which aims to provide a good approximation to the maximum a posteriori solution for all available data simultaneously, are offline equations. We modified them and created an online algorithm where data points (patterns and corresponding classifications) are introduced one at a time. This allows one to explore a new degree of freedom which does not exist in the off-line version, namely the order of data presentation.
2. **Replicating the algorithm.** We introduced a series of real replicated systems exposed to the same set of data/constraints, but setting a different path through example space for each of them. Final estimates are obtained by averaging over the inferred solutions calculated by each one of the replicated systems.

The method proved to be extremely good in tackling the binary Ising perceptron capacity problem, which is computationally hard in both worst and typical cases. One of the strengths of this method is in its generality; in principle, it can be easily applied to any non-pathological densely connected inference problem with minimal modifications. The basis of the method, which we called rOnMP (Replicated Online MP), is of course the usual MP equations. Because we are dealing with a densely connected system, we however need to derive an approximation for these equations. This is done by means of an expansion which is valid in the limit of large systems. The details of this approximation were given in [18] and we therefore provide below only a brief description of its derivation.

The ordinary MP equations are given as pairs of coupled equations for each cavity magnetization  $m_\mu$ . These equations are

$$\hat{m}_{\mu k}^t = \frac{\sum_{b_k} b_k \mathcal{P}^{t+1}(y_\mu | b_k, \{y_{\nu \neq \mu}\})}{\sum_{b_k} \mathcal{P}^{t+1}(y_\mu | b_k, \{y_{\nu \neq \mu}\})}, \quad (6)$$

$$m_{\mu k}^t = \tanh \left[ \sum_{\nu \neq \mu} \text{atanh } \hat{m}_{\nu k}^t \right] \approx \tanh \left( \sum_{\nu \neq \mu} \hat{m}_{\nu k}^t \right). \quad (7)$$

The temporal index  $t$  in the variables indicates the update order as the equations should be solved by iteration until they converge. The variables  $\hat{m}$  are auxiliary variables used to calculate the actual cavity magnetizations  $m$  and are sometimes called *conjugate magnetizations*.

The method proposed in [18] can then be easily adapted to the present compression scheme. Using the non-linear activation function for the current perceptron, one can calculate analytically both the numerator and denominator of the equation for the conjugate magnetizations as a power series in  $K$ . This is accomplished by uncoupling the synaptic vectors by means of Hubbard-Stratonovich fields, which can then be exactly integrated at the end. By expanding the solution to leading order

in  $K$  we finally obtain

$$\hat{m}_{\mu k} = \frac{2s_{\mu k}y_{\mu}}{\sqrt{K}} \frac{\mathcal{N}(\Delta| -u_{\mu k}, \sigma_{\mu k}^2) - \mathcal{N}(\Delta| u_{\mu k}, \sigma_{\mu k}^2)}{1 + y_{\mu} \left[ \operatorname{erf}\left(\frac{\Delta + u_{\mu k}}{\sqrt{2\sigma_{\mu k}^2}}\right) + \operatorname{erf}\left(\frac{\Delta - u_{\mu k}}{\sqrt{2\sigma_{\mu k}^2}}\right) - 1 \right]}, \quad (8)$$

$$\text{where } \mathcal{N}(x|u, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-u)^2}{2\sigma^2}\right], \quad (9)$$

$$\sigma_{\mu k}^2 = \frac{1}{K} \sum_{l \neq k} (1 - m_{\mu l}^2) s_{\mu l}^2, \quad (10)$$

$$u_{\mu k} = \frac{1}{\sqrt{K}} \sum_{l \neq k} m_{\mu l} s_{\mu l}, \quad (11)$$

and  $\operatorname{erf}(x)$  is the error function given by

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x d\tau e^{-\tau^2}. \quad (12)$$

The new pair of equations to be iterated until convergence is now (7) and (8). By dropping the index  $t$  to indicate fixed point values once convergence is attained, we can write for the inferred value of the perceptron synaptic vector  $b_k$  the equation

$$b_k = \operatorname{sgn} m_k, \quad m_k = \tanh\left(\sum_{\mu} \hat{m}_{\mu k}\right). \quad (13)$$

#### IV. OFFLINE $\times$ ONLINE

The equations derived in the previous section are the offline version of the MP algorithm, which means that the dataset used during the learning phase is presented in full as a batch to the algorithm. The algorithm then has the freedom to use the whole dataset at once to extract the necessary information for the inference task. In this case, it is obvious that the order in which each data point is presented is irrelevant as the algorithm has the freedom to go back and forth within the dataset as needed to extract the maximum possible amount of information.

Equations (6), (7) and (13) are fairly general offline equations for binary systems and the (non-replicated) MP approach of [17] is, in practice, equivalent to them. This set of equations, however, have an inconvenient symmetry which gives rise to an ambiguity in deciding on the sign of the inferred variables. The reason is that, due to this symmetry, the equations always give the result  $m_k = 0$ . In the same way as we did above, their approach also considers the first significant term in  $\hat{m}_{\mu k}$ . To solve the symmetry problem, a heuristic inertia term was introduced [20] which depends on a parameter that has to be adjusted by trial and error.

Given the success we obtained by turning the offline MP algorithm into its online version in the perceptron capacity problem, we were led to introduce the same modification here. The online version of the MP equations

is obtained via an additional expansion for large system sizes, this time using the second of equations (13).

Because in this equation each term of the summation inside the hyperbolic tangent is of order  $1/\sqrt{K}$ , we can single out one of these terms and expand the tanh around the remaining terms for large  $K$ . By singling out the  $\nu$ -th term, the general formula becomes

$$m_k = \sum_{n=0}^{\infty} \frac{\hat{m}_{\nu k}^n}{n!} F_n(m_{\nu k}), \quad (14)$$

where

$$m_{\nu k} = \tanh\left(\sum_{\mu \neq \nu} \hat{m}_{\mu k}\right), \quad (15)$$

and

$$F_n(m_{\nu k}) = \frac{d}{dx} \tanh x \Big|_{x=\sum_{\mu} \hat{m}_{\mu k}}. \quad (16)$$

The fact that each  $F_n$  depends only on  $m_{\nu k}$  is a consequence of the derivative of the hyperbolic tangent being a function of the hyperbolic tangent itself.

We now re-interpret the term which was singled out as a new example, introduced after the previous patterns have already been learned. Alternatively, one may interpret the index  $\nu$  in the MP equations as a time step  $t$  and substitute  $m_{\nu k}^t$  by  $m_k(t-1)$  and  $m_k^t$  by  $m_k(t)$ . We run an extensive series of tests that led to the conclusion that by doing this expansion up to the third term in equation (14) one can avoid the problematic symmetry effects of the off-line MP and also provide extremely good compression performance. Explicitly, the expansion gives the following update rule

$$m_k(t) = m_k(t-1) + [1 - m_k^2(t-1)] \hat{m}_{tk} - m_k(t-1) [1 - m_k^2(t-1)] \hat{m}_{tk}^2. \quad (17)$$

At first sight, it would seem that the second order term would contribute to further stabilization of the zero solution as it works as a shrinking factor multiplying the previous value of  $m_k$ . However, due to the fact that the first order term makes the solution identically zero, this term actually works as a perturbation away from zero and, although small, it breaks the symmetry enough to allow the algorithm to pick a sign for the magnetization.

The modification from an offline algorithm to an online one is a very important one as it opens up the possibility of using a degree of freedom which was not available before - the order of data presentation. As we have seen in the beginning of this section, an offline algorithm like MP has access to the whole dataset and can use it in any order and how many times it is necessary to extract information. Online algorithms, however, can have access only to the information available *before* some point in time. Inference is then updated for each time step using new data in the order it arrives.

Although online algorithms seem restrictive in comparison to offline ones, they have some advantages in changing environments. The characteristic that is useful for us, however, is the element of disorder introduced by the possible random orders of the data points. Randomness can help the system to avoid getting trapped in dynamical local minima, something that is a fundamental limitation of the offline MP.

## V. REPLICATION

The final and most important step of the rOnMP algorithm is the system replication. Replication, in the sense of this algorithm, means to introduce real replicas of the same system which carry on parallel inference tasks and interact at very specific points.

This replication is done by generating randomly  $n$  different paths in the example space. Each path is composed by the same example pairs  $(y_\mu, \mathbf{s}_\mu)$ , but in a different order. The idea, as we have already discussed briefly, is to use the new degree of freedom encoded by the order of example presentation to facilitate a better search in solution space and avoid being trapped in suboptimal minima. For  $N$  examples, there are  $N!$  possible orders of presentation, but we will choose only a number  $n$  of these sequences, with  $n$  being of polynomial order in  $N$ . In previous applications [18] we observed that this is enough to considerably improve the performance of the non-replicated algorithm.

Once  $n$  different paths through the example space have been generated, parallelization takes place. For each example path, a different replicon of the system is created. Each replicon works as an independent system performing the online MP learning after each corresponding example is presented to it and inferring a new  $\mathbf{b}^a$ , where  $a$  is a replica index ranging from 1 to  $n$ , and the components of each vector are given by equation (13) applied to each replicon.

After each one of these learning steps, an averaged inferred vector is calculated by taking a weighted average of all replica as

$$\bar{b}_k = \text{sgn} \left( \sum_{a=1}^n w^a b_k^a \right), \quad (18)$$

and this is used as the initial point for the next learning step for each replicon.

The crucial point in the rOnMP algorithm is clearly how to decide on the weights for the averaging. Although white averages, with all weights equal to 1, are usually faster to calculate, they exhibit very poor performance in the present compression case. One can alternatively adopt a procedure based on a Boltzmann weight

$$w^a \propto e^{-\beta E(\mathbf{b}^a)}, \quad (19)$$

with the energies  $E(\mathbf{b}^a)$  being a measure of performance, which here we define as the number of misclassified examples. The parameter  $\beta$  works as an inverse temperature

and we recover the white average for  $\beta = 0$  (infinite temperature).

For the compressor, we observed that a much better performance is attained when the average is highly biased, which is equivalent to choosing a very low temperature to select lower energy states. We adopted then the rather simplified criteria of choosing the best replicon as the inferred vector for the next learning step, which amounts to choosing the zero temperature weights.

## VI. COMPUTATIONAL COMPLEXITY

There is of course a trade off between performance and computational complexity which cannot be avoided in the present situations. The complexity of the energy landscape for the present problem suggests that exact algorithms are invariably computationally hard. The more sophistication we add to the search algorithm then, the more we expect the computational complexity to increase. However, given the difficulty of the task, as long as the complexity of the resulting algorithm remains polynomial in the system size with a small exponent, we can accept this trade off.

As we are assuming that the system size  $K$  scales with the size of the dataset  $N$ , the naive MP algorithm, summarized in equations (6) and (7), requires the calculation of  $2K^2$  terms as each equation depends on two indices. In order to calculate the hatted variables, equation (8) requires two loops, each one of order  $K$ . The magnetizations have also one internal loop of order  $K$ , making the total number of operations scale with  $K^3$ . The final piece of inference does not increase the complexity as it requires only  $K^2$  operations.

Although replication increases the computational complexity of the original MP equations, because we work with order  $N$  replicas, the replicated algorithm scales with  $K^4$ , which is still polynomial in the system size with an exponent which has increased only one order in  $K$ . The replica averaging operation does not change this result as it scales only with  $K$ .

Therefore, the computational complexity of the rOnMP algorithm remains polynomial in the number of examples  $N$  as we aimed from the beginning.

It is not difficult to see that the complexity that should have appeared from the improvement in the performance of the algorithm might be hiding in the procedure for deciding on the order of example presentation. The most efficient way of choosing this order is indeed a difficult problem and needs to be considered with much more detail. During our tests, we have used a random order for the examples. Although this is far from optimal, even this very naive approach resulted in a considerable improvement of the algorithm's performance. This gives grounds for us to suggest that by modifying this procedure, one can achieve even better results.

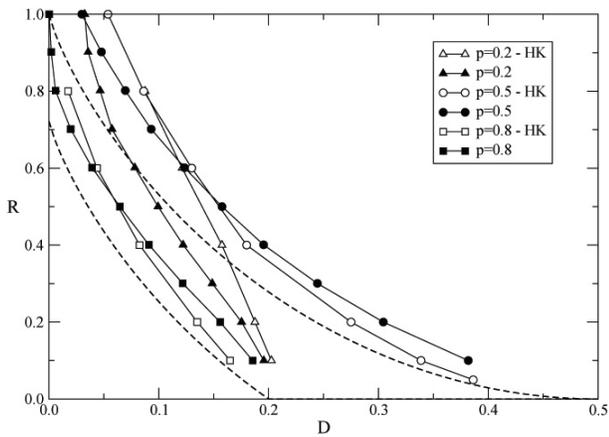


FIG. 2. Performance of the rOnMP-based compression scheme for different levels of bias. Dashed lines in this and the following figures represent the theoretical bounds; the top line for bias  $p = 0.5$  and the lower one for  $p = 0.2, 0.8$ . Solid lines with full symbols are averages over 100 instances using  $n = 10^4$  replica while those with hollow symbols represent results presented in [15] for comparison.

## VII. RESULTS

We tested the performance of the rOnMP algorithm against results published in the literature for different bias values of the pattern components. Trials with different weighting options for the averaging of the replicas indicate that, contrary to results with the binary Ising perceptron, there is a considerable difference between results obtained using white and weighted averages. For the suggested compression scheme, choosing the best performing replicon at each step turns out to be much more efficient than any other choice; it has therefore been used in all the experiments reported below.

Figure 2 shows the performance of the compression scheme in terms of the average bit error or distortion  $D$  versus the compression rate  $R$  for three different bias values  $p = 0.2, 0.5, 0.8$  and system size  $K = 101$ . The dashed curves show the theoretical bounds, while the solid lines with full symbols show averages over 100 different sets of Gaussian distributed randomly generated patterns with zero mean and unit variance. Each experiment was run with a total of  $n = 10000$  replica.

The performance shown in Fig. 2 is better than the results presented in [17] (hollow symbols) for high  $R$  values and deteriorates in the lower  $R$  regime for  $p = 0.5, 0.8$ . For  $p = 0.2$  our results are better for all  $R$  values. As with previous uses of the rOnMP algorithm, performance improves as the number of replica increases. Such an improvement is exemplified in Fig. 3, where we present results for  $n = 10, 100, 1000, 10000$  replica for the case of  $K = 101$  and  $p = 0.5$ .

This shows that the main limit for further improvement is computing time. Another notable feature of 2 is that our results for  $p = 0.2$  and  $p = 0.8$  are much closer

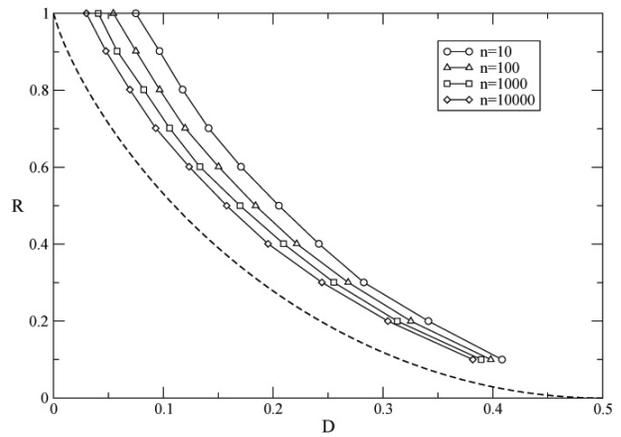


FIG. 3. Performance dependence on the number of replica  $n$  for bias  $p = 0.5$  and  $K = 101$ .

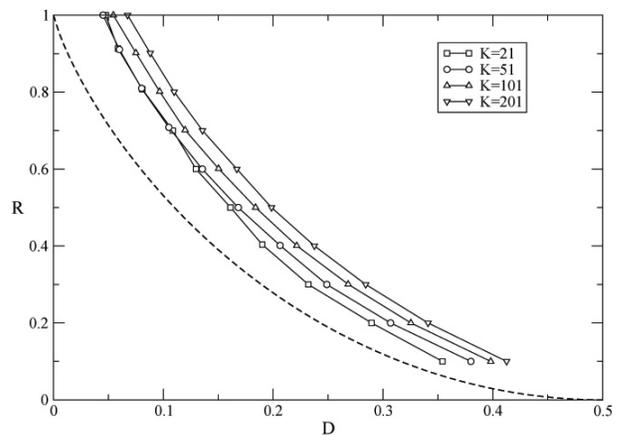


FIG. 4. Performance dependence on the system size  $K$  for  $p = 0.5$  and  $n = 100$ .

to each other than in previous works. Further experiments indicate that the smaller the size of the system, the closer the curves become using the same algorithm.

Finally, Fig. 4 shows how results change with increasing system size. The graph shows results obtained for  $K = 21, 51, 101, 201$  with a bias  $p = 0.5$  and  $n = 100$  replica. We can see an effect common to most systems with a complex energy landscape. The larger the system, the larger the number of local minima with a higher probability for the algorithm to get trapped; the number of replica needed to attain the same performance increases.

## VIII. CONCLUSIONS

We applied the recently introduced replicated online message passing algorithm (rOnMP) [18] to the promising compression method based on a non-linear perceptron suggested in [15].

The rOnMP algorithm is based on insights from statistical physics and uses a parallel replication of the approx-

imate Bayesian inference procedure known as message passing (MP) to explore the complex energy landscape that characterizes the parameter estimation problem of the non-linear perceptrons. In addition, the algorithm explores a new way of effecting message passing by changing the usual offline MP equations to an online version and by using an expansion for a large system size  $K$ .

We showed that our algorithm offers superior performance with respect to conventional MP methods with several additional advantages: (i) The performance of the algorithm is only limited by the available running time as our tests indicate that the larger the number  $n$  of replica, the closer to the theoretical performance limits the algorithm gets. (ii) The particular compression scheme we employ suffers from inherent symmetries which prevent the algorithm from converging to the correct value of the binary variables; this usually requires the introduction of

a heuristic inertia term [20] in the MP equations. This term is characterized by a constant that has to be fine-tuned. On the other hand, our online replicated version of MP does not require any adjustable parameters.

We believe that there are still room for further improvement of the results presented. The natural step is to judiciously choose the path in the example space in a way that maximizes the extraction of information from the set of examples. However, given the complexity of the solution space this requires new tools and approaches that are currently investigated.

## ACKNOWLEDGEMENTS

Support by the Leverhulme trust (F/00 250/M) is acknowledged. We also thank the suggestions of the anonymous referees.

- 
- [1] H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing* (OUP, Oxford, UK, 2001).
  - [2] Y. Kabashima and D. Saad, *EPL* **45** (1999).
  - [3] R. H. Swendsen and J.-S. Wang, *Phys. Rev. Lett.* **57**, 2607 (1986).
  - [4] E. Marinari and G. Parisi, *EPL* **19**, 451 (1992).
  - [5] J. Pearl, *Probabilistic Reasoning in Intelligent Systems* (Morgan Kaufmann, San Francisco, USA, 1988).
  - [6] M. Mézard and A. Montanari, *Information, Physics, and computation* (Oxford University Press, Oxford, UK, 2009).
  - [7] C. E. Shannon, *Bell Syst. Tech. J.* **27**, 379 (1948).
  - [8] C. E. Shannon, *Bell Syst. Tech. J.* **27**, 623 (1948).
  - [9] T. M. Cover and J. A. Thomas, *Elements of Information Theory* (Wiley-Interscience, 2006).
  - [10] D. MacKay, *Information Theory, Inference and Learning Algorithms* (Cambridge University Press, Cambridge, UK, 2004).
  - [11] G. Caire, S. S. Shamai, and S. Verdú, in *ITW'2003, IEEE Information theory workshop* (Paris, 2003).
  - [12] S. Ciliberti, M. Mézard, and R. Zecchina, *Phys. Rev. Lett.* **95**, 038701 (2005).
  - [13] A. Braunstein, F. Kayhan, and R. Zecchina, in *International Symposium on Information Theory* (IEEE Press, NJ, USA, 2009) pp. 1978–1982.
  - [14] M. Wainwright, E. Maneva, and E. Martinian, *Information Theory, IEEE Transactions on* **56**, 1351 (2010).
  - [15] T. Hosaka, Y. Kabashima, and H. Nishimori, *Phys. Rev. E* **66**, 066126 (2002).
  - [16] T. Hosaka and Y. Kabashima, *Jour. Phys. Society of Japan* **74**, 488 (2005).
  - [17] T. Hosaka and Y. Kabashima, *Physica A* **365**, 113 (2006).
  - [18] R. C. Alamino, J. P. Neirotti, and D. Saad, *Phys. Rev. E* **88**, 013313 (2013).
  - [19] A. Engel and C. van den Broeck, *Statistical Mechanics of Learning* (Cambridge University Press, 2001).
  - [20] T. Murayama, *Phys. Rev. E* **69**, 035105 (2004).