arXiv:cond-mat/0512296 v1 14 Dec 2005

Two Lectures on Iterative Coding and Statistical Mechanics

Andrea Montanari

*Laboratoire de Physique Théorique de l'Ecole Normale Supérieure**,

24, rue Lhomond, 75231 Paris CEDEX 05, France

(Dated: May 23, 2006)

These are the notes for two lectures delivered at the Les Houches summer school *Mathematical Statistical Mechanics*, held in July 2005.

I review some basic notions on sparse graph error correcting codes with emphasis on 'modern' aspects, such as, iterative belief propagation decoding. Relations with statistical mechanics, inference and random combinatorial optimization are stressed, as well as some general mathematical ideas and open problems.

I. INTRODUCTION

Imagine to enter the auditorium and read the following (partially erased) phrase on the blackboard

TH* L*CTU*E OF *********** WA* EX**EMELY B*RING.

You will be probably able to reconstruct most of the *words* in the phrase despite the erasures. The reason is that English language is redundant. One can roughly quantify this redundancy as follows. The English dictionary contains about 10^6 words including technical and scientific terms. On the other hand, the average length of these words is about 8.8 letters [1]. A conservative estimate of the number of 'potential' English words is therefore $26^8 \approx 2 \cdot 10^{11}$. A tiny fraction (about 10^{-5}) of these possibilities is realized [1]. This is of course a waste from the point of view of information storage, but it allows for words to be robust against errors (such as the above erasures). Of course, they are not infinitely robust: above some threshold the information is completely blurred out by noise (as in the case of the name of the speaker in our example).

A very naïve model for the redundancy of English could be the following. In order for a word to be easily pronounced, it must contain some alternation of vowels and consonants. Let us be rough and establish that an English word is a sequence of 8 letters, not containing two consecutive vowels

---

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

[1] This estimate was obtained by the author averaging over 40 words randomly generated by the site `http://www.wordbrowser.net/wb/wb.html`.

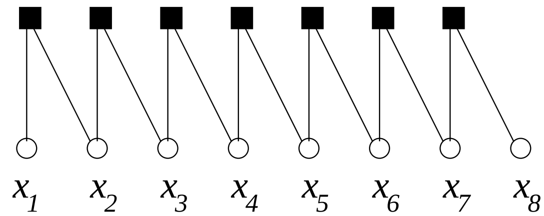

FIG. 1: A (too) naïve model for the English language.

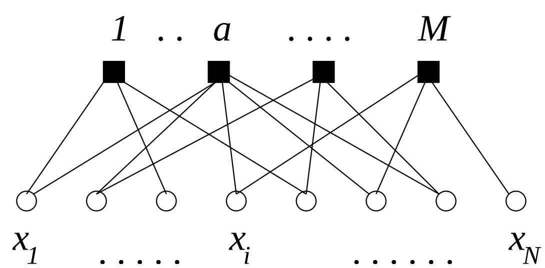

FIG. 2: Example of a Tanner graph.

and consonants. This yields $2 \cdot 21^4 5^4 \approx 2.4 \cdot 10^8$ distinct words, which overestimates the correct number 'only' by a factor 240. A graphical representation of this model is reproduced in Fig. 1.

The aim of coding theory is to construct an optimal 'artificial dictionary' allowing for reliable communication through unreliable media. It is worth introducing some jargon of this discipline. Words of natural languages correspond to **codewords** in coding. Their length (which is often considered as fixed) is called the **blocklength**: we shall denote it by N throughout these lectures. The dictionary (i.e. the set of all words used for communication) is called **codebook** and denoted as $\mathcal{C}$. As in our example, the dictionary size is usually exponential in the blocklength $|\mathcal{C}| = 2^{NR}$ and R is called the code **rate**. Finally, the communication medium referred to as the **channel** and is usually modeled in a probabilistic way (we shall see below a couple of examples).

II. CODES ON GRAPHS

We shall now construct a family of such 'artificial dictionaries' (codes). For the sake of simplicity, codewords will be formed over the binary alphabet $\{\mathtt{0},\mathtt{1}\}$. Therefore a codeword $\underline{x} \in \mathcal{C}$ will be an element of the Hamming space $\{\mathtt{0},\mathtt{1}\}^N$ or, equivalently a vector of coordinates $(x_1, x_2, \dots, x_N) \equiv \underline{x}$.

The codebook $\mathcal{C}$ is a subset of $\{\mathtt{0},\mathtt{1}\}^N$. Inspired by our simple model of English, we shall define $\mathcal{C}$ by stipulating that $\underline{x}$ is a codeword if and only if a certain number M of constraints on the bits

$x_1, \dots, x_N$ are met. In order to specify these constraints, we will draw a bipartite graph (**Tanner graph**) over vertices sets $[N]$ and $[M]$. Vertices in these sets will be called, respectively, variable nodes (denoted as $i, j, \dots$) and check nodes ($a, b, \dots$). If we denote by $i_1^a, i_2^a, \dots, i_k^a$ the variable nodes adjacent to check node a in the graph, then $x_{i_1^a}, x_{i_2^a}, \dots, x_{i_k^a}$ must satisfy some constraint in order for $\underline{x}$ to be a codeword. An example of a Tanner graph is depicted in Fig. 2.

Which type of constraints are we going to enforce on the symbols $x_{i_1^a}, x_{i_2^a}, \dots, x_{i_k^a}$ adjacent to the same check? The simplest and most widespread choice is a simple parity check condition: $x_{i_1^a} \oplus x_{i_2^a} \oplus \cdots \oplus x_{i_k^a} = \mathtt{0}$ (where $\oplus$ is my notation for sum modulo 2). We will stick to this choice, although several of the ideas presented below are easily generalized. Notice that, since the parity check constraint is linear in $\underline{x}$, the code $\mathcal{C}$ is a linear subspace of $\{\mathtt{0}, \mathtt{1}\}^N$, of size $|\mathcal{C}| \geq 2^{N-M}$ (and in fact $|\mathcal{C}| = 2^{N-M}$ unless redundant constraints are used in the code definition). For general information theoretic reasons one is particularly interested in the limit of large blocklength $N \to \infty$ at fixed rate. This implies that the number of checks per variable is kept fixed: $M/N = 1 - R$.

Once the general code structure is specified, it is useful to define a set of parameters which characterize the code. Eventually, these parameters can be optimized over to obtain better error correction performances. A simple set of such parameters is the **degree profile** (Λ, P) of the code. Here $\Lambda = (\Lambda_0, \dots, \Lambda_{l_{\max}})$, where Λ_l is the fraction of variable nodes of degree l in the Tanner graph. Analogously, $P = (P_0, \dots, P_{k_{\max}})$, where P_k is the fraction of check nodes of degree k.

Given the degree profile, there is of course a large number of graphs having the same profile. How should one chose among them? In his seminal 1948 paper, Shannon first introduced the idea of randomly constructed codes. We shall follow his intuition here and assume that the Tanner graph defining $\mathcal{C}$ is generated uniformly at random among the ones with degree profile (Λ, P) and blocklength N. The corresponding code (graph) ensemble is denoted as $\mathrm{LDPC}_N(\Lambda, P)$ (respectively $\mathsf{G}_N(\Lambda, P)$), an acronym for **low-density parity-check** codes. Generically, one can prove that some measure of the code performances concentrates in probability with respect to the choice of the code in the ensemble. Therefore, a random code is likely to be (almost) as good as (almost) any other one in the ensemble.

An particular property of the random graph ensemble will be useful in the following. Let $G \stackrel{\mathrm{d}}{=} \mathsf{G}_N(\Lambda, P)$ and i a uniformly random variable node in G. Then, with high probability (i.e. with probability approaching one in the large blocklength limit), the shortest loop in the G through i is of length $\Theta(\log N)$.

III. A SIMPLE–MINDED BOUND AND BELIEF PROPAGATION

A. Characterizing the code performances

Once the code is constructed, we have to convince ourselves (or somebody else) that it is going to perform well. The first step is therefore to produce a model for the communication process, i.e. a noisy channel. A simple such model (usually referred to as **binary symmetric channel**, or BSC(p)) consists in saying that each bit x_i is flipped independently of the others with probability p. In other words the channel output y_i is equal to x_i with probability $1-p$ and different with probability p. This description can be encoded in a transition probability kernel $Q(y|x)$. For BSC(p) we have $Q(\mathtt{0}|\mathtt{0}) = Q(\mathtt{1}|\mathtt{1}) = 1-p$ and $Q(\mathtt{1}|\mathtt{0}) = Q(\mathtt{0}|\mathtt{1}) = p$. More generally, we shall consider transition probabilities satisfying the 'symmetry condition' $Q(y|\mathtt{0}) = Q(-y|\mathtt{1})$ (in the BSC case, this condition fulfilled if we use the $+1$, -1 notation for the channel output).

The next step consists in establishing a measure of the performances of our code. There are several natural such measures, for instance the expected number of incorrect symbols, or of incorrect words. To simplify the arguments below, it is convenient to consider a slightly less natural measure, which conveys essentially the same information. Recall that, given a discrete random variable X, with distribution $\{p(x) : x \in \mathcal{X}\}$, its entropy, defined as

$$H(X) = -\sum_{x\in\mathcal{X}} p(x) \log p(x) \tag{1}$$

is a measure of how 'uncertain' is X. Analogously, if X, Y are two random variables with joint distribution $\{p(x,y) : x \in \mathcal{X},\, y \in \mathcal{Y}\}$, the conditional entropy

$$H(X|Y) = -\sum_{x\in\mathcal{X},\, y\in\mathcal{Y}} p(x,y) \log p(x|y) \tag{2}$$

is a measure of how 'uncertain' is X once Y is given.

Now consider a uniformly random codeword $\underline{X}$ and the corresponding channel output $\underline{Y}$ (as produced by the binary symmetric channel). The conditional entropy $H(\underline{X}|\underline{Y})$ measures how many additional bits of information (beyond the channel output) do we need for reconstructing $\underline{x}$ from $\underline{y}$. This is a fundamental quantity but sometimes difficult to evaluate because of its non-local nature. We shall therefore also consider the bitwise conditional entropy

$$h_{\rm b} = \frac{1}{N}\sum_{i=1}^{N} H(X_i|\underline{Y})\,. \tag{3}$$

B. Bounding the conditional entropy

Before trying to estimate these quantities, it is convenient to use the channel and code symmetry in order to simplify the task. Consider for instance the conditional entropy. Denoting by $\underline{0}$ the 'all zero' codeword, we have

$$\begin{aligned} H(\underline{X}|\underline{Y}) &= -\sum_{\underline{x},\underline{y}} p(\underline{x})p(\underline{y}|\underline{x}) \log p(\underline{x}|\underline{y}) = & (4)\\ &= -\sum_{\underline{y}} p(\underline{y}|\underline{0}) \log p(\underline{0}|\underline{y}) = & (5)\\ &= -\mathbb{E}_y \log p(\underline{0}|\underline{y})\,, & (6)\end{aligned}$$

where $\mathbb{E}_y$ denotes expectation with respect to the probability measure $p(\underline{y}|\underline{0}) = \prod_i Q(y_i|0)$. In the BSC($p$) case, under this measure, the y_i are i.i.d. Bernoulli random variables with parameter p. Furthermore, by using Bayes theorem, we get

$$\begin{aligned} H(\underline{X}|\underline{Y}) &= -\mathbb{E}_y \log p(\underline{y}|\underline{0}) + \mathbb{E}_y \log \left\{ \sum_{\underline{x}\in\mathcal{C}} p(\underline{y}|\underline{x}) \right\} = & (7)\\ &= -N\sum_y Q(y|0)\log Q(y|0) + \mathbb{E}_y \log \left\{ \sum_{\underline{x}} \prod_i Q(y_i|x_i) \prod_a \mathbb{I}(x_{i_1^a} \oplus \cdots \oplus x_{i_k^a} = 0) \right\}. & (8)\end{aligned}$$

The first term is easy to evaluate consisting of a finite sum (or, at most, a finite-dimensional integral). The second one can be identified as the quenched free energy for a disordered model with binary variables (Ising spins) associated to vertices of the the Tanner graph G. Proceeding as in (6) one also gets the following expression for the single bit conditional entropy

$$H(X_i|\underline{Y}) = -\mathbb{E}_y \log p(x_i = 0|\underline{y})\,. \tag{9}$$

A simple idea for bounding a conditional entropy is to use the 'data processing inequality'. This says that, if $X \to Y \to Z$ is a Markov chain, then $H(X|Y) \le H(X|Z)$. Let $\mathsf{B}(i,r)$ denote the subgraph of G whose variable nodes lie at a distance at most r from i (with the convention that a check node a belongs to $\mathsf{B}(i,r)$ only if all of the adjacent variable nodes belong to G). Denote by $\underline{Y}_{i,r}$ the vector of output symbols Y_j, such that $j \in \mathsf{B}(i,r)$. The data processing inequality implies

$$H(X_i|\underline{Y}) \le H(X_i|\underline{Y}_{i,r}) = -\mathbb{E}_y \log p(x_i = 0|\underline{y}_{i,r})\,. \tag{10}$$

A little more work shows that this inequality still holds if $p(x_i|\underline{y}_{i,r})$ is computed as if there weren't parity checks outside $\mathsf{B}(i,r)$. In formulae, we can substitute $p(x_i = 0|\underline{y}_{i,r})$ with

$$p_{i,r}(x_i = x|\underline{y}_{i,r}) \equiv \sum_{\underline{x}_{i,r}=x} p_{i,r}(\underline{x}_{i,r}|\underline{y}_{i,r})\,. \tag{11}$$

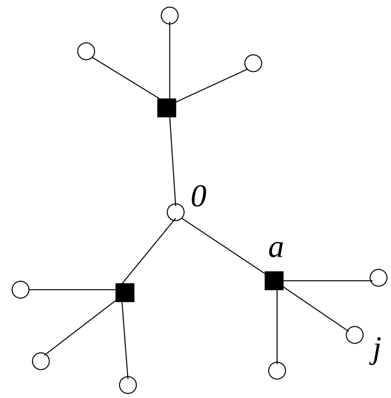

FIG. 3: Radius 1 neighborhood of a typical site in the i in the Tanner graph.

where

$$p_{i,r}(\underline{x}_{i,r}|\underline{y}_{i,r}) \equiv \frac{1}{Z_{i,r}(\underline{y}_{i,r})} \prod_{j\in \mathsf{B}(i,r)} Q(y_j|x_j) \prod_{a\in \mathsf{B}(i,r)} \mathbb{I}(x_{i_1^a} \oplus \cdots \oplus x_{i_k^a} = 0)\,. \tag{12}$$

and $Z_{i,r}(\underline{y}_{i,r})$ ensures the correct normalization of $p_{i,r}(\underline{x}_{i,r}|\underline{y}_{i,r})$.

We are left with the task of computing $p_{i,r}(x_i = \mathsf{0}|\underline{y}_{i,r})$. As a warmup exercise, let us consider the case $r = 1$. Without loss of generality, we set $i = 0$. Because of the remark made in the previous Section, the subgraph $\mathsf{B}(0,1)$ is, with high probability, a tree and must look like the graph in Fig. 3. Using the notations introduced in this figure, and neglecting normalization constants (which can be computed at the very end), we have

$$p_{0,1}(x_0|\underline{y}_{0,1}) \propto \sum_{\{x_j\}} Q(y_0|x_0) \prod_{a\in\partial 0} \prod_{j\in\partial a\backslash 0} Q(y_j|x_j) \prod_{a\in\partial 0} \mathbb{I}(x_0 \oplus x_{\partial a\backslash 0} = 0)\,. \tag{13}$$

Here we used the notation ∂i (∂a) to denote the set of check nodes (respectively variable nodes) adjacent to variable node i (resp. to check node a). Moreover, for $A = \{i_1, \dots, i_k\}$, we wrote $x_A = x_{i_1} \oplus \cdots \oplus x_{i_k}$. Rearranging the various summations, we get the expression

$$p_{0,1}(x_0|\underline{y}_{0,1}) \propto Q(y_0|x_0) \prod_{a\in\partial 0} \sum_{x_j,\, j\in\partial a\backslash 0} \mathbb{I}(x_0 \oplus x_{\partial a\backslash 0} = 0) \prod_{j\in\partial a\backslash 0} Q(y_j|x_j)\;, \tag{14}$$

which is much simpler to evaluate due to its recursive structure. In order to stress this point, we can write the above formula as

$$p_{0,1}(x_0|\underline{y}_{0,1}) \;\propto\; Q(y_0|x_0) \prod_{a\in\partial 0} \hat{\mu}_{a\to 0}(x_0)\,, \tag{15}$$

$$\hat{\mu}_{a\to 0}(x_0) \;\propto\; \sum_{x_j,\, j\in\partial a\backslash 0} \mathbb{I}(x_0 \oplus x_{\partial a\backslash 0} = 0) \prod_{j\in\partial a\backslash 0} \mu_{j\to a}(x_j)\,, \tag{16}$$

$$\mu_{j\to a}(x_j) \;\propto\; Q(y_i|x_i)\,. \tag{17}$$

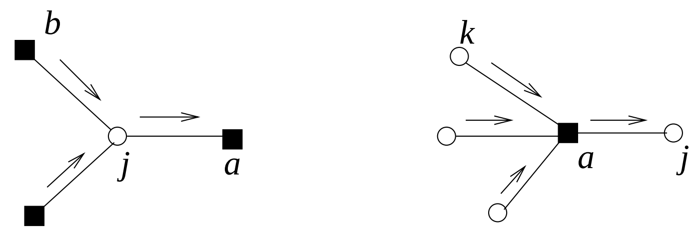

FIG. 4: Graphical representation of the belief propagation equations.

The quantities $\{\mu_{j\to a}(x_j)\}$, $\{\hat{\mu}_{j\to a}(x_j)\}$ are normalized distributions associated with the directed edges of G. They are referred to as **beliefs** or, more generally, **messages**. Notice that, in the computation of $p_{0,1}(x_0|\underline{y}_{0,1})$ only messages along edges in $\mathsf{B}(0,1)$, directed toward the site 0, were relevant.

In the last form, the computation of $p_{i,r}(x_i|\underline{y}_{i,r})$ is easily generalized to any finite r. We first notice that that $\mathsf{B}(i,r)$ is a tree with high probability. Therefore, we can condition on this event without much harm. Then we associate messages to the directed edges of $\mathsf{B}(i,r)$ (only messages directed towards i are necessary). Messages are computed according to the rules

$$\mu_{j\to a}(x_j) \;\propto\; Q(y_j|x_j) \prod_{b\in\partial j\setminus a} \hat{\mu}_{b\to j}(x_j)\,, \tag{18}$$

$$\hat{\mu}_{a\to j}(x_j) \;\propto\; \sum_{x_k,\, k\in\partial a\setminus j} \mathbb{I}(x_j\oplus x_{\partial a\setminus j}=0) \prod_{k\in\partial a\setminus j} \mu_{k\to a}(x_j)\,, \tag{19}$$

with boundary condition $\hat{\mu}_{b\to j}(x_j) = 1/2$ for all b's outside $\mathsf{B}(i,r)$. These equations are represented graphically in Fig. 4. Finally, the desired marginal distribution is obtained as

$$p_{i,r}(x_i|\underline{y}_{i,r}) \;\propto\; Q(y_i|x_i) \prod_{a\in\partial i} \hat{\mu}_{a\to i}(x_i)\,. \tag{20}$$

Let us now forget for a moment our objective of proving an upper bound on the conditional entropy $H(X_i|\underline{Y})$. The intuitive picture is that, as r increases, the marginal $p_{i,r}(x_i|\underline{y}_{i,r})$ incorporates information coming form a larger number of received symbols and becomes a more accurate approximation of $p(x_i|\underline{y})$. Ideally, optimal decoding of the received message would require the computation of $p(x_i|\underline{y})$, for which no efficient algorithm is known. In particular, the expected number of incorrect bits is minimized by the rule $\hat{x}(\underline{y}) = \mathrm{argmax}_{x_i}\, p(x_i|\underline{y})$. We can however hope that nearly optimal performances can be obtained through the rule

$$\hat{x}_{i,r}(\underline{y}) = \mathrm{argmax}_{x_i}\, p_{i,r}(x_i|\underline{y}_{i,r})\,. \tag{21}$$

Furthermore, a moment of thought shows that the recursive procedure described above can be implemented in parallel for all the variables $i\in[N]$. We just need to initialize $\hat{\mu}_{b\to j}(x_j) = 1/2$ for all the check-to-variable messages, and then iterate the update equations (18), (19) at all nodes in G, exactly r times. For any fixed r, this requires just $\Theta(N)$ operations which is probably the smallest computational effort one can hope for.

Finally, although Eqs. (18), (19) only allow to compute $p_{i,r}(x_i|\underline{y}_{i,r})$ as far as $\mathsf{B}(i,r)$ is a tree, the algorithm is well defined for any value of r. One can hope to improve the performances by taking larger values of r.

C. A parenthesis

The algorithm we 'discovered' in the previous Section is in fact well known under the name of **belief propagation** (BP) and is widely adopted for decoding codes on graphs. This is in turn an example of a wider class of algorithms which are particularly adapted to problems defined on sparse graphs, and are called (in a self-explanatory way) **message passing algorithms**. We refer to Sec. VI for some history and bibliography.

Physicists will quickly recognize that Eqs. (18), (19) are just the equations for Bethe-Peierls approximation in the model at hand [2, 3]. Unlike the original Bethe equations, because of the quenched disorder, the solutions of these equations depend on the particular sample, and are not 'translation invariant'. These two features make Eqs. (18), (19) analogous to Thouless, Anderson, Palmer (TAP) equations for mean field spin glasses. In fact Eqs. (18), (19) are indeed the correct generalization of TAP equations for diluted models [2]. Of course many of the classical issues in the context of the TAP (such as the existence of multiple solutions, treated in the lectures of Parisi at this School) approach have a direct algorithmic interpretation here.

Belief propagation was introduced in the previous paragraph as an algorithm for approximately computing the marginals of the probability distribution

$$p(\underline{x}|\underline{y}) \equiv \frac{1}{Z(\underline{y})} \prod_{j\in[N]} Q(y_j|x_j) \prod_{a\in[M]} \mathbb{I}(x_{i_1^a} \oplus \cdots \oplus x_{i_k^a} = 0)\,, \tag{22}$$

which is 'naturally' associated to the graph G. It is however clear that the functions $Q(y_j|x_j)$ and $\mathbb{I}(x_{i_1^a} \oplus \cdots \oplus x_{i_k^a} = 0)$ do not have anything special. They could be replaced by any set of **compatibility functions** $\psi_i(x_i)$, $\psi_a(x_{i_1^a}, \dots, x_{i_k^a})$. Equations (18), (19) are immediately generalized. BP can therefore be regarded as a general inference algorithm for probabilistic graphical models.

[2] In the theory of mean field disordered spin models, one speaks of diluted models whenever the number of interaction terms (M in the present case) scales as the size of the system.

The success of message passing decoding has stimulated several new applications of this strategy: let us mention a few of them. Mézard, Parisi and Zecchina [4] introduced 'survey propagation', an algorithm which is meant to generalize BP to the case in which the underlying probability distribution decomposes in an exponential number of pure states (replica symmetry breaking, RSB). In agreement with the prediction of RSB for many families of random combinatorial optimization problems, survey propagation proved extremely effective in this context.

One interesting feature of BP is its decentralized nature. One can imagine that computation is performed locally at variable and check nodes. This is particularly interesting for applications in which a large number of elements with moderate computational power must perform some computation collectively (as is the case in sensor networks). Van Roy and Moallemi [5] proposed a 'consensus propagation' algorithm for accomplishing some of these tasks.

It is sometimes the case that inference must be carried out in a situation where a well established probabilistic model is not available. One possibility in this case is to perform 'parametric inference' (roughly speaking, some parameters of the model are left free). Sturmfels proposed a 'polytope propagation' algorithm for these cases [6, 16].

IV. DENSITY EVOLUTION A.K.A. DISTRIBUTIONAL RECURSIVE EQUATIONS

Evaluating the upper bound (10) on the conditional entropy described in the previous Section, is essentially the same as analyzing the BP decoding algorithm defined by Eqs. (18) and (19). In order to accomplish this task, it is convenient to notice that distributions over a binary variable can be parametrized by a single real number. It is customary to choose this parameters to be the **log-likelihood ratios** (the present definition differs by a factor 1/2 from the traditional one):

$$v_{i\to a} \equiv \frac{1}{2}\log\frac{\mu_{i\to a}(0)}{\mu_{i\to a}(1)}\,, \qquad \hat{v}_{a\to i} \equiv \frac{1}{2}\log\frac{\hat{\mu}_{a\to i}(0)}{\mu_{a\to i}(1)}\,. \tag{23}$$

We further define $h_i \equiv \frac{1}{2}\log\frac{Q(y_i|0)}{Q(y_i|1)}$. In terms of these quantities, Eqs. (18) and (19) read

$$v^{(r+1)}_{j\to a} = h_j + \sum_{b\in\partial j\setminus a}\hat{v}^{(r)}_{b\to j}\,, \qquad \hat{v}^{(r)}_{a\to j} = \operatorname{atanh}\left[\prod_{k\in\partial a\setminus j}\tanh v^{(r)}_{k\to a}\right]\,. \tag{24}$$

Notice that we added an index $r\in\{0,1,2,\dots\}$ that can be interpreted in two equivalent ways. On the one hand, the message $v^{r}_{j\to a}$ conveys information on the bit x_j coming from a ('directed') neighborhood of radius r. On the other r indicates the number of iterations in the BP algorithm. As for the messages, we can encode the conditional distribution $p_{i,r}(x_i|\underline{y}_{i,r})$ through the single number $u_i^{(r)} \equiv \frac{1}{2}\log\frac{p_{i,r}(0|\underline{y}_{i,r})}{p_{i,r}(1|\underline{y}_{i,r})}$. Equation (20) then reads

$$u_i^{(r+1)} = h_i + \sum_{b\in\partial i} \hat{v}_{b\to i}^{(r)}\,. \tag{25}$$

Assume now that the graph G is distributed accordingly to the $\mathsf{G}_N(\Lambda, P)$ ensemble and that $i \to a$ is a uniformly random (directed) edge in this graph. It is intuitively clear (and can be proved as well) that, for any giver r $v_{i\to a}^{(r)}$ will converge in distribution to some well defined random variable $v^{(r)}$. This can be defined recursively by setting $v^{(0)} \stackrel{\rm d}{=} h$ and, for any $r \ge 0$

$$v^{(r+1)} \stackrel{\rm d}{=} h + \sum_{b=1}^{l-1} \hat{v}_b^{(r)}\,, \qquad \hat{v}^{(r)} \stackrel{\rm d}{=} \text{atanh}\left[\prod_{j=1}^{k-1} \tanh v_j^{(r)}\right]\,, \tag{26}$$

where $\{\hat{v}_b^{(r)}\}$ are i.i.d. random variables distributed as $\hat{v}^{(r)}$, and $\{v_j^{(r)}\}$ are i.i.d. random variables distributed as $v^{(r)}$. Furthermore l and k are integer random variable with distributions, respectively λ_l, ρ_k depending on the code ensemble:

$$\lambda_l = \frac{l\Lambda_l}{\sum_{l'} l'\Lambda_{l'}}\,, \qquad \rho_k = \frac{kP_k}{\sum_{k'} k'P_{k'}}\,. \tag{27}$$

In other words $v^{(r)}$ is the message at the root of a random tree whose offspring distributions are given by λ_l, ρ_k. This is exactly the asymptotic distribution of the tree inside the ball $\mathsf{B}(i,r)$. The recursions (26) are known in coding theory as **density evolution** equations. They are indeed the same (in the present context) as Aldous' recursive distributional equations [7], or replica symmetric equations in spin glass theory. From Eq. (25) one easily deduces that $u_i^{(r)}$ converges in distribution to $u^{(r)}$ defined by $u^{(r+1)} \stackrel{\rm d}{=} h + \sum_{b=1}^{l} \hat{v}_b^{(r)}$ with l distributed according to Λ_l.

At this point we can use Eq. (10) to derive a bound on the bitwise entropy $h_{\rm b}$. Denote by $h(u)$ the entropy of a binary variable whose log-likelihood ratio is u. Explicitly

$$h(u) = -(1+e^{-2u})^{-1}\log(1+e^{-2u})^{-1} - (1+e^{2u})^{-1}\log(1+e^{2u})^{-1}\,. \tag{28}$$

Then we have, for any r,

$$\lim_{N\to\infty} \mathbb{E}_{\mathcal{C}}\, h_{\rm b} \le \mathbb{E}\, h(u^{(r)})\,, \tag{29}$$

where we emphasized that the expectation on the left hand side has to be taken with respect to the code.

It is easy to show that the right hand side of the above inequality is non-increasing with r. It is therefore important to study its asymptotic behavior. As $r \to \infty$, the random variables $v^{(r)}$ converge to a limit $v^{(\infty)}$ which depends on the code ensemble as well as on the channel transition

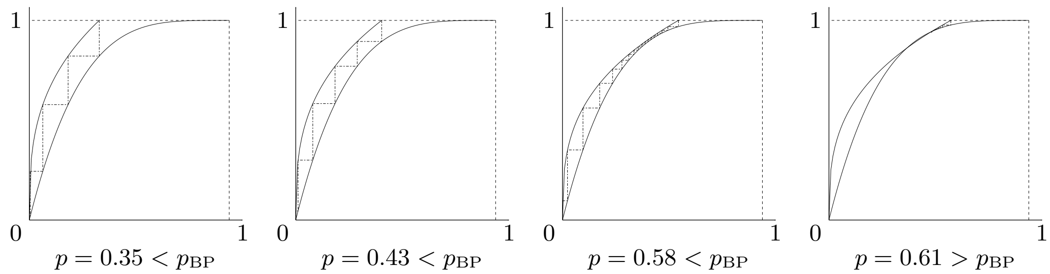

FIG. 5: Graphical representation of density evolution for the binary erasure channel, cf. Eq. (30).

probabilities $Q(y|x)$. Usually one is interested in a continuous family of channels indexed by a noise parameter p as for the BSC(p), indexed in such a way that the channel 'worsen' as p increases (this notion can be made precise and is called **physical degradation**). For 'good' code ensembles, the following scenario holds. For small enough p, $v^{(\infty)} = +\infty$ with probability one. Above some critical value $p_{\rm BP}$, $v^{(\infty)} \leq 0$ with non-zero probability. It can be shown that no intermediate case is possible. In the first case BP is able to recover the transmitted codeword (apart, eventually, from a vanishingly small fraction of bits), and the bound (29) yields $\mathbb{E}_{\mathcal{C}}\, h_{\rm b} \to 0$. In the second the upper bound remains strictly positive in the $r \to \infty$ limit.

A particularly simple channel model, allowing to work out in detail the behavior of density evolution, is the binary erasure channel BEC(p). In this case the channel output can take the values $\{\mathtt{0}, \mathtt{1}, ?\}$. The transition probabilities are $Q(\mathtt{0}|\mathtt{0}) = Q(\mathtt{1}|\mathtt{1}) = 1-p$ and $Q(?|\mathtt{0}) = Q(?|\mathtt{1}) = p$. In other words, each input is erased independently with probability p, and transmitted incorrupted otherwise. We shall further assume, for the sake of simplicity, that the random variables l and k in Eq. (26) are indeed deterministic. In the other words all variable nodes (parity check nodes) in the Tanner graph have degree l (degree k). It is not hard to realize, under the assumption that the all zero-codeword has been transmitted, that in this case $v^{(r)}$ takes values 0 or $+\infty$. If we denote by z_r the probability that $v^{(r)} = 0$, the density evolution equations (26) become simply

$$z_{r+1} = p\left(1 - (1 - z_r)^{k-1}\right)^{l-1}\,. \tag{30}$$

The functions $f_p(z) \equiv (z/p)^{1/(l-1)}$ and $g(z) \equiv 1 - (1-z)^{k-1}$ are plotted in Fig. 5 for $l = 4$, $k = 5$ and a few values of p approaching $p_{\rm BP}$. The recursion (30) can be described as 'bouncing back and forth' between the curves $f_p(z)$ and $g(z)$. A little calculus shows that $z_r \to 0$ if $p < p_{\rm BP}$ while $z_r \to z_*(p) > 0$ for $p \geq p_{\rm BP}$, where $p_{\rm BP} \approx 0.6001110$ in the case $l = 4$, $k = 5$. A simple exercise for the reader is to work out the upper bound on the r.h.s. of Eq. (29) for this case and studying it as a function of p.

V. THE AREA THEOREM AND SOME GENERAL QUESTIONS

Let us finally notice that general information theory considerations imply that $H(\underline{X}|\underline{Y}) \leq \sum_i H(X_i|\underline{Y})$. As a consequence the total entropy per bit $H(\underline{X}|\underline{Y})/N$ vanishes as well for $p < p_{\rm BP}$. However this inequality greatly overestimates $H(\underline{X}|\underline{Y})$: bits entering in the same parity check are, for instance, highly correlated. How can a better estimate be obtained?

The bound in Eq. (29) can be expressed by saying that the actual entropy is strictly smaller than the one 'seen' by BP. Does it become strictly positive for $p > p_{\rm BP}$ because of the sup-optimality of belief propagation or because $H(\underline{X}|\underline{Y})/N$ is genuinely positive?

More in general, below $p_{\rm BP}$ BP is essentially optimal. What happens above? A way to state more precisely this question consists in defining the distortion

$$D_{{\rm BP},r} \equiv \frac{1}{N}\sum_{i=1}^{N}\sum_{x_i\in\{0,1\}} \left|p(x_i|\underline{y}) - p_{i,r}(x_i|\underline{y})\right| \, , \tag{31}$$

which measures the distance between the BP marginals and the actual ones. Below $p_{\rm BP}$, $D_{{\rm BP},r} \to 0$ as $r \to \infty$. What happens above?

It turns out that all of these questions are strictly related. We shall briefly sketch an answer to the first one and refer to the literature for the others. However, it is worth discussing why they are challenging, considering in particular the last one (which somehow implies the others). Both $p(x_i|\underline{y})$ and $p_{i,r}(x_i|\underline{y})$ can be regarded as marginals of some distribution on the variables associated to the tree $\mathsf{B}(i,r)$. While, in the second case, this distribution has the form (12), in the first one some complicated (and correlated) boundary condition must be added in order to keep into account the effect of the code outside $\mathsf{B}(i,r)$. Life would be easy if the distribution of x_i were asymptotically decorrelated from the boundary condition as $r \to \infty$, for *any boundary condition.* In mathematical physics terms, the infinite tree (obtained by taking $r \to \infty$ limit after $N \to \infty$) supports a *unique Gibbs measure* [8]. In this case $p(x_i|\underline{y})$ and $p_{i,r}(x_i|\underline{y})$ simply correspond to two different boundary conditions and must coincide as $r \to \infty$. Unhappily, it is easy to convince oneself that this is never the case for good codes! In this case no degree 0 or 1 variables exists and a fixed boundary condition always determines uniquely x_i (and more than one such condition is admitted).

As promised above, we conclude by explaining how to obtain a better estimate of the conditional entropy $H(\underline{X}|\underline{Y})$. It turns out that this also provides a tool to tackle the other questions above, but we will not explain how. Denote by $w_i = \frac{1}{2}\log\frac{p(x_i=0|\underline{y}_{\sim i})}{p(x_i=1|\underline{y}_{\sim i})}$ the log-likelihood ratio which keeps into account all the information pertaining bits x_j, with j different from i, and let $w_i^{(r)}$ be the corresponding r-iterations BP estimates. Finally, let $w^{(r)}$ be the weak limit of $w_i^{(r)}$ (this is given by density evolution, in terms of $v^{(r-1)}$). We introduce the so-called **GEXIT** function $g(w)$. For the channel BSC(p) this reads

$$g(w) = \log_2 \left\{ 1 + \frac{1-p}{p} \, e^{-2w} \right\} - \log_2 \left\{ 1 + \frac{p}{1-p} \, e^{-2w} \right\} . \tag{32}$$

And a general definition can be found in [9]. It turns out that $\mathbb{E} g(w^{(r)})$ is a decreasing function of r (in this respect, it is similar to the entropy kernel $h(u)$, cf. Eq. (28)). Remarkably, the following **area theorem** holds

$$H(\underline{X}|\underline{Y}(p_1)) - H(\underline{X}|\underline{Y}(p_0)) = \sum_{i=1}^{N} \int_{p_0}^{p_1} g(w_i) \, \mathrm{d}p \, , \tag{33}$$

where $\underline{Y}(p_0)$, $\underline{Y}(p_1)$ denotes the output upon transmitting through channels with noise levels p_0 and p_1. Estimating the w_i's through their BP version, fixing $p_0 = 1/2$ (we stick, for the sake of simplicity to the BSC(p) case) and noticing that $H(\underline{X}|\underline{Y}(1/2)) = NR$, one gets

$$H(\underline{X}|\underline{Y}(p))/N \geq R - \int_{p}^{1/2} \mathbb{E} \, g(w^{(r)}) \, \mathrm{d}p \, . \tag{34}$$

The bound obtained by taking $r \to \infty$ on the r.h.s. is expected to be asymptotically (as $N \to \infty$) exact for a large variety of code ensembles.

VI. HISTORICAL AND BIBLIOGRAPHICAL NOTE

Information theory and the very idea of random code ensembles were first formulated by Claude Shannon in [10]. Random code constructions were never taken seriously from a practical point of view until the invention of turbo codes by Claude Berrou and Alain Glavieux in 1993 [11]. This motivated a large amount of theoretical work on sparse graph codes and iterative decoding methods. An important step was the 're-discovery' of low density parity check codes, which were invented in 1963 by Robert Gallager [12] but soon forgotten afterwards. For an introduction to the subject and a more comprehensive list of references see [13] as well as the upcoming book [14]. See also [15] for a more general introduction to belief propagation with particular attention to coding applications.

The conditional entropy (or mutual information) for this systems was initially computed using non-rigorous statistical mechanics methods [17, 18, 19] using a correspondence first found by Nicolas Sourlas [20]. These results were later proved to provide a lower bound using Guerra's interpolation technique [21], cf. also Francesco Guerra's lectures at this School. Finally, an independent (rigorous) approach based on the area theorem was developed in [9, 23] and matching upper bounds were proved in particular cases in [22].

---

[1] R. McCrum, W. Cran, and R. MacNeil, *The Story of English*, New York, Penguin 1992

[2] Y. Kabashima and D. Saad, "Belief propagation vs. TAP for decoding corrupted messages", Europhys. Lett. **44**, 668-674 (1998).

[3] J. S. Yedidia, W. T. Freeman, Y. Weiss, "Constructing Free-Energy Approximations and Generalized Belief Propagation Algorithms", IEEE Trans. on Inf. Theory, vol. 51, pp. 2282-2312, 2005

[4] M. Mézard, G. Parisi, and R. Zecchina, "Analytic and algorithmic solution of random satisfiability problems". Science **297**, 812 (2002).

[5] C. C. Moallemi and B. Van Roy, "Consensus Propagation," submitted for publication. Available at `http://www.stanford.edu∼bvr/`

[6] L. Pachter and B. Sturmfels "Tropical geometry of statistical models". Proc. Nat. Acad. Sciences vol. 101, pp. 16132-16137, 2004.

[7] D. Aldous and J. M. Steele, "The objective method: Probabilistic combinatorial optimization and local weak convergence" *Probability on Discrete Structures*, H. Kesten (Ed.), Springer, New York, 2004.

[8] S. Tatikonda and M. I. Jordan. "Loopy belief propagation and Gibbs measures". XVIII Conference on Uncertainty in Artificial Intelligence, 2002.

[9] C. Méasson, A. Montanari, T. Richardson, and R. Urbanke. "Life Above Threshold: From List Decoding to Area Theorem and MSE" 2004 IEEE Information Theory Workshop, San Antonio, Texas. Available at `http://arxiv.org/abs/cs.IT/0410028`

[10] C. E. Shannon, "A mathematical theory of communication", Bell Sys. Tech. Journal, vol. 27, pag. 379-423, 623-655, 1948. Available electronically at `http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html`

[11] C. Berrou, A. Glavieux, and P. Thitimajshima. "Near Shannon limit error-correcting coding and decoding:Turbo-Codes" 1993 IEEE International Conference on Communications, pp. 1064-1070.

[12] R. G. Gallager, *Low-Density Parity-Check Codes*, MIT Press, Cambridge (MA), 1

[13] T. Richardson and R. Urbanke, "An Introduction to the Analysis of Iterative Coding Systems", in *Codes, Systems, and Graphical Models*, edited by B. Marcus and J. Rosenthal, Springer, New York, 2001.

[14] T. Richardson and R. Urbanke, *Modern Coding Theory*, draft available at `http://lthcwww.epfl.ch/index.php`

[15] F. R. Kschischang, B. J. Frey, and H.-.A. Loeliger, "Factor Graphs and the Sum-Product Algorithm", IEEE Trans. Inform. Theory **47**, 498 (2001).

[16] B. Sturmfels, Plenary talk at the Fotry-second Allerton Conference on Communications, Control and

Computing, Monticello (IL), October 2004.

[17] T. Murayama, Y. Kabashima, D. Saad, and R. Vicente, "Statistical Physics of Regular Low-Density Parity-Check Error-Correcting Codes," Phys. Rev. **E 62**, 1577 (2000)

[18] A. Montanari, "The glassy phase of Gallager codes", Eur. Phys. J. B **23**, 121-136 (2001).

[19] S. Franz, M. Leone, A. Montanari, and F. Ricci-Tersenghi, "Dynamic phase transition for decoding algorithms", Phys. Rev. E **66**, 046120 (2002)

[20] N. Sourlas. "Spin-glass models as error-correcting codes" Nature **339**, 693-694 (1989).

[21] A. Montanari "Tight bounds for LDPC and LDGM codes under MAP decoding", IEEE Trans. Inf. Theory, vol. 51, pp. 3221–3246, 2005

[22] C. Méasson, A. Montanari, and R. Urbanke. "Maxwell Construction: The Hidden Bridge between Iterative and Maximum a Posteriori Decoding". Submitted for publication on IEEE Trans. Inf. Theory. Available at `http://fr.arxiv.org/abs/cs.IT/0506083`

[23] C. Méasson, A. Montanari, T. Richardson, and R. Urbanke. "The Generalized Area Theorem and Some of its Consequences". Submitted for publication on IEEE Trans. Inf. Theory. Available at `http://fr.arxiv.org/abs/cs.IT/cs.IT/0511039`