Assignment 3

Dmitry Adamskiy David Barber

December 12, 2018

1 LDPC codes

1.1 Background: LDPC-codes

Here we briefly review the concepts and the construction of LPDC decoder and we invite you to study [1] and [2] for more details and motivations. All the addition and multiplication operations in what follows are operations in $\mathbb{F}_2$ (that is, $0+1=1+0=1$,$1+1=0+0=0$, $1\cdot 0=0\cdot 1=0\cdot 0=0$ and $1\cdot 1=1$).

The message to be transmitted is split into blocks of length K. Let's denote one such block to be transmitted as $t \in \{0,1\}^K$. Let N be the codeword length. The code is a subset of 2^K codewords of length N and the ratio K/N is called a rate of the code. A *linear* code is a code such that all the codewords form a K-dimensional linear subspace of $\{0,1\}^N$. Such subspace could be defined by a parity check matrix H or rank $M = N - K$. The codewords are then defined as the solutions of a system of linear equations $Hx = 0$. In what follows we assume that matrix H is full-rank of size $(N-K, N)$.

1.1.1 Encoding

Suppose we are given some parity check matrix H and we need to find a generator matrix G. If we know the basis vector of codeword subspace, we could use the basis vectors as the columns of matrix G. This way $x = Gt$ is going to be a codeword. The encoding is called systematic if all the bits of t are copied to the specific location of the transmitted message x (for example in the first K bits). Then reconstructing the signal from the decoded message becomes trivial: you just read it from the first K bits. One way to build a systematic encoder G is to perform Gaussian elimination. As described in the tutorial, up to the permutation of columns, the echelon form of H is equivalent to $[PI_{N-K}]$. Then you could select G to be $[I_K P]^T$, as $HGt = (I + I)t = 0$ for every t.

1.1.2 Decoding

The probabilistic model of LDPC decoder is as follows. Let x be a transmitted vector and y the received one. The noise model specifies conditional probability

distributions $P(y|x)$. For example, in the Binary Symmetric Channel model each bit is independently flipped with probability p, so we have

$$P(y|x) = \prod_{n=1}^{N} p(y_n|x_n) = \prod_{n=1}^{N} p^{x_n - y_n}(1-p)^{x_n - y_n + 1}$$

The joint distribution of (x, y) is then defined as $p(x, y) = p(y|x)p(x)$, where $p(x)$ is a uniform prior distribution over all the valid codewords:

$$p(x) \propto \mathbb{I}[Hx = 0]$$

Decoding is done using Loopy Belief Propagation as described in the tutorial slides and in [1, p.p. 560-561].

1.2 Assignment

1. Write a function that receives a parity check matrix H and builds a systematic encoding matrix G for it. This may require altering H by swapping columns, so the function should return two matrices: $\hat{H}$ and G, such that $\hat{H}$ is equal to H up to a column permutation and $HGt = 0$ for all t (all the operations are performed in $\mathbb{F}_2$). Print the outputs of the function for the following matrix:

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

   [15 marks]

2. Draw the factor-graph for the same matrix. Write the distribution corresponding to this factor-graph and the updates used for the messages. [5 marks]

3. Write an LDPC-decoder based on Loopy Belief Propagation for Binary Symmetric Channel. Specifically, write a function that receives a parity check matrix $\hat{H}$, a received word y, a noise ratio p and an optional parameter of a maximum number of iterations (with default value of 20). The function should return a decoded vector along with the following return code: 0 for success, -1 if the maximum number of iterations is reached without a successful decoding. Try to make your code efficient. Print the result of the decoding for a given parity check matrix H_1 (in H1.txt) and vector y_1 (in y1.txt). The noise ratio was $p = 0.1$. How many iterations did your algorithm take to converge? [25 marks]

4. The original message is located in the first 252 bits of the decoded signal. Recover the original English message by reading off the first 248 bits of the 252-bit message and treating them as a sequence of 31 ASCII symbols. [5 marks]

5. Perform an empirical study of the performance of the decoder. Namely, run a sufficiently large number of encoding-decoding experiments for various levels of noise and plot the percentage of successful decodings as a function of noise (for a fixed matrix H_1). [Bonus 10 marks]

1.3 Notes on assignment

1. You are not allowed to include in your program any code from existing LDPC-related packages.

2 Mean Field Approximation and Gibbs Sampling

Consider the Ising model on the $n \times n$ lattice as in Exercise 6.7 from [3] with the potentials modified to include a temperature-like parameter β: $P(x) = Z^{-1} \prod_{i>j} \phi(x_i, x_j)$ with $\phi(x_i, x_j) = e^{\beta \mathbb{I}[x_i = x_j]}$ for i a neighbour of j on a lattice and $i > j$ (to avoid overcounting).

2.1 Assignment

You will need to compute the joint probability distribution of the top and bottom nodes of the rightmost column of the 10×10 lattice. If $x_{i,j}$ is the node in i-th row and j-th column, that would be nodes $x_{1,10}$ and $x_{10,10}$, so you need to provide the probability table for $P(x_{1,10}, x_{10,10})$. You have to do it for the three values of β: $\beta = 4$, $\beta = 1$ and $\beta = 0.01$. For each of them, you have to do it in the following three ways, printing the resulting probability distribution for each of them.

1. Perform exact inference, using techniques from Exercise 6.7. That is, treat each column as one variable with 2^n states and perform message passing on the induced factor-graph. [10 marks]
2. Use Mean Field Approximation and coordinate ascent. [20 marks]
3. Use Gibbs Sampling. [20 marks]

For each of the methods, write in your report the description of the methods and all the update equations used.

3 Notes on Grading

There are 100 marks on offer plus 10 bonus ones with the maximum grade capped at 100. Thus, it is possible to get the maximum grade even if you don't get all the marks.

References

[1] David J. C. MacKay. *Information Theory, Inference & Learning Algorithms.* Cambridge University Press, New York, NY, USA, 2002.

[2] Amin Shokrollahi. LDPC codes: An introduction. 2002.

[3] D. Barber. *Bayesian Reasoning and Machine Learning.* Cambridge University Press, 2012.