6 TAP For Parity Check Error Correcting Codes

David Saad, Yoshiyuki Kabashima and Renato Vicente

We employ the methods presented in the previous chapter for decoding corrupted codewords, encoded using sparse parity check error correcting codes. We show the similarity between the equations derived from the TAP approach and those obtained from belief propagation, and examine their performance as practical decoding methods.

1 Introduction

In the previous chapter we presented our general derivation of the *Thouless-Anderson-Palmer* (TAP) [29] approach and concentrated on its application to cases of extensively connected systems, showing the consistency between our approach and existing results obtained via the conventional derivation of the TAP equations.

In the current chapter we concentrate on the actual application of the method for decoding corrupted codewords, encoded using sparse parity-check error-correcting codes. The main motivation for developing the new formulation is the inapplicability of conventional TAP approaches to intensively connected systems and as decoding methods in this context.

The decoding method obtained from the new TAP formalism in the case of parity-check error-correcting codes is identical to that obtained from Belief Propagation (BP), an iterative method for approximating the marginal posterior probability of the dynamical variables given the quenched variables (data). The origins of BP are within the field of graphical models [22] and it has been successfully used for decoding in a broad range of error correcting codes [5]. Belief propagation is based on propagating probabilities through a tree structure, and thus provides an exact estimate to the marginal probability distributions when no recurrent connections are present.

This chapter is organized as follows: In the next two sections, we introduce the general background to the problem of error-correction and present the sparse parity-check codes examined here. We then derive, in section 4, the iterative decoding equations via the methods of BP. In section 5 we employ our formulation to the TAP approach to re-derive the same iterative equations, and point to the link between the two approaches. The decoding performance of the method is then demonstrated in several cases (section 6) and compared to analytical results obtained from the replica method. We end the chapter by discussing the advantages and limitations of the method and its relations to other existing techniques.

2 Noisy information transmission

Error-correcting codes are of significant practical importance as they provide mechanisms for retrieving the original message after possible corruption due to noise during transmission. They are being used extensively in most means of information transmission from satellite communication to the storage of information on hardware devices. The coding efficiency, measured in the percentage of informative transmitted bits, defines the information redundancy used to compensate for the corruption during transmission. Rigorous bounds [25] have been derived for the maximal channel capacity for which codes, capable of achieving arbitrarily small error probability, can be found.

One family of codes, introduced originally by Gallager [6; 7], and abandoned in favor of other methods due to the limited computing facilities of the time, has recently been re-introduced by MacKay and Neal [15; 16], showing excellent performance with respect to most existing codes. In fact, some of the irregular constructions presented recently have superior performance [17; 23; 12; 13], comparable to those of the turbo codes [3], and nearly saturate Shannon's bound for infinite message size. Gallager-type methods are generally based on the introduction of random sparse matrices for encoding the message as well for decoding the corrupted codeword. Various decoding methods have been successfully employed; here we mainly focus on the leading technique of belief propagation [22; 5] and its similarity to the method derived from our formulation of the TAP approach [9; 30].

In a general scenario, depicted in Fig.6.1, the sender encodes an N dimensional Boolean (or binary) message vector $\boldsymbol{\xi}$ to an $M(> N)$ dimensional codeword $\boldsymbol{t}$, which is then being transmitted through a noisy communication channel. Different channel types and noise models may be used [4], the most common ones being the Gaussian and Binary Symmetric Channel (BSC). In the former, the message bits are transmitted as real values and are being corrupted by white Gaussian noise; in the latter, the transmitted bits are flipped with some probability p characterizing the channel noise. Although both BP and our TAP decoding can be applied to both channel and noise models we will focus here on the BSC as the treatment is simpler and more transparent.

In the BSC, noise corruption during transmission can be modeled by the noise vector $\boldsymbol{\zeta}$ such that the received corrupted codeword takes the form $\boldsymbol{r} = \boldsymbol{t} + \boldsymbol{\zeta}$ in modulo 2. The received message is then decoded by the receiver for retrieving the original message $\boldsymbol{\xi}$. As we already mentioned, the error-correcting ability comes at the expense of information [25]; in the case of BSC, for unbiased messages, error-free communication is theoretically possible if the code rate $R = N/M$ satisfies the condition

$$R \leq 1 + p \log_2 p + (1-p) \log_2 (1-p) \ . \tag{1}$$

In the more general case of biased messages $(P(\xi_i = 1) = f_s \ , \ \forall i)$ and allowing a decoding bit error probability p_b, the maximal code rate R_c, for a given flip rate p,

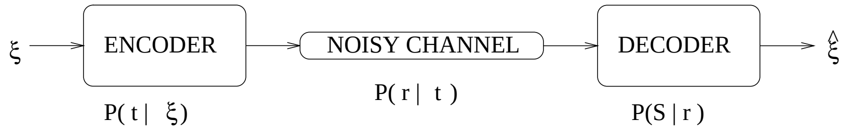

Figure 6.1
The encoding, the corruption by noise in the channel and the decoding can be though as a probabilistic network. The aim is to obtain a good estimative $\widehat{\boldsymbol{\xi}}$ for the message sent.

which equals the channel capacity, is given explicitly [4] by

$$R_c = H_2(f_s)(1-H_2(p))/(1-H_2(p_b)) \ , \qquad (2)$$

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

3 Sparse parity-check codes

The origin of sparse parity-check error-correcting codes can be traced back to the papers of Gallager [6; 7] where his original algorithms have been presented.

Gallager's code

Gallager's original code is defined by a binary matrix $\boldsymbol{A} = [\boldsymbol{C_s} \mid \boldsymbol{C_n}]$ constructed by concatenating two very sparse matrices known by both sender and receiver, with C_n (of dimensionality $(M-N)\times(M-N)$) being invertible and the matrix C_s of dimensionality $(M-N)\times N$.

Encoding is carried out by mapping the original message $\boldsymbol{\xi} \in \{0,1\}^N$ onto a binary vector $\boldsymbol{t} \in \{0,1\}^M$ $(M > N)$ defined by $\boldsymbol{t} = \boldsymbol{G^T}\boldsymbol{\xi}$ (mod 2), where all operations are performed in the field $\{0,1\}$ as indicated by the (mod 2) operation. The (dense) generator matrix used in the encoding process is $\boldsymbol{G} = [\boldsymbol{I} \mid \boldsymbol{C_n^{-1}} C_s]$ (mod 2), where $\boldsymbol{I}$ is the $N \times N$ identity matrix; this implies that $\boldsymbol{AG^T}$ (mod 2) $= 0$ and that the message $\boldsymbol{\xi}$ is set as the first N bits of $\boldsymbol{t}$. In a *regular* Gallager code the number of ones in each row of $\boldsymbol{A}$ is chosen to be exactly K and the number of elements per column is $C = (1-R)K$, where the code rate is $R = N/M$. These elements can be chosen either systematically or randomly. In irregular constructions the number of unit elements per row and connections per column may vary.

In a BSC, the encoded vector $\boldsymbol{t}$ is then corrupted by noise represented by the vector $\boldsymbol{\zeta} \in \{0,1\}^M$ with components independently drawn from the probability distribution

$$P(\zeta_i) = (1-p)\,\delta(\zeta_i) + p\,\delta(\zeta_i - 1) \ , \ \forall i \ .$$

The received vector takes the form $\boldsymbol{r} = \boldsymbol{G^T}\boldsymbol{\xi} + \boldsymbol{\zeta}$ (mod 2).

Decoding is carried out by multiplying the received message by the matrix $\boldsymbol{A}$, producing the *syndrome* vector $\boldsymbol{z} = \boldsymbol{Ar} = \boldsymbol{A\zeta}$ (mod 2) from which an estimate $\widehat{\boldsymbol{\zeta}}$ for the noise can be produced. An estimate for the original message is then obtained as the first N bits of $\boldsymbol{r}+\widehat{\boldsymbol{\zeta}}$ (mod 2). The Bayes optimal estimator (also known as *marginal posterior maximiser*, MPM) for the noise is defined as $\widehat{\zeta}_j = \mathrm{argmax}_{\zeta_j} P(\zeta_j \mid \boldsymbol{z})$. The performance of this estimator can be measured by the probability of bit error

$$p_b = 1 - 1/M \sum_{j=1}^{M} \delta[\widehat{\zeta}_j ; \zeta_j] \ ,$$

where $\delta[;]$ is the Kronecker delta. Knowing the matrices C_n and C_s, the corrupted codeword $\boldsymbol{r}$, and the noise level p it is possible to apply Bayes theorem and compute the posterior probability as:

$$P(\boldsymbol{\tau} \mid \boldsymbol{z}) = \frac{1}{Z}\chi\left[\boldsymbol{z} = \boldsymbol{A}\boldsymbol{\tau}(\mathrm{mod}\ 2)\right] P(\boldsymbol{\tau}), \tag{3}$$

where $\chi[X]$ is an indicator function that is 1 if X is true and 0 otherwise. To compute the MPM one has to compute the marginal posterior $P(\tau_j \mid \boldsymbol{z}) = \sum_{\tau_{i\neq j}} P(\boldsymbol{\tau} \mid \boldsymbol{z})$. In general, this requires $\mathcal{O}(2^M)$ operations, and becomes impractical as the message size increases. To obtain a more efficient algorithm one can use the sparseness of $\boldsymbol{A}$ to design algorithms that require $\mathcal{O}(M)$ operations for performing the same computation. One of these methods is the BP algorithm, also known as probability propagation, sum-product algorithm (see [14] and references therein) or generalized distributive law [1].

Gallager's code set the general framework for the family of sparse parity check codes, it has been all but abandoned for about three decades, until MacKay and Neal [15; 16] introduced independently a code which is essentially a variation of Gallager's original code.

The MN code

MacKay and Neal [15; 16] recently introduced (independently) a variation of the Gallager's original method termed the MN code. In these codes, a message $\boldsymbol{\xi}$ is encoded into a codeword $\boldsymbol{t}$ using two randomly constructed Boolean sparse matrices C_s and C_n, which may be characterized in the following manner.

The random matrix C_s is rectangular and of dimensionality $M \times N$, having K non-zero unit elements per row and C per column. The matrix C_n is an $M \times M$ (mod 2)-invertible matrix randomly constructed having L non-zero elements per row and column. These matrices are shared by both sender and receiver.

Using these matrices, one can encode a message $\boldsymbol{\xi}$ into a codeword $\boldsymbol{t}$ in the following manner

$$\boldsymbol{t} = C_n^{-1} C_s \boldsymbol{\xi} \quad (\mathrm{mod}\ 2), \tag{4}$$

which is then transmitted via a noisy channel. Note that all matrix and vector components are of Boolean $(0, 1)$ representation, and all summations are carried out in this field, i.e., (mod 2).

During transmission, through a BSC, a noise vector $\boldsymbol{\zeta}$ is added to $\boldsymbol{t}$ and a corrupted codeword $\boldsymbol{r} = \boldsymbol{t} + \boldsymbol{\zeta}$ (mod 2) is received at the other end of the channel. Decoding is then carried out by taking the product of the matrix C_n and the received codeword $\boldsymbol{r}$, which results in the syndrome vector $\boldsymbol{z} = C_s\boldsymbol{\xi} + C_n\boldsymbol{\zeta} = C_n\boldsymbol{r}$. The main difference between these codes and Gallager's original code is that the syndrome vector contains also information on the original message. The message itself is directly estimated and there is no need for recovering the noise vector perfectly. Decoding the corrupted message in these codes can be formulated, similarly to that of Gallager's code, as finding the most probable vectors $\boldsymbol{S}$ and $\boldsymbol{\tau}$, which correspond to the signal and noise vectors respectively, that satisfy

$$C_s\boldsymbol{\xi} + C_n\boldsymbol{\zeta} = C_s\boldsymbol{S} + C_n\boldsymbol{\tau} \text{ (mod 2)}, \tag{5}$$

given the matrices C_s and C_n and the prior distributions for $\boldsymbol{S}$ and $\boldsymbol{\tau}$.

Constructions where the number of unit elements per row (K and L) and per column (C and L) is fixed are termed regular constructions, while other constructions where the number of unit elements per row/column is taken from some distribution are termed irregular. Irregular constructions generally show improved performance with respect to regular ones [17; 23; 12; 31].

In spite of the similarity between the two codes they have slightly different properties [32], in their equilibrium characteristics as well as in their dynamical behavior; these were investigated using the methods of statistical physics.

Before presenting the iterative equations derived using BP and our formulation of TAP, we would like to introduce another member of the same family of codes presented and analyzed by Sourlas [27]. Although the original code was presented within the framework of statistical physics, it can be mapped back to the framework of sparse parity-check error-correcting codes.

The code of Sourlas

Described as a parity check code, the message $\boldsymbol{\xi}$ is encoded into a codeword $\boldsymbol{t}$ using as generator a single randomly constructed Boolean sparse matrix C_s, of dimensionality $M \times N$, randomly composed of K non-zero unit elements per row and C per column.

The message $\boldsymbol{\xi}$ is encoded into a codeword $\boldsymbol{t}$ in the following manner

$$\boldsymbol{t} = C_s\boldsymbol{\xi} \quad \text{(mod 2)}, \tag{6}$$

which is then transmitted via a noisy channel and is corrupted by flip noise of probability p. Unlike Gallager/MN codes, where a syndrome vector $\boldsymbol{z}$ is generated by the receiver in a preprocessing stage, the code of Sourlas uses the corrupted codeword directly in the decoding process. Decoding may be carried out by different methods, one of which is an MPM based estimation similar to the one used in both Gallager and MN codes [9; 30].

In the reminder of the chapter we will focus on the Sourlas and MN codes. Despite the differences in the encoding and preprocessing stages; the derivation of the decoding algorithm, based on our TAP approach, is similar in the three code types, and the numerical results obtained are of a similar nature.

4 Decoding: Belief propagation

The Bayesian message estimate (MPM) potentially provides the optimal retrieval of the original messages. However, it is computationally difficult to carry out the exact calculation as it requires a sum over $\mathcal{O}(2^N)$ terms. Belief propagation [5; 22] can efficiently be used for obtaining an approximate estimate.

For brevity we will first consider the code of Sourlas; the extension of the derivation to the MN code (and Gallager's) will follow directly. The decoding process in this case relies on computing averages over the marginal posterior probability $P(S_j \mid \boldsymbol{z})$ for each of the N message bits S_j given the corrupted encoded bits z_μ (checks), where $\mu = 1 \ldots M$. The probabilistic dependencies present in the code can be represented as a bipartite graph known as *belief network* where the nodes in one layer correspond to the M checks z_μ and the nodes in the other to the N bits S_j. Each check is connected to exactly K bits and each bit is connected exactly C checks (see Fig. 6.2a).

Belief propagation is an iterative algorithm proposed by Pearl [22]; it is based on local updates of a set of marginal probabilities and the propagation of beliefs (conditional probabilities) within the network. The convergence of these iterations requires a tree like network structure with no loops. Typically, the belief networks which represent sparse parity-check error-correcting codes suffer from a significant number of loops as illustrated in Fig.6.2a. However, it has recently been shown that in some cases Pearl's algorithm provides good approximation even with the presence of loops [33]. In the particular case considered here one may also argue that the effect of loops is negligible due to the network size, which is assumed to be large and thus reduces the probability of small loops; these have the most significant effect on the accuracy of the approximation obtained.

The general framework of Pearl [22] was adapted to the specific decoding problem of sparse parity-check error-correcting codes by MacKay and Neal [15; 16]; their algorithm relies on computing the conditional probabilities $q_{\mu j}^{(S)}$ and $r_{\mu j}^{(S)}$ (not to be confused with the received vector $\boldsymbol{r}$):

$$q_{\mu j}^{(S)} = P(S_j = S \mid \{z_\nu : \nu \in \mathcal{M}(j) \backslash \mu\})$$

is the probability of the $S_j = S$ given information on all checks other than μ and

$$r_{\mu j}^{(S)} = \sum_{\mathcal{L}(\mu)\backslash j} P(z_\mu \mid S_j = S, \{S_l : l \in \mathcal{L}(\mu)\backslash j\}) \prod_{l \in \mathcal{L}(\mu)\backslash l} q_{\mu l}^{(S_l)}$$

is the probability of the check z_μ if the site j is fixed to $S_j = S$ and the contribution from the other bits involved is factorized with the related probability distributions given by $q_{\mu i}^{(S_i)}$. The sets $\mathcal{L}(\mu)$ and $\mathcal{M}(j)$ define the set of bits in the check μ and the set of checks over the bit j respectively.

Figure 6.2b provides a graphical representation of $r_{\mu j}^{(S)}$ as the total influence of

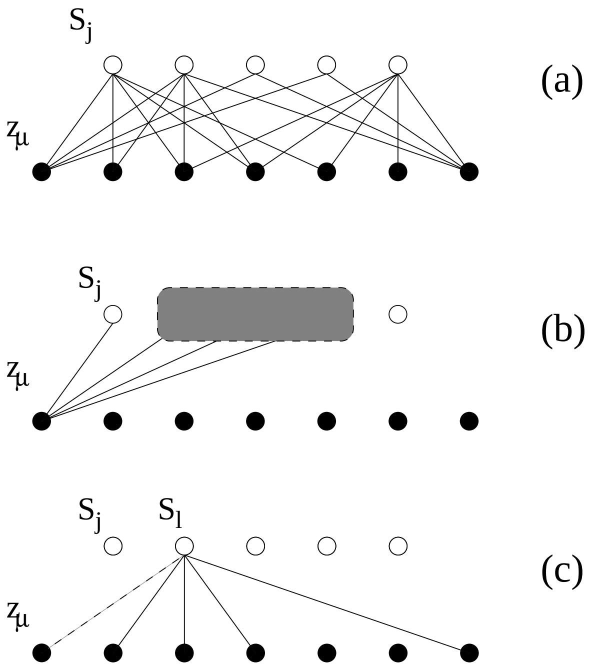

Figure 6.2
(a) Belief network representing an error-correcting code. Each bit S_j (white circles) is linked to exactly C checks (codeword bits) and each check (black circles)z_μ is linked to exactly K sites. (b) Graphical representation of the field $r_{\mu j}$. The grey box represents the mean field contribution $\prod_{l \in \mathcal{L}(\mu) \setminus j} q_{\mu l}$ of all bits other than S_j on the check (codeword bit) z_μ. (c) Representation of one of the fields $q_{\mu l}$ in (b).

the bit S_j and a local mean field $\prod_{l \in \mathcal{L}(\mu) \setminus l} q_{\mu l}^{(S)}$ (representing factorized contribution from the other sites in $\mathcal{L}(\mu)$) on the check z_μ. Figure 6.2c represents graphically the field $q_{\mu l}^{(S)}$ as the influence of the checks in $\mathcal{M}(l)$ excluding μ on the bit S_l, this exclusion is required for avoiding loops in the network.

Employing Bayes theorem $q_{\mu j}^{(S)}$ can be rewritten as:

$$q_{\mu j}^{(S)} = a_{\mu j}\, P(\{z_\nu : \nu \in \mathcal{M}(j) \setminus \mu\} \mid S_j)\, p_j^{(S)}, \tag{7}$$

where $a_{\mu j}$ is a normalization constant such that $q_{\mu j}^{(0)} + q_{\mu j}^{(1)} = 1$ and $p_j^{(S)}$ is the prior probability over the bit j. The distribution $P(\{z_\nu : \nu \in \mathcal{M}(j) \setminus \mu\} \mid S_j)$ can be replaced by a mean field approximation in a way that factorizes the dependencies using the fields $r_{\mu j}^{(S)}$, obtaining

$$\begin{aligned} q_{\mu j}^{(S)} &= a_{\mu j}\, p_j^{(S)} \prod_{\nu \in \mathcal{M}(j) \setminus \mu} r_{\nu j}^{(S)} \\ r_{\mu j}^{(S)} &= \sum_{\mathcal{L}(\mu) \setminus j} P(z_\mu \mid S_j = S, \{S_i : i \in \mathcal{L}(\mu) \setminus j\}) \prod_{i \in \mathcal{L}(\mu) \setminus j} q_{\mu i}^{(S_i)}. \end{aligned} \tag{8}$$

An estimate $\widehat{\xi}_j = \mathrm{argmax}_{S \in \{0,1\}} \left\{ q_j^{(S)} \right\}$ of the original message bits is obtained by solving the above equations, what can be done iteratively using several different schedules, the efficiency of which depends on the particular topology of the network [1]; and computing the pseudo-posterior:

$$q_j^{(S)} = a_j p_j^{(S)} \prod_{\nu \in \mathcal{M}(j)} r_{\nu j}^{(S)}, \tag{9}$$

where a_j is a normalization constant.

Notice that the field $r_{\mu j}^S$ is not originally normalized with respect to bit variables S while $q_{\mu j}^S$ is the case. However, one may introduce an extra normalization condition $r_{\mu j}^{(0)} + r_{\mu j}^{(1)} = 1$ without changing any result. By taking advantage of this extra condition, one can reduce the set of equations to $\delta q_{\mu j} = q_{\mu j}^{(0)} - q_{\mu j}^{(1)}$ and $\delta r_{\mu j} = r_{\mu j}^{(0)} - r_{\mu j}^{(1)}$. The pseudo posterior can be calculated in this manner obtaining an estimate to the original message bits following a rule

$$\widehat{\xi}_j = \begin{cases} 0, & \text{if } \delta q_j > 0, \\ 1, & \text{otherwise.} \end{cases} \tag{10}$$

Extending the formulation to both Gallager and MN codes is straightforward, as after preprocessing these codes also involve a decoding task with very sparse matrices; in the latter case one extends the set of dynamical variables to include both signal and noise vectors [15; 16].

This algorithm has been employed in a variety of decoding scenarios for both parity-check codes and turbo codes [16; 5] proving to be highly efficient.

5 Decoding: the TAP approach

So far we have described the sparse parity check coding scheme using the conventional Boolean $(0, 1)$ representation. However, in order to apply methods of statistical physics, it is highly convenient to introduce an equivalent representation using binary variables ± 1. More specifically, we hereafter convert all the Boolean variables to the binary ones, by employing the isomorphism

$$\begin{array}{ccc} \textbf{Boolean} & \leftrightarrow & \textbf{binary} \\ (0, 1, +) & & (+1, -1, \times). \end{array} \tag{11}$$

One can easily check the equivalence between these two groups by observing the following simple isomorphic map:

$$(-1)^{x+y+\ldots+z \ (\mathrm{mod}\ 2)} = X \times Y \times \ldots \times Z, \tag{12}$$

where $x, y, \ldots, z$ are the Boolean $(0, 1)$ variables while $X = (-1)^x, Y = (-1)^y, \ldots,$ $Z = (-1)^z$ are the corresponding binary (± 1) ones.

Mapping to an Ising Spin System

Two advantages in the novel representation are worthwhile mentioning. The first is the compactness of the description. For example, one can describe the conditional probabilities standing for the transmission through a BSC in a simple manner as

$$P(r|t) = \frac{1+\rho rt}{2} = \frac{\exp\left[\beta_n rt\right]}{2\cosh(\beta_n)}, \tag{13}$$

in the binary representation, where $t(\in[-1,+1])$ and $r(\in[-1,+1])$ are the transmitted and received message bits respectively, p is the flip probability of the channel and $\rho = 1-2p$ and $\beta_n = (1/2)\ln\left[(1-p)/p\right]$. In particular, the last term on the right in Eq. (13) makes calculations like those in Eqs. (8) easier to handle as one can convert the product operations to simple summations.

In addition, the adoption of the binary representation makes the similarity to Ising spin models explicit, enabling one to take advantage of the techniques developed in statistical physics for analysing such systems. Employing an expression like the one on the right hand side of Eq. (13) for the distributions of binary variables, one can generally represent posterior probabilities after finding the syndrome $\boldsymbol{z}$ (the received message itself as in Sourlas' code or the preprocessed vector as in Gallager/MN codes)

$$P(\boldsymbol{S} \mid \boldsymbol{z}) = \frac{\exp\left[-\beta H(\boldsymbol{S}|\boldsymbol{z})\right]}{Z(\boldsymbol{z})}, \tag{14}$$

with

$$\beta H(\boldsymbol{S}|\boldsymbol{z}) = -\beta \sum_{\mu=1}^{M} z_\mu \prod_{l\in\mathcal{L}(\mu)} S_l - F\sum_{l=1}^{N} S_l, \tag{15}$$

where $Z(\boldsymbol{z}) = \mathrm{Tr}_{\boldsymbol{S}} \exp[-\beta H(\boldsymbol{S}|\boldsymbol{z})]$, β and F are hyper-parameters determined by the type of codes, the channel noise and the prior distribution of messages. Parity check codes can be generally mapped onto Ising spin systems with multi-spin interactions described by a Hamiltonian of the type (15) facilitating the use of methods developed in physics for analysing the current system [27; 28; 20; 24; 9; 10; 8].

In this context, our formulation of the decoding problem is strongly linked to the Bethe [2] approximation and its extensions [32], and to the conventional TAP approach [29]. In [9] we have shown that this framework provides a similar set of iterative equations to that of BP.

The motivation for developing this formulation is the excellent approximation provided by the Bethe lattice approach for finitely connected systems in the thermodynamic limit [26]. Finite loops linking the different network sites vanish as the system size grows and can be neglected without introducing significant errors in this scenario. The approximation used also has mean field properties in the way one takes into account the mean influence of the whole lattice on a particular site.

Due to the transparency of the derivation in this case, we start by explaining the TAP formulation for the code of Sourlas.

The code of Sourlas

To develop the new approach we notice that the likelihood $P(z_\mu \mid S)$ is proportional to the Boltzmann weight, for a given inverse temperature $\beta(= 1/T)$:

$$w_B(z_\mu \mid \boldsymbol{S}) = \exp\left(-\beta z_\mu \prod_{i\in\mathcal{L}(\mu)} S_i\right) , \tag{16}$$

that can be rewritten in the more convenient form:

$$w_B(z_\mu \mid \boldsymbol{S}) = \frac{1}{2}\cosh(\beta z_\mu) \times \left(1 \ + \ \tanh(\beta z_\mu) \prod_{j\in\mathcal{L}(\mu)} S_j\right) . \tag{17}$$

In fact, the inverse temperature β has an optimal value given by Nishimori's temperature $\beta_n = (1/2)\ln[(1-p)/p]$ [20] if the flip probability p in BSC is known. However, we deal with it as a control parameter in order to consider general situations where p is not exactly known to receiver.

The conditional probability $r_{\mu j}^{(S_j)}$ can then be seen as an normalized effective Boltzmann weight (effective Boltzmann probability)

$$\begin{aligned} r_{\mu l}^{(S_l)} &= a_{\mu l}\ w_{\text{eff}}(z_\mu \mid S_l, \{z_{\nu\neq\mu}\}) \\ &= a_{\mu l}\ \text{Tr}_{\{S_{k\neq l}\}}\ w_B\left(z_\mu \mid \boldsymbol{S}\right) \prod_{k\neq l} q_{\mu l}^{S_k} \end{aligned} \tag{18}$$

obtained by taking the connection μ out of the system, and taking into consideration the (factorized) dependence of the variables $\boldsymbol{S}$ on all other connections $(q_{\mu j}, \forall j)$; $a_{\mu l}$ being a normalization coefficient. The term $q_{\mu j}$ is identified as the mean field contribution to a specific site, from which the first of Eqs.(8) follows directly. Plugging the form (17) for the likelihood in the equations (8), using the fact that the prior probability is given by $p_j^{(S)} = \frac{1}{2}\left(1 + \tanh(\beta SF)\right)$ (which constitutes the definition of F) and computing $\delta q_{\mu j}$ and $\delta r_{\mu j}$ we find:

$$\begin{aligned} \delta r_{\mu j} &= \tanh(\beta z_\mu) \prod_{l\in\mathcal{L}(\mu)\setminus j} \delta q_{\mu l} \\ \delta q_{\mu j} &= \tanh\left(\sum_{\nu\in\mathcal{M}(j)\setminus\mu} \tanh^{-1}(\delta r_{\nu j}) + \beta F\right) . \end{aligned} \tag{19}$$

Solving these equations iteratively enables one to derive the pseudo-posterior through the expression:

$$\delta q_j = \tanh\left(\sum_{\nu\in\mathcal{M}(j)} \tanh^{-1}(\delta r_{\nu j}) + \beta F\right) , \tag{20}$$

This provides a way for computing the Bayes optimal decoding $\widehat{\xi}_j = \text{sign}(\delta q_j)$. It is interesting to note that the somewhat arbitrary use of the differences $\delta q_{\mu l} = \langle S_l^\mu\rangle_q$ and $\delta r_{\mu l} = \langle S_l^\mu\rangle_r$ in the BP approach becomes clear form the TAP formulation, where they represent the expectation values of the dynamical variables with respect to the fields.

It is important at this point to list and interpret the mean field assumptions used here [9]:

1. We assume a mean field behavior for the dependence of the dynamical variables $\boldsymbol{S}$ on a certain realization of the message sites $\boldsymbol{z}$, i.e., the posterior distribution is factorizable with respect to dynamical variables $S_{i=1,\ldots,N}$ and may be replaced by a product of mean fields.
2. Boltzmann weights for a specific site S_l are factorizable with respect to the message sites z_μ.
3. The contribution of single variables $S_{i=1,\ldots,N}$, and $z_{\mu=1,\ldots,M}$ to the macroscopic variables is small and can be isolated.

The factorizability of the probability distributions provides a good approximation due to the absence of short loops in the lattice and by the cluster property:

$$\lim_{N\to\infty} \frac{1}{N^2} \sum_{\forall i\neq j} \left(\langle S_i S_j \rangle_{p(\boldsymbol{S}|\boldsymbol{z})} - \langle S_i \rangle_{p(\boldsymbol{S}|\boldsymbol{z})} \langle S_j \rangle_{p(\boldsymbol{S}|\boldsymbol{z})} \right)^2 \to 0 \tag{21}$$

that the bits S_j are supposed to obey within a pure state [18].

The MN codes

The derivation presented above can be easily extended to the case of MN codes. In this case one treats both variable types (signal and noise, $\boldsymbol{S}$ and $\boldsymbol{\tau}$ respectively) on equal footing, aiming to calculate the marginal posterior probabilities

$$P(S_i|\boldsymbol{z}') = \underset{\{\{S_{k\neq i}\},\boldsymbol{\tau}\}}{\mathrm{Tr}} P(\boldsymbol{S},\boldsymbol{\tau}|\boldsymbol{z}')$$

and

$$P(\tau_j|\boldsymbol{z}') = \underset{\{\boldsymbol{S},\{\tau_{k\neq j}\}\}}{\mathrm{Tr}} P(\boldsymbol{S},\boldsymbol{\tau}|\boldsymbol{z}')$$

based on similar three assumptions, as in the case of Sourlas, including both $\boldsymbol{S}$ and $\boldsymbol{\tau}$. Here, we denote $\boldsymbol{z}'$ as the binary equivalent to the Boolean syndrome $C_n\boldsymbol{z}$ computed in MN codes.

From a statistical physics point of view, the main difference between the current codes and those of Sourlas is the temperature at which the codes are appropriately mapped onto Ising spin systems. Since condition (5) is introduced to posterior distribution through an indicator function as

$$\chi\left[C_n\boldsymbol{z} = C_s\boldsymbol{S} + C_n\boldsymbol{\tau} \ (\mathrm{mod}\ 2)\right] = \lim_{\beta\to\infty} \frac{\exp\left[\beta \sum_{\mu=1}^{M} z'_\mu \prod_{k\in\mathcal{L}_s(\mu)} S_k \prod_{j\in\mathcal{L}_n(\mu)} \tau_j\right]}{(2\cosh\beta)^M}, \tag{22}$$

in the binary representation, the MN codes are mapped onto Ising models with a new effective temperature β^{-1} which is set to be zero constraining the space of configurations to those obeying the constraints defined by $\chi[.]$. Here, we have introduced notations $\mathcal{L}_s(\mu)$ and $\mathcal{L}_n(\mu)$ in order to denote the set of all indices of non-zero components in μ-th row of the sparse matrix C_s and C_n, respectively.

In the statistical physics community, it is widely known that Ising spin systems with quenched disorder can be highly frustrated at low temperatures, which makes efficient computation by mean field approximations infeasible. However, it should be stressed here that the interactions described by Eq. (22) produce no frustration in the current system even at the effective temperature β^{-1} set to zero because this model is flat [18], *i.e.* the disorder can be trivially gauged and there are more $(M+N)$ dynamical variables than the number of constraints (M).

In addition to these constraints, prior knowledge about the message and noise vectors $\boldsymbol{S}$ and $\boldsymbol{\tau}$ is also taken into account by introducing the following prior distributions

$$P(\boldsymbol{S}) = \frac{\exp[F_s \sum_{l=1}^{N} S_l]}{(2\cosh F_s)^N}, \quad P(\boldsymbol{\tau}) = \frac{\exp[F_n \sum_{j=1}^{M} \tau_j]}{(2\cosh F_n)^M}, \tag{23}$$

where $F_n = (1/2)\ln[(1-p)/p]$ is set to its optimal value, and non-zero field F_s is introduced for biased messages. These $\mathcal{O}(1)$ fields work to compensate the insufficiency of conditions for uniquely determining each bit sequence without causing frustration. This effect becomes stronger for larger F_n and F_s. Therefore, one can expect that for sufficiently large F_n, F_s, which implies sufficiently small flip rate p if the message is not biased, unique bit sequences $\boldsymbol{S}$ and $\boldsymbol{\tau}$ can be determined by the posterior distribution and can possibly be computed effectively by the TAP approach.

Parameterizing pseudo-marginal posteriors and marginalized conditional probabilities as

$$\begin{aligned} P(S_i|\{z'_{\nu\neq\mu}\}) &= \frac{1+\delta q^S_{\mu i} S_i}{2}, \\ P(\tau_j|\{z'_{\nu\neq\mu}\}) &= \frac{1+\delta q^n_{\mu j} \tau_j}{2}, \end{aligned} \tag{24}$$

$$\begin{aligned} P(z'_\mu|S_i,\{z'_{\nu\neq\mu}\}) &\sim \frac{1+\delta r^S_{\mu i} S_i}{2}, \\ P(z'_\mu|\tau_j,\{z'_{\nu\neq\mu}\}) &\sim \frac{1+\delta r^n_{\mu j} \tau_j}{2}, \end{aligned} \tag{25}$$

a set of self-consistent equations can be found [9; 11; 30]

$$\begin{aligned} \delta q^S_{\mu l} &= \tanh\left(F_s + \sum_{\nu\in\mathcal{M}_S(l)/\mu} \tanh^{-1}(\delta r^S_{\nu l})\right), \\ \delta q^n_{\mu l} &= \tanh\left(F_n + \sum_{\nu\in\mathcal{M}_n(l)/\mu} \tanh^{-1}(\delta r^n_{\nu l})\right), \end{aligned} \tag{26}$$

and

$$\begin{aligned} \delta r^S_{\mu l} &= z'_\mu \prod_{k\in\mathcal{L}_S(\mu)/l} \delta q^S_{\mu k} \prod_{j\in\mathcal{L}_n(\mu)} \delta q^n_{\mu j}, \\ \delta r^n_{\mu l} &= z'_\mu \prod_{k\in\mathcal{L}_S(\mu)} \delta q^S_{\mu k} \prod_{j\in\mathcal{L}_n(\mu)/l} \delta q^n_{\mu j}, \end{aligned} \tag{27}$$

similar to those obtained in the case of Sourlas code (19). Here, $\mathcal{M}_s(l)$ and $\mathcal{M}_n(l)$ indicate the set of all indices of non-zero components in the l-th column of the sparse matrix C_s and C_n, respectively. The notation $\mathcal{M}_s(l)/\mu$ represents the set of all indices belonging to $\mathcal{M}_s(l)$ except l and similar for the others.

Equations (26) and (27) are solved iteratively using appropriate initial conditions. After obtaining a solution to all $\delta q_{\mu l}$ and $\delta r_{\mu l}$, an approximated posterior mean can be calculated as

$$\delta q_i^S = \tanh\left(F_s + \sum_{\mu\in\mathcal{M}_S(l)} \tanh^{-1}(\delta r_{\mu i}^S)\right), \tag{28}$$

which provides an approximation to the Bayes-optimal estimator of the form $\hat{\xi}^B = \mathrm{sign}(\delta q_i^S)$.

By introducing the new variables $x_i = \xi_i \delta q_{\mu i}^S$, $\hat{x}_i = \xi_i \delta r_{\mu i}^S$, $y_j = \zeta_j \delta q_{\mu j}^n$ and $\hat{y}_j = \zeta_j \delta r_{\mu j}^n$ and assuming that they are independently drawn from distributions $\pi(x)$, $\hat{\pi}(\hat{x})$, $\rho(y)$ and $\hat{\rho}(\hat{y})$ (an assumption that has been verified experimentally), one can link the equations (26,27) to those obtained using the replica method [19; 11]. This connection can be extended further by providing an expression for the TAP free energy which equations(26) and (27) extremize

$$\begin{aligned}
f_{\mathrm{TAP}}\left(\{\delta q\},\{\delta r\}\right) &= \frac{M}{N}\ln 2 + \frac{1}{N}\sum_{\mu=1}^{M}\sum_{i\in\mathcal{L}_S(\mu)} \ln\left(1+\delta q_{\mu i}^S \delta r_{\mu i}^S\right) \\
&+ \frac{1}{N}\sum_{\mu=1}^{M}\sum_{j\in\mathcal{L}_n(\mu)} \ln\left(1+\delta q_{\mu j}^n \delta r_{\mu j}^n\right) \\
&- \frac{1}{N}\sum_{\mu=1}^{M}\ln\left(1+z'_\mu \prod_{i\in\mathcal{L}_S(\mu)}\delta q_{\mu i}^S \prod_{j\in\mathcal{L}_n(\mu)}\delta q_{\mu j}^n\right) \\
&- \frac{1}{N}\sum_{i=1}^{N}\ln\left[e^{F_s}\prod_{\mu\in\mathcal{M}_S(i)}\left(1+\delta r_{\mu i}^S\right) + e^{-F_s}\prod_{\mu\in\mathcal{M}_S(i)}\left(1-\delta r_{\mu i}^S\right)\right] \\
&- \frac{1}{N}\sum_{j=1}^{M}\ln\left[e^{F_n}\prod_{\mu\in\mathcal{M}_n(j)}\left(1+\delta r_{\mu j}^n\right) + e^{-F_n}\prod_{\mu\in\mathcal{M}_n(j)}\left(1-\delta r_{\mu j}^n\right)\right] .
\end{aligned} \tag{29}$$

This expression may be used for selecting the best estimate when Eqs.(26) and (27) have several solutions.

This derivation can be easily extended to accommodate Gallager's code.

6 Experimental results

As our TAP formulation arrives at exactly the same iterative equations as those obtained using BP, we briefly presents a couple of examples demonstrating the efficacy of the method for decoding corrupted messages encoded using the MN and Sourlas error correcting codes.

We first presented some experiments using the code of Sourlas and equations (19). In these experiments we used the closed set of iterative equations to decode messages encoded by the code of Sourlas and corrupted during transmission by flip noise of probability p. For each run, a fixed code is used to generate 20000 bit codewords from 10000 bit messages; corrupted versions of the codewords are then decoded using (19). For each trial we monitor the overlap between the decoded vector and the original message (magnetization) $m = 1/N \ \sum_{i=1}^{N} \xi_i \widehat{\xi}_j$ (in binary). Numerical solutions for 10 individual runs are presented in Fig.6.3a, the initial conditions are chosen as $\delta r_{\mu l} = 0$ and $\delta q_{\mu l} = \tanh(\beta F)$ reflecting prior beliefs for both signal and noise. In figure 6.3 we show results for $K = 2$ and $C = 4$, corresponding to a code rate $R = 1/2$, in the unbiased case (prior probability $P(\xi_j = 1) = f_s = 0.5, \ \forall j)$ at a temperature as low as $T = 0.26$. We also show the agreement between the results obtained and those coming from the replica symmetric calculation [9; 30]. In the same figure we show the performance for the case of biased examples ($P(\xi_j = 1) = f_s = 0.1 \ \forall j$). Again the agreement with results obtained using the replica method [9; 30] is rather convincing. The third curve in the Fig.6.3a shows the performance for biased messages at Nishimori's temperature, $1/T_n = 1/2 \ln[(1-p)/p]$ [20], as expected [24; 28; 21; 8] it is superior to low temperature decoding, being equivalent to having the correct prior in the Bayesian framework. The agreement with the replica based results is even better.

In Fig.6.3b we show results for $K = 5$ and $C = 10$, again the code rate is $R = 0.5$. For unbiased messages the system is extremely sensitive to initial conditions and does not perform well on average even at Nishimori's temperature, ending up in some sub-optimal solution. For biased messages the results are far better and in agreement with the replica based results [9; 30].

Applying the same algorithm to the case of regular and irregular MN codes we obtain the results presented in figure 6.4, demonstrating the improvement in performance achieved by simple irregularity in the construction. The irregularity used is based on the following probability distribution, from which the (column) connectivities of the signal matrix $\boldsymbol{C_s}$ are derived:

$$\mathcal{P}_C(C) = (1-\theta)\, \delta(C-C_o) \, + \, \theta \, \delta(C-C_e). \tag{30}$$

The mean connectivity is $\overline{C} = (1-\theta)\ C_o \ + \ \theta\ C_e$ and $C_o < \overline{C} < C_e$ and the noise matrix $\boldsymbol{C_n}$ is chosen to be regular.

To gain some insight on the effect of irregularity on solving the TAP/BP equations (26) and (27) we performed several runs starting from the fixed initial conditions $\delta q^s_{\mu j}(0) = 1 - 2f_s$ and $\delta q^n_{\sigma l}(0) = 1 - 2p$. For comparison we obtained analytical solutions based on the replica symmetric theory [31].

In Figure 6.4 we show a typical curve for the magnetization as a function of the noise level. The analytical results agree very well with TAP/BP decoding results, indicating that the addition of irregularity improves the performance considerably.

7 Summary

In this paper we discuss the application of our TAP formulation to the decoding problem in sparse parity-check error-correcting codes. We show that using simple

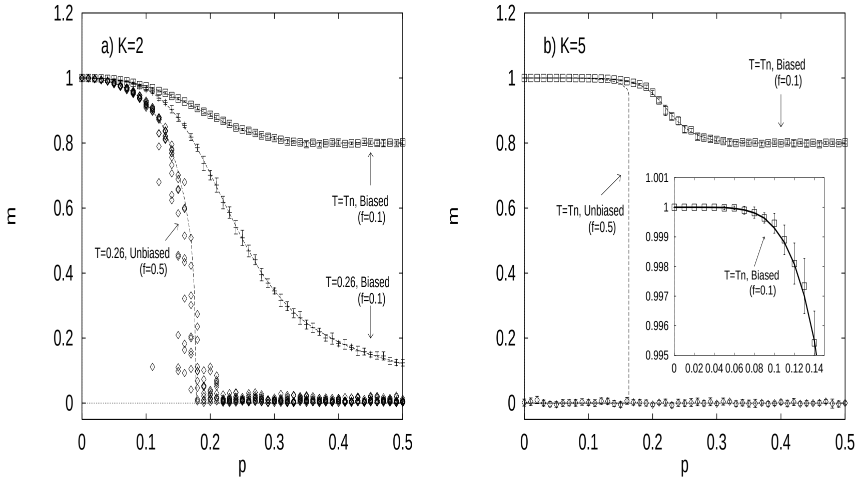

Figure 6.3
The overlap (magnetization) obtained from numerical solutions for different flip rate p. (a) For the case $K=2$, different biases ($f_s=P(\xi_j=1)=0.1, 0.5\ \forall j$) and temperatures ($T=0.26, T_n$), we see good agreement between the TAP/BP solutions and the theoretical values [9; 30]. Results for the unbiased patterns are shown as raw data, i.e., results of 10 runs for each flip rate value p (diamond), while the theoretical solution is marked by the dashed line. Results for biased patterns are shown by their mean and standard deviation, showing a suboptimal improvement in performance as expected for $T=0.26$ and an optimal one at Nishimori's temperature -T_n. Note that in the case of $T=T_n$ the standard deviation is significantly smaller than the symbol size. Figure (b) shows results for the case $K=5$ and $T=T_n$ in similar conditions to (a). Also here iterative solutions may generally drift away from the theoretical values where temperatures other than T_n are employed (not shown); using Nishimori's temperature alleviates the problem only in the case of biased messages and the results are in close agreement with the theoretical solutions (focusing on low p values in the inset).

mean field arguments and interpreting the effective Boltzmann weight as the local site conditional probability, once a single connection has been taken out of the system, one retrieves the same iterative equations obtained from the BP method.

We employ the TAP/BP iterative equations for decoding corrupted messages, encoded using the MN codes, and the code of Sourlas, in particular scenarios. We compared the results obtained to the analytical solutions obtained by the replica method. In the case of Sourlas, the solutions indicate that the method is particularly useful in the case of biased messages and that using Nishimori's temperature is highly beneficial; solutions obtained using other temperature values may be sub-optimal. For unbiased messages and $K\geq 3$ we may obtain erroneous solutions using these methods.

The TAP/BP approach is extremely useful in the case of MN codes where, below a certain corruption level, they converge to the solution which shows excellent retrieval of the original vector [11; 19; 31]. Above this point the algorithm tend to converge to sub-optimal solutions, but this is due to the inherent limitation of the constructions rather than a failure of the decoding algorithm. In the current chapter we used the TAP/BP approach to show the improvement in performance emerging

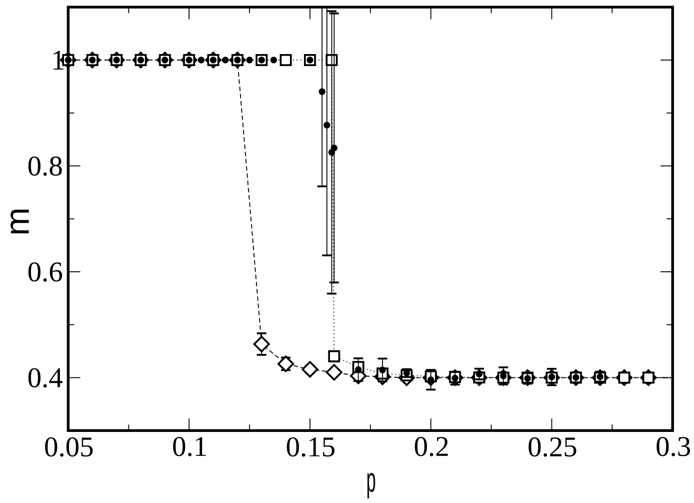

Figure 6.4
Overlap (magnetization) as a function of the noise level p for codes with $K = L = 3$ and $\overline{C} = 15$ with message bias $f_s = 0.3$. Analytical solutions for the regular code are denoted as ◊ and for the irregular code, with $C_o = 4$ and $C_e = 30$, as □. Simulation results are averaged over 10 runs of the TAP/BP algorithm in an irregular construction with message length of $N = 6000$, starting from fixed initial conditions (see the text); they are plotted as ● in the rightmost curve for comparison. TAP/BP results for the regular case agree with the theoretical solutions and have been omitted to avoid overloading the figure.

from the introduction of irregularity in the matrix construction.

It would be interesting to utilize more refined approximation techniques, adopted from the statistical physics literature, to find better coding/decoding schemes, evaluating the trade off between performance improvement obtained and the increasing computational costs.

Acknowledgments

This work was partially supported by the program "Research For the Future" (RFTF) of the Japanese Society for the Promotion of Science (YK), by EPSRC grant GR/N00562, and a Royal Society travel grant (DS,RV).

References

[1]Aji, S.M., and McEliece, R.J., IEEE Trans. Info. Theory, **46**:325, 2000.

[2]Bethe, H.A., Proc. R. Soc. London, Ser A, **151**:552, 1935.

[3]Berrou, C., Glavieux, A., and Thitimajshima, P., in proceedings of the 1993 IEEE Intl. Conference on Communications, Geneva Switzerland 1064, 1993; Berrou, C., and Glavieux, A., IEEE Trans.Comm., **44**:1261, 1996.

[4]Cover, T.M., and Thomas, J.A., Elements of Information Theory (Wiley, New York), 1991.

[5]Frey, B.J., Graphical Models for Machine Learning and Digital Communication (MIT Press, Cambridge, MA.), 1998.

[6]Gallager, R.G., IRE Trans. Info. Theory **8**:21, 1962.

[7]Gallager, R.G., Low Density Parity Check Codes (MIT Press, Cambridge, MA.), 1963.

[8]Iba, Y., J. Phys. A: Math. and Gen. **32**:3875, 1999.

[9]Kabashima, Y., and Saad, D., Europhys. Lett. **44**:668, 1998.

[10]Kabashima, Y., and Saad, D., Europhys. Lett. **45**:97, 1999.

[11]Kabashima, Y., Murayama, T., and Saad, D., Phys. Rev. Lett. **84**:1355, 2000.

[12]Kanter, I., and Saad, D., Phys. Rev. Lett. **83**:2660, 1999.

[13]Kanter, I., and Saad, D., J. Phys. A: Math. and Gen. **33**:1675, 2000.

[14]Kschischang, F.R. and Frey, B.J. IEEE J. Select. Areas in Comm. **16**:153, 1998.

[15]MacKay, D.J.C., and Neal, R.M., Electr. Lett. **32**:1645, 1996.

[16]MacKay, D.J.C., IEEE Trans. Info. Theory **45**:399, 1999.

[17]MacKay, D.J.C., Wilson, S.T. and Davey, M.C., IEEE Trans.Comm., **47**:1449, 1999.

[18]Mezard, M., Parisi, G., and Virasoro, MA., Spin Glass Theory and Beyond (World Scientific, Singapore), 1987.

[19]Murayama, T., Kabashima, Y., Saad, D., and Vicente, R., Phys Rev. E. **62**:1577, 2000.

[20]Nishimori, H., J. Phys. C: Solid State Phys. **13**:4071, 1980; Prog. Theor. Phys. **69**:1169, 1981.

[21]Nishimori, H., J. Phys. Soc. Jpn. **62**:2793, 1993.

[22]Pearl, J., Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (Morgan Kaufmann, San Francisco, CA), 1988.

[23]Richardson, T., Shokrollahi, A., and Urbanke, R., Design of Provably Good Low-Density Parity Check Codes, http://cm.bell-labs.com/cm/ms/who/tjr/pub.html, 1999.

[24]Ruján, P., Phys. Rev. Lett. **70**:2698, 1993.

[25]Shannon, C.E., Bell Sys.Tech.J. **27**:379 and 623, 1948.

[26]Sherrington, D., and Wong, K.Y.M., J. Phys A **20**:L785, 1987.

[27]Sourlas, N., Nature **339**:693, 1989.

[28]Sourlas, N., Europhys. Lett. **25**:159, 1994.

[29]Thouless, D.J., Anderson, P.W., and Palmer, R.G., Phil. Mag. **35**:593, 1977.

[30]Vicente, R., Saad, D. and Kabashima, Y., Phys. Rev. E **60**:5352, 1999.

[31]Vicente, R., Saad, D. and Kabashima, Y., J. Phys. A, **33**:1527, 2000.

[32]Vicente, R., Saad, D., and Kabashima, Y., Europhys. Lett., **51**:698, 2000.

[33]Weiss, Y., Neural Computation **12**:1, 2000.