ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
School of Computer and Communication Sciences

Handout 7 Modern Coding Theory
Take Home Exam June 2, 2008

Rules:

- Please hand in the exam by Monday June 9th, noon.
- You can consult books, web pages, programs, article, lecture notes, but you are only allowed to discuss this problem with either Amin Karbase or Ruediger Urbanke. If you found any resource useful, please reference it in your report. This is just good academic practice.
- On purpose the question is quite broad and no hints are given. Rather than getting stuck for days, you can consult us any time and tell us your current thinking.
- You are expected to hand in a written report in which you document what you have done or tried and what the result was. If you made particular choices, explain your reason.
- The exam will we graded on the quality of your results as well as on the quality of the report itself (English, references, ...)

PROBLEM 1. We want to compress a binary symmetric source (BSS) at a rate R using iterative techniques. [If you are not familiar with the lossy source compression problem, please stop by so we can discuss it, or consult the book by Cover and Thomas.] A BSS emits iid Bernoulli($\frac{1}{2}$) random variables. It is well known that the smallest Hamming distortion which we can achieve with any scheme is $D_{\text{Shannon}} = h_2^{-1}(1-R)$, $0 \le R \le 1$, or in other words, that the smallest rate which we need to achieve Hamming distortion D, $0 \le D \le \frac{1}{2}$, is $R_{\text{Shannon}} = 1 - h_2(D)$, where $h_2(\cdot)$ is the binary entropy function.

Our aim is to analyze and simulate the following very simple iterative scheme. Slightly more sophisticated versions of this scheme are known to be able to achieve a performance very close to the rate-distortion function.

We use an LDGM code as discussed in class (see the course notes), with m information bits and n code bits. Let $\mathcal{C}$ denote the set of codewords of the LDGM code. Assume we are given a *source word* s of length n. (Note: there are source bits and information bits, these are not the same.) The *encoding task* consists of finding a codeword $x \in \mathcal{C}$ which has minimum Hamming distortion with s:

$$\hat{x}(s) = \text{argmin}_{x \in \mathcal{C}} d(x, s),$$

where $d(\cdot, \cdot)$ is the Hamming distance. Let u be the information word which generates x. Since $m < n$, we then have compressed the source word s to the smaller information word u. The incurred distortion is $d(x(u), s)$.

In general it is difficult to find the *best* representative x for a given source word s. We will therefore attempt to find a *good* representative in a greedy fashion.

Let us first define the specific LDGM ensemble which we consider. All variables have degree 3. The check nodes have a Poisson distribution. More precisely, for each variable node we pick 3 check nodes uniformly at random.

In order to find a good representative we proceed as follows. We are given the source word s. In the sequel we will say that check node i, $i \in [n]$, has value α, $\alpha \in \{0,1\}$, to mean that $s_i = \alpha$. We want to find the values of the information word u, so that the induced codeword $x = x(u)$ has a "small" distortion.

We proceed in a greedy fashion by fixing the value of one information node at a time. At any point in time some of the information bits have been set and some other ones are still free. Once we set the value of an information node we delete its edges from the graph. This gives us a sequence of *residual* graphs. To determine at time t which information bit to set and what value to set it to we proceed as follows. For each yet undecided information node we determine its *type*. The type of an information node is the number of connected check nodes of residual degree 1. The type is therefore an integer in the range $\{0,1,2,3\}$.

Let $N_i(t)$ denote the number of variables in the residual graph at time t of type i. Let w_i, $i \in \{0,1,2,3\}$,denote *weights*, i.e., non-negative reals. The algorithm picks an information bit of type i with probability

$$\frac{w_i N_i(t)}{\sum_{j=0}^{3} w_j N_j(t)}.$$

It then sets the value of the chosen information bit to the majority of the values of all connected check nodes that have residual degree 1. (If there is no clear majority the algorithm picks the value uniformly at random.) The algorithm adds the chosen value to all connected check nodes and then removes all the edges adjacent to the chosen variable.

After m steps the algorithm stops. We are interested in the resulting distortion of this scheme for given weights w_i as well as how we should choose these weights in order to minimize the distortion.

You are asked to analyze this scheme as well as to implement it and to show that your analysis matches the simulation results. How does your result compare to the Shannon rate-distortion bound?

Once you managed the simplest case there are many possible extensions you can try. Can you think of ways to improve upon the given scheme in order to achieve a better rate-distortion performance? What is the best scheme you can think of and how well does it perform?