STATISTICAL MECHANICS APPROACH TO ERROR-CORRECTING CODES

Nicolas Sourlas

Laboratoire de Physique Théorique de l' Ecole Normale Supérieure *
24 rue Lhomond, 75231 Paris CEDEX 05, France.
e-mail: sourlas@lpt.ens.fr

Abstract: I will review the relationship between error-correction codes and certain mathematical models of spin glasses. I will show that there is a one to one relationship between error correcting codes and spin glass models. Minimum error probability decoding is equivalent to finding the magnetisation of the corresponding spin system. Convolutional codes correspond to one-dimensional spin systems and Viterbi's decoding algorithm to the transfer matrix algorithm of Statistical Mechanics.

I will also show how the recently discovered (or rediscovered) capacity approaching codes (turbo codes and low density parity check codes) can be analysed using statistical mechanics. Turbo codes correspond to two coupled spin chains, while low density parity check codes are spin models on a diluted random graph. It is possible to show, using statistical mechanics, that these codes allow error-free communication for signal to noise ratio above a certain threshold. This threshold, which corresponds to a phase transition in the spin model, depends on the particular code, and can be computed analytically in many cases.

LPTENS 02/ 23

The mathematical theory of communication[1, 2] is probabilistic in nature. Both the production of information and its transmission are considered as probabilistic events. A source is producing information messages according to a certain probability distribution. Each message consists of a sequence of K bits $\vec{\sigma} = \{\sigma_1, \cdots, \sigma_K\}$, $\sigma_i = \pm 1$ and it is assumed that the probability $P_s(\vec{\sigma}) \equiv \exp -H_s(\vec{\sigma})$ of any particular sequence $\vec{\sigma}$ is known. According to Shannon the information content of the message is $-\ln P_s(\vec{\sigma})$ and the average information of the source is given by

$$-\sum_{\vec{\sigma}} P_s(\vec{\sigma}) \ln P_s(\vec{\sigma})$$

* UMR 8549, Unité Mixte de Recherche du Centre National de la Recherche Scientifique et de l' Ecole Normale Supérieure.

The messages are sent through a transmission channel. In general there is noise during transmission (which may have different origins) which corrupts the transmitted message. If a $\sigma = \pm 1$ is sent through the transmission channel, because of the noise, the output will be a real number J, in general different from σ. Again, the statistical properties of the transmission channel are supposed to be known. Because of the noise during the transmission, there is a loss of information. The channel capacity $\mathcal{C}$ is defined as the maximum information per unit time which can be transmitted through the channel. The maximum is taken over all possible sources.

For reasons of simplicity, we will assume in the following that all the source symbols are statistically independent and that the noise is independent for any pair of bits ("memoryless channel"). In the case of a memoryless channel and of a gaussian noise, Shannon[1, 2] calculated the channel's capacity

$$\mathcal{C} = \frac{1}{2} \log_2 (1 + \frac{v^2}{w^2})$$

where v^2/w^2 is the signal to noise power ratio.

Under the above assumptions, communication is a statistical inference problem. Given the transmission channel's output and the statistical properties of the source and of the channel, one has to infer what message was sent. In order to reduce communication errors, one may introduce (deterministic) redundancy into the message ("channel encoding") and use this redundancy to infer the message sent through the channel ("decoding"). The algorithms which transform the source outputs to redundant messages are called error-correcting codes. The inverse of the redundancy (see later for a precise definition) is called the rate R of the code.

The famous Shannon coding theorem[1, 2] states that for infinite long messages, it is possible to communicate error free, provided the rate of the code is smaller than the channel capacity. For practical purposes it is also required that the computational complexity of the code (the amount of computation required both for encoding and decoding) is not very large. It must be possible to encode and decode in a reasonable amount of time. A code which is very good for very long messages of length N but requires an exponential in N decoding time is obviously not very interesting.

Until recently there were no known codes of reasonable computational complexity allowing communication with a very small error, for noise level not too far from capacity. This situation changed drastically with the recent discoveries of the "capacity approaching" codes. First came the discovery of turbo codes by Berrou and Glavieux[3] and later the rediscovery of low density parity check codes[4], first discovered by Gallager[5, 6], in his thesis, in 1962. Both turbo codes and low density parity check (LPDC) codes are based on random constructions. Because of this randomness, it is not easy to analyse them with the traditional methods of communication theory.

I have shown some time ago$[7-10]$ that there is a mathematical eqivalence of error-correcting codes to some theoretical spin-glass models.

I will explain later that it is possible to use this equivalence with spin glasses, to study the properties of these capacity approaching codes using the methods of statistical mechanics developped in the study of disordered systems.

Let me start by fixing the notations. Each information message consists of a sequence of K bits $\vec{u} = \{u_1, \cdots, u_K\}, u_i = 0$ or 1. The binary vector $\vec{u}$ is called the source-word. Encoding introduces redundancy into the message. One maps $\vec{u} \to \vec{x}$ by encoding. $\vec{u} \to \vec{x}$ has to be a one to one map for the code to be meaningful. The binary vector $\vec{x}$ has $N > K$ components. It is called a code-word. The ratio $R = K/N$ which specifies the redundancy of the code, is called the rate of the code. One particularly important family of codes are the so-called linear codes. Linear codes are defined by

$$\vec{x} = G\vec{u}$$

G is a binary (i.e; its elements are zero or one) $(N \times K)$ matrix and the multiplication is modulo two. G is called the generating matrix of the code. Obviously by construction all the components x_i of a code-word x are not independent. Of all the 2^N binary vectors only $2^K = 2^{NR}$, those corresponding to a vector $\vec{u}$, are codewords. Codewords satisfy the linear constraints (called parity check constraints) $H\vec{x} = 0$ (modulo two), where H is a $(K \times N)$ binary matrix, called the parity check matrix. The connection with spin variables is straightforward. $u_i \to \sigma_i = (-1)^{u_i}$, $x_i \to J_i = (-1)^{x_i}$. It follows that $u_i + u_j \to \sigma_i \sigma_j$ and

$$J_i = (-1)^{\sum_j G_{ij} u_j} = C^i_{k_1 \cdots k_i} \sigma_{k_1} \cdots \sigma_{k_i} \tag{1}$$

The previous equation defines the "connectivity matrix " C in terms of the generating matrix of the code G. Similarly one can write the parity check constraints in the form:

$$(-1)^{\sum_j H_{lj} x_j} = 1 \;\; \to M^l_{k_1 \cdots k_l} J_{k_1} \cdots J_{k_l} = 1 \tag{2}$$

This defines the "parity constraint matrix " M in terms of the parity check matrix H of the code.

Codewords are sent through a noisy transmission channel and they get corrupted because of the channel noise. If a $J_i = \pm 1$ is sent, the output will be different, in general a real number J_i^{out}. Let us call $Q(\vec{J}^{out}|\vec{J})d\vec{J}^{out}$ the probability for the transmission channel's output to be between $\vec{J}^{out}$ and $\vec{J} + d\vec{J}^{out}$, when the input was $\vec{J}$. The channel "transition matrix" $Q(\vec{J}^{out}|\vec{J})$ is supposed to be known. We will assume that the noise is independent for any pair of bits ("memoryless channel"), i.e.

$$Q(\vec{J}^{out}|\vec{J}) = \prod_i q(J_i^{out}|J_i) \tag{3}$$

Knowing the noise probability i.e. $q(J_i^{out}|J_i)$, the code (i.e. in the present case of linear codes knowing the generating matrix G or the parity check matrix H) and the channel output $\vec{J}^{out}$, one has to infer the message that was sent. The quality of inference depends on the choice of the code.

We will now show that there exists a close mathematical relationship between error-correcting codes and theoretical models of disordered systems. To every possible information message (source word) $\vec{\tau}$ we can assign a probability $P^{source}(\vec{\tau}|\vec{J}^{out})$, conditional on the channel output $\vec{J}^{out}$. Or, equivalently, to any code-word $\vec{J}$ we can assign a probability $P^{code}(\vec{J}|\vec{J}^{out})$.

Because of Bayes theorem, the probability for any code-word symbol ("letter") $J_i = \pm 1$, $p(J_i|J_i^{out})$, conditional on the channel output J_i^{out}, is given by

$$p(J_i|J_i^{out}) \;=\; \frac{q(J_i^{out}|J_i)}{\sum_{J_i} q(J_i^{out}|J_i)}$$

It follows that

$$\ln p(J_i|J_i^{out}) \;=\; c1 \;+\; \ln q(J_i^{out}|J_i) \;=\; c2 \;+\; h_i J_i \tag{4}$$

where $c1$ and $c2$ are constants (non depending on J_i) and

$$h_i \;=\; \frac{1}{2}\ln\frac{q(J_i^{out}|+1)}{q(J_i^{out}|-1)} \tag{5}$$

The two previous equations illustrate the well known fact that the most general function of a variable $J = \pm 1$ is a linear function (because $J^{2k} = 1$,$J^{2k+1} = J$). h_i which will play the role of an external field (see equ. (8) below) or of a coupling constant (see equ. (10)), is called in coding theory the log-likelyhood or the "extrinsic information".

It follows that

$$P^{code}(\vec{J}|\vec{J}^{out}) = c\prod_l \delta(M^l_{k_1\cdots k_l} J_{k_1}\cdots J_{k_l};1)\exp\,(\sum_i h_i J_i) \tag{6}$$

where c is a normalising constant. The Kronecker δ's enforce the constraint that $\vec{J}$ obeys the parity check equations (Equ. (2)), i.e. that it is a code-word. The δ's can be replaced by a soft constraint,

$$P^{code}(\vec{J}|\vec{J}^{out}) = const\,\exp\;[u\;\sum_l M^l_{k_1\cdots k_l} J_{k_1}\cdots J_{k_l}\;+\;\sum_i h_i J_i\;] \tag{7}$$

where $u \to \infty$. We now define the corresponding spin Hamiltonian by:

$$-H^{code}(\vec{J}) = \ln P^{code}(\vec{J}|\vec{J}^{out}) = u\sum_l M^l_{k_1\cdots k_l} J_{k_1}\cdots J_{k_l} + \sum_i h_i J_i \tag{8}$$

There are two models of memoryless channel noise, i.e. of $q(J_i^{out}|J_i)$, that are extensively studied. The first is the "gaussian channel" for which the output J^{out} can take any real value and

$$q(J_i^{out}|J_i) \;=\; c \exp - \frac{(J_i^{out} - J_i)^2}{2w^2}$$

where w^2 is the variance of the gaussian noise and c a normalising constant. The other is the " binary symmetric channel ", for which the output is a binary variable, i.e. $J^{out} = \pm 1$, and

$$q(J_i^{out}|J_i) \;=\; (1-p)\delta_{J_i^{out},J_i} + p\delta_{J_i^{out},-J_i}$$

i.e. every symbol J_i is transmitted without error with probability $1-p$ and is flipped with probability p. For the gaussian channel the field h_i is given by (see equation (5)) $h_i = J_i^{out}/w^2$, while for the binary symmetric channel $h_i = \frac{1}{2} J_i^{out} \, \ln((1-p)/p)$.

Alternatively, one may proceed by solving the parity check constraints

$$J_i = C^i_{k_1 \cdots k_i} \sigma_{k_1} \cdots \sigma_{k_i}$$

by expressing the codewords in terms of the sourcewords.

$$P^{source}(\vec{\sigma}|\vec{J}^{out}) = const. \;\exp\,(\sum_i h_i C^i_{k_1 \cdots k_i} \sigma_{k_1} \cdots \sigma_{k_i}) \qquad (9)$$

where the h_i's are given as before. The logarithm of $P^{source}(\vec{\sigma}|\vec{J}^{out})$,

$$H^{source}(\vec{\sigma}) = -\ln P^{source}(\vec{\sigma}|\vec{J}^{out}) = -\sum_i h_i C^i_{k_1 \cdots k_i} \sigma_{k_1} \cdots \sigma_{k_i} \qquad (10)$$

In equation (10), and in equation (8), the h_i's, are known because the channel output is known (see equation (5)). They are known numbers once the channel output is known.

We imagine the case where we transmit the *same* word a large number of times. Because of the randomness of the noise, every time we will get a different channel output, althowgh the input was the same. We will consider the *ensemble* of all these transmitions and the ensemble of the resulting outputs. This is completely analogous to the case of disordered magnetic systems, where in every sample the positions of the magnetic ions is fixed, but one considers the ensemble of samples obtained with the same experimental procedure (i.e. exactly the same chemical composition, exactly the same concentrations, etc). In statistical mechanics one computes the average value of an observable in this ensemble. There are two reasons for doing this. The first reason is that "good" observables, as for example the magnetization per spin, the energy per spin etc, are "self-averaging". An observable is called self-averaging if its probability distribution over the ensemble of samples becomes a delta function when the size of the sample becomes large. (This property of self-averaging has been recently studied by probabilists and they proved it in several cases. They call it concentration of the measure.) The other reason is that we have developped the tools of computing analytically the ensemble average. Without averaging we are unable, up to now, to perform any analytical computation.

Viewed in this way, the Hamiltonian defined in equation (8) is the Hamiltonian of a spin system with multispin interactions with infinite ferromagnetic coupling and a random external magnetic field, while the Hamiltonian in equation (10), is a spin glass Hamiltonian. I will show later that the error probability per bit is simply related to the magnetization of the corresponding spin model (at the appropriate temperature, see later). It follows that the error probability per bit is self-averaging.

We have given two different statistical mechanics formulations of error correcting codes. One in terms of the souceword probability P^{source} and the other in terms of the codeword probability P^{code}.

Because of the one to one correspondence between codewords and sourcewords, the two formulations are equivalent. In practice however it may make a difference. It may be more convenient to work with P^{source} rather than P^{code}, depending on the case. For the case of turbo codes (see later) it will be more convenient to define another probability, the "register word" probability.

It follows that the most probable symbol sequence ("word maximum a posteriori probability" or " word MAP decoding"), i.e. the symbol sequence that maximises the probability P^{source} or P^{code} (depending on the case), is given by the ground state of this Hamiltonian (H^{code} or H^{source}). Instead of considering the most probable symbol sequence, one may only be interested in the most probable value τ_i^p of the i'th symbole or "bit" τ_i[9, 10, 11], ignoring the values of the other symboles ("symbol MAP decoding"). The sequence of the most probable symboles does not necessarily coincide with the most probable sequence. Because $\tau_i = \pm 1$, the probability p_i for $\tau_i = 1$ is related to the average of τ_i m_i, by $p_i = (1 + m_i)/2$.

$$m_i \;=\; \frac{1}{Z} \sum_{\{\tau_1 \cdots \tau_N\}} \tau_i \; \exp -H(\vec{\tau}) \quad Z = \sum_{\{\tau_1 \cdots \tau_N\}} \exp -H(\vec{\tau}) \quad \tau_i^p = \text{sign } (m_i) \quad (11)$$

In the previous equation m_i is obviously the thermal average at temperature $T = 1$. It is amusing to notice that $T = 1$ corresponds in spin glasses to Nishimori's temperature[14].

When all messages are equally probable and the transmission channel is memoryless and symmetric, i.e. when $q(J_i^{out}|J_i) = q(-J_i^{out}| - J_i)$, the error probability is the same for all input sequences. It is enough to compute it in the case where all input bits are equal to one, i.e. when the transmitted code-word is the all zero's code-word. In this case, the error probability per bit P_e is $P_e \;=\; \frac{1-m^{(d)}}{2}$, where

$m^{(d)} = \frac{1}{N}\sum_{i=1}^{N}\tau_i^{(d)}$ and $\tau_i^{(d)}$ is the symbole sequence produced by the decoding procedure.

This means that it is possible to compute the bit error probability, if one is able to compute the magnetization in the corresponding spin system.

Let me give a simple example of an $R = 1/2$ "convolutional" code. From the N source symbols (letters) u_i's we construct the $2N$ code-word letters x_k^1, x_k^2, $k = 1, \cdots, N$.

$$x_i^1 = u_i + u_{i-1} + u_{i-2}\ ,\quad x_i^2 = u_i + u_{i-2} \tag{12}$$

It follows that

$$J_k^1 = \sigma_k\sigma_{k-1}\sigma_{k-2}\ ,\quad J_k^2 = \sigma_k\sigma_{k-2} \tag{13}$$

$$C^{(1,k)}_{i_{k_1}i_{k_2}i_{k_3}} = \delta_{k,i_{k_1}}\delta_{k,i_{k_2}+1}\delta_{k,i_{k_3}+2}\ ,\quad C^{(2,k)}_{i_{k_1}i_{k_3}} = \delta_{k,i_{k_1}}\delta_{k,i_{k_3}+2} \tag{14}$$

The corresponding spin Hamiltonian is

$$-H = \frac{1}{w^2}\sum_k J_k^{1,out}\tau_k\tau_{k-1}\tau_{k-2} + J_k^{2,out}\tau_k\tau_{k-2} \tag{15}$$

Here I assumed a Gaussian noise. In that case, Equ. (5) reduces to $h_k = J_k^{out}/w^2$, where w^2 is the variance of the noise. This is a one dimensional spin glass Hamiltonian. In fact it is easy to see that convolutional codes correspond to one dimensional spin systems. Their ground state can be found using the $T = 0$ transfer matrix algorithm. The $T = 0$ transfer matrix algorithm corresponds to the Viterbi algorithm in coding theory. For symbol MAP decoding, one can use the $T = 1$ transfer matrix algorithm. The $T = 1$ transfer matrix algorithm is the BCJR algorithm in coding theory[15].

I have illustrated above the mathematical correspondance between disordered spin systems and error correcting codes. Using this correspondance it has been possible to analyse both LPDC codes and turbocodes using the methods of statistical mechanics. Most of the results have been obtained with the "replica" method. (For a lucid exposition of this method see reference (16)). This is a method developped in the context of spin-glass theory and which has not yet been made rigorous. Within the "replica" method there are approximation schemes. The simplest is the replica symmetric approximation. For symbol MAP decoding, i.e. for the temperature $T = 1$, there are very strong arguments that replica symmetry is not broken. It is outside the scope of the present paper to explain the replica method.

To fix the notations, let me remind that Gallager's low density parity check (k, l) codes are defined by choosing at random a sparse parity check $K \times N$ matrix H as follows. H has N columns (we consider the case of codewords of length N). Each column of H has k elements equal to one and all other elements equal to zero. Each row has l non zero elements.

It is convenient to use graphical representations (slightly different from Tanner's graphical representation used in coding theory) to represent the interaction terms appearing in the Hamiltonian. Each spin is represented by a point in the graph. The spins which are multiplied by the same coupling are connected by a line. It follows from equation (8) that Gallager's k, l codes correspond to random "diluted" (sparse) graphs. Such models are called diluted spin models with l-spin infinite strength ferromagnetic interactions in an external random field. It is known that in the case of extreme dilution, one can analyse these models in the mean field approximation. Very sparse graphs have locally a tree structure, i.e. there are no loops of short length. In such a graph with N vertices, the size of the typical loop is known to be $\ln N$. This is the reason why one can apply mean field in this case.

Gallager[5, 6] proposed an approximate iterative decoding algorithm for LDPC codes. This is an iterative computation of the log-likelyhood (or extrinsic information or cavity field, according to the terminology) $h_i(t)$, where t is the iteration time. The probability $p(\sigma_i)$ of the spin σ_i is related to h_i by $p(\sigma_i) = \exp(h_i)/cosh(h_i)$. $h_i(0)$ is given by equ. (5). At $t = 1$ one considers the interaction of σ_i with his neighbors σ_j on the graph. Let us remind that the interaction (see equ. (8) where the limit $u \to \infty$ has to be taken) imposes the product of the spins present in an interaction term to be equal to plus one. Taking into account this information, together with the values of $h_j(0)$, one computes $h_i(1)$. It easy to imagine how this procedure can be iterated. At time t one takes into account the information coming from all the spins which are up to distance t on the graph. It is hoped that this procedure will converge to a fixed point for $p(\sigma_i)$ after a reasonable number of iterations. It is obvious that this number of iterations will depend on the amount of noise. If the noise is too strong there will be no convergence.

This updating of $h_i(t)$, which today is called the sum-product algorithm, would be exact in a graph without loops. It is approximate because of the presence of loops on a random graph. It is worth noticing that decoding with the sum-product algorithm is equivalent to "solving" the corresponding spin model, i.e. computing the local magnetizations, by iteration of the Thouless Anderson Palmer[17] (TAP) equations, which where invented fifteen years later in the context of mean field spin glasses. A more general derivation for spin glasses, called the cavity method, was later developped by Mézard, Parisi and Virasoro[18]. The same algorithm was rediscovered recently in computer science, where it is called the belief propagation algorithm.

As we saw, low density parity check codes are based on a random construction, a random parity check matrix more precisely. We will see that the same is true for Turbo Codes, a random permutation in that case. By the same random construction, for example in Gallager's case matrices with fixed k and l, we can construct several codes, i.e. there exist several random matrices even if k and l are fixed. In order to be able to use statistical mechanics, we have to consider the ensemble of codes defined by these matrices and compute the average error probability per bit in this ensemble. This is justified because it can be shown a posteriori that the error probability per bit is self-averaging.

Low density parity check codes have been analysed using Statistical Mechanics methods by Kabashima Kanter and Saad[19, 20] in the replica symmetric approximation. More recently Montanari[21] was able to go beyond replica symmetry. He established the entire phase diagram of LDPC codes. For $k,\ l \to \infty$ with rate $R = 1 - k/l$ fixed, he showed that k, l codes can be analysed without replicas, similarly to the random energy model of Derrida[22]. There is a phase transition in this model, which occurs at a critical value of the noise n_c. Phase transitions can appear only in the infinite volume limit (the thermodynamic limit), i.e. in the limit of strings of symbols of infinite length. n_c separates a zero error phase, i.e. a phase with a magnetization equel to one, from a high error phase. It turns out that n_c, in this limit, coincides with Shannon's channel capacity. For finite k and l Montanari found an exact one step replica symmetry breaking solution. He computed the location of the phase transition, i.e. the critical value of the noise n_c for which the phase transition occurs. n_c is given in terms of an implicit equation which has to be solved numerically. n_c has a simple asymptotic expansion for large k, l and fixed rate R. For the binary symmetric channel with error probability p, the first term of the asymtotic expansion for the threshold $p_c(k, l)$ is

$$p_c(k,l) = p_c^0 - \frac{1-R}{2(\ln(p_c^0) - \ln(1-p_c^0))}(1-2p_c^0)^{2k} + O((1-2p_c^0)^{4k})$$

p_c^0 is the threshold for $k,\ l \to \infty$, i.e. the threshold provided by the channel capacity. We see that the approach to the $k,\ l \to \infty$ limit is exponential.

The thresholds above were obtained by maximising the appropriate probabilities. This means that they can only be reached by an optimal (but unknown) decoder. The actual decoder may behave differently.

The only alternative to statistical mechanics to theoretically understand LPDC codes and turbocodes is the method of "density evolution" which was devised by Richardson and Urbanke[23]. This method, applied to Gallager's k, l codes, consists in considering the ensemble of Gallager's codes with fixed k and l and the ensemble of channel outputs, when the input is the all zeros codeword. This method of considering an ensemble of codes and an ensemble of channel outputs is very new in coding theory. It can be considered as the rediscovery by coding theorists of the methods developped in the seventies in the study of disordered systems. Richardson and Urbanke study the probability density $\mathcal{P}(h)$ of the log-likelyhoods (or cavity fields) h_i, for this ensemble. As we stated earlier, the sum product decoding algorithm can be viewed as a time evolution process of these h_i's. They study how $\mathcal{P}(h)$ evolves with "time", i.e. when iterating the decoding algorithm. They showed that the probability density converges to the zero error limit provided the noise is less than some value n_c^{bp}. They computed n_c^{bp} performing a local stability analysis of density evolution, starting from the no error regime. n_c^{bp} is not equal to the threshold n_c computed by statistical mechanics for regular Gallager codes, i.e. the decoding is not optimal near the threshold. The reason for this is not yet understood. See the remarks at the end of this paper for a possible explanation.

Turbo Codes also have been analysed using statistical mechanics[24, 25]. They are based on recursive convolutional codes. An example of non recursive convolutional code was given in Equ. (12). The corresponding recursive code is given, most conveniently, in terms of the auxiliary bits b_i, defined below. The b_i's are stored in the encoder's memory registers, that's why I call $\vec{b}$ the the "register word".

$$x_i^1 = u_i, \; x_i^2 = b_i + b_{i-2}, \; b_i = u_i + b_{i-1} + b_{i-2} \tag{16}$$

It follows that the source letters u_i are given in terms of the auxiliary "register letters" b_i

$$u_i = b_i + b_{i-1} + b_{i-2} \tag{17}$$

All additions are modulo two.

To construct a turbo code, one artificially considers a second source word $\vec{v}$, by performing a permutation, chosen at random, of the original code-word $\vec{u}$. So one considers $v_j = u_{P(i)}$ where $j = P(i)$ is a (random) permutation of the K indices i and a second "register word" c_i, $c_i = v_i + c_{i-1} + c_{i-2}$. Obviously

$$v_i = c_i + c_{i-1} + c_{i-2} = u_j = b_j + b_{j-1} + b_{j-2}, \;\; j = P(i) \tag{18}$$

Equ. (18) can be viewed as a constraint on the two register words $\vec{b}$ and $\vec{c}$. Finally in the present example, a rate $R = 1/3$ turbo code, one transmits the $N = 3K$ letter code-word $x_i^1 = u_i$, $x_i^2 = b_i + b_{i-2}$, $x_i^3 = c_i + c_{i-2}$, $i = 1, \cdots, K$. Let's call, as before,

$$J_i^\alpha = (-1)^{x_i^\alpha}, \quad \alpha = 1, 2, 3$$

the channel inputs and $J_i^{out,\alpha}$ the channel outputs. In the previous, for reasons of convenience, we formulated convolutional codes using the source-word probability P^{source} and LDPC codes using the code-word probability P^{code}.

The statistical mechanics of turbo codes is most conveniently formulated in terms of the "register words" probability $P^{reg}(\vec{\sigma}, \vec{\tau} | \vec{J}^{out})$ conditional on the channel outputs $\vec{J}^{out}$, where $\tau_i = (-1)^{b_i}$ and $\sigma_i = (-1)^{c_i}$. The logarithm of this probability provides the spin Hamiltonian

$$-H = \frac{1}{w^2} \sum_k J_k^{out,1} \tau_k \tau_{k-1} \tau_{k-2} + J_k^{out,2} \tau_k \tau_{k-2} + J_k^{out,3} \sigma_k \sigma_{k-2} \tag{19}$$

Because of Equ. (18), the two spin chains $\vec{\tau}$ and $\vec{\sigma}$ obey the constraints

$$\sigma_i \sigma_{i-1} \sigma_{i-2} = \tau_j \tau_{j-1} \tau_{j-2}, \;\; j = P(i) \tag{20}$$

(As previously, we have considered the case of a Gaussian noise of variance w^2.) This is an unusual spin Hamiltonian. Two short range one dimensional chains are coupled through the infinite range, non local constraint, Equ. (20). This constraint is non local because neighboring i's are not mapped to neighboring j's under the random permutation. It turns out that this Hamiltonian can be solved by the replica method.

The equations one gets cannot be solved exactly as in the case of LPDC codes. One can verify that, when the noise is sufficiently weak, zero error probability is a solution of the equations. One can perform a local stability analysis of this zero error probability solution.

One finds a phase transition at a critical value of the noise n_{crit}. For noises less than n_{crit}, it is possible to communicate error free. (For example for the $R = 1/3$ Turbo Code we described above we get that the error probability vanishes for signal to noise ratio above $\Theta \simeq -2.240$ db.) In this respect, turbo codes are similar to Gallager's LDPC codes. The statistical mechanical models however, are completely different.

Figure 1 shows the results of a numerical simulation for the Turbo Code presented above. The bit error probability vanishes above a certain value of the signal to noise ratio, but the value of this threshold seems to be slightly different from the one we obtain from our equations. This disagreement could have different origins. One possibility is that the true threshold is different from that provided by the local stability analysis (possibility of a first order phase transition. Local stability analysis assumes there is a second order transition). Another possibility is that the turbodecoding algorithm does not find the optimal configuration for this code (possibility of a "dynamical transition"). It was explained above that the algorithm tries to find the configuration which maximises the appropriate probability function. It may happen that this function has a large number of local maxima and that the algorithm is trapped in one of them. A third possibility is that when close to the threshold, convergence becomes extremely slow and that one should run the algorithm for an infinite time to reach the correct threshold. (This is called "aging"). It would be interesting to understand which of the above scenaria occurs.

Let me also mention that, under some reasonable assumptions, the iterative decoding algorithm for turbo codes (turbodecoding algorithm), can be viewed[25] as a time discretisation of the Kolmogorov, Petrovsky and Piscounov equation[26]. It is known that this KPP equation has traveling wave solutions. The velocity of the traveling wave, which is computable analytically, corresponds to the convergence rate of the turbodecoding algorithm. The agreement with numerical simulations is excellent, as this is illustrated in Figure 2.

I would like to conclude by pointing out some open questions.

As it was emphasised above, belief propagation decoding is expected to work in the absence of loops. For random graphs the typical loop length $L \sim \log N$, where N is the number of vertices. For $N = 10^6$, $L \sim 10$. However it is known empirically that, in the case of a not very weak noise, one has to iterate t times the decoding algorithm with $t >> L$ ($t \sim 150$ is a typical value), i.e. in practice one cannot ignore the presence of loops. It is not known why the algorithm works in the presence of loops as it does in practice.

We saw that there is a phase transition both in LDPC codes and in Turbo Codes. What is the order of the phase transition? This question is particularly relevant for turbo codes where we assumed a second order transition. Without this assumption we are unable to compute the signal to noise threshold above which communication is error free.

Using statistical mechanics we computed the properties of infinite systems, i.e. infinite message length in the case of error correction codes. In practice of course all messages have finite length. In some applications this length is short. What are the finite size effects? We know from the theory of phase transitions that near a phase transition finite size effects can be very important. Is there finite size scaling? The answer will depend on the order of the phase transition.

It is empirically known that the number of iterations required for the decoding algorithm to converge increases dramatically as the noise increases and one gets close to the phase transition. How does the decoding complexity behave as one approaches the zero error noise threshold? Is there a critical slowing down, as it is usually the case for physical systems near a phase transition? As it was said before, the decoding algorithms both for LDPC codes and turbo codes are heuristic and there are not known results as one approaches the phase transition.

It is well known that disordered systems often exhibit a "glassy" behaviour. This means that below a certain temperature they get trapped in metastable states and do not reach equilibrium in any finite time. Is there a glassy phase in decoding? In other terms, do the heuristic decoding algorithms reach the threshold of optimum decoding (which we computed by equilibrium statistical mechanics) in a finite number of iteration steps, or is there a (lower) noise "dynamical" threshold ("dynamical" transition in the language of disordered systems) where the decoding algorithm gets trapped in metastable states? In that case the decoding algorithm would be unable to reach optimal performance as computed by equilibrium statistical mechanics.

I hope that at least some of the above questions will be answered in the near future.

References

1) Shannon, C. E., A Mathematical Theory of Communication, *Bell Syst. Tech. J.* 27, 379 and 623 (1948)
2) Shannon, C. E. and Weaver W., A Mathematical Theory of Communication, (Univ. of Illinois Press, 1963)
3) C. Berrou, A. Glavieux, and P.Thitimajshima. Proc.1993 Int.Conf.Comm. 1064-1070
4) MacKay, D. J. C. Neal, R. M. *Elect. Lett.* 33, 457 (1997).
5) Gallager, R. G. *IRE Trans. Inform. Theory* , IT-8, 21 (1962).
6) Gallager, R. G. *Low-Density Parity-Check Codes* , MIT Press, Cambridge MA (1963).
7) Sourlas, N., *Nature* 339, 693 (1989)
8) Sourlas, N., in *Statistical Mechanics of Neural Networks*, Lecture Notes in Physics 368, ed. L. Garrido, Springer Verlag (1990)
9) Sourlas, N., Ecole Normale Supérieure preprint (April 1993)
10) Sourlas, N., in *From Statistical Physics to Statistical Inference and Back,* ed. P. Grassberger and J.-P. Nadal, Kluwer Academic (1994) p. 195.
11) Ruján, P., *Phys. Rev. Lett.* 70, 2968 (1993)
12) Nishimori, H., *J. Phys. Soc. Jpn.* , 62, 2973 (1993)
13) Sourlas, N., *Europhys. Lett.* 25, 169 (1994)
14) Nishimori, H., *Progr. Theor. Phys.* 66, 1169 (1981)
15) L. Bahl, J. Cocke, F. Jelinek, and J. Raviv. IEEE Trans.Inf.Theory **IT-20**(1974) 248-287
16) Mézard, M. Parisi, G. Virasoro, M. A. Spin Glass Theory and Beyond, World Scientific, 1987
17) Thouless, D. J. Anderson, P. W. Palmer, R. G. *Phil. Mag.* 35, 593 (1977)
18) Mézard, M. Parisi, G. Viresoro, M. A. *Europhys. Lett.* 1, 77 (1986)
19) Kanter, I. and Saad, D. *Phys. Rev. Lett.* 83, 2660 (1999)
20) Kabashima, Y. Murayama T. and Saad, D. *Phys. Rev. Lett.* 84, 1355 (2000)
21) Montanari, A. *Eur. Phys. J.* B 23, 121 (2001)
22) Derrida, B., Random-energy model: An exactly solvable model of disordered systems, *Phys. Rev.* B24, 2613-2626 (1981)
23) Richardson, T. J. Urbanke, R. L. *IEEE Trans. Inform. Theory* 47, 638 (2001).
24) Montanari, A. Sourlas, N. *Eur. Phys. J.* B 18, 107 (2000)
25) Montanari, A. *Eur. Phys. J.* B 18, 121 (2000)
26) Kolmogorov, A. Petrovsky, I and Piscounov, N. *Moscou Univ. Math. Bull.* 1, 1 (1937).

Figure Captions

Figure 1:

Numerical results for the average error probability per bit of the Turbo Code described in the text (gaussian channel). Stars ($*$) are obtained for a random permutation, diamonds ($\diamond$) correspond to the identity permutation. The continuous curve corresponds to the uncoded message. The leftmost vertical line is located at the Shannon capacity, while the rightmost one is the threshold computed using statistical mechanics (see text).

Figure 2:

The dynamics of the turbo decoding algorithm in the low-noise regime. Triangles, diamonds and circles represent the average extrinsic information as a function of the number of iterations for different sizes ($L = 5000, 50000, 500000$) of the source message. Notice that the saturation after a large number of iterations is a finite size effect. The slope of the straight line describes the asymptotic behavior for an infinitely long message and is obtained analytically from the KPP equation.