Background

In addition to turbo codes, LDPC codes is a class of codes decoded iteratively and with good practical performance. LDPC codes were originally discovered by Gallager in the early 1960s and rediscovered by MacKay and Neal in 1996.

LDPC codes are occasionally called **Gallager codes**.

LDPC Codes (1)

Low-density parity check (LDPC) codes are

- linear block codes with
- a sparse parity check matrix $\boldsymbol{H}$.

Sparse means that most of the elements are 0. Note that the direction of constructing matrices is opposite to the normal one: design $\boldsymbol{H}$ and then calculate a generator matrix $\boldsymbol{G}$, not design $\boldsymbol{G}$ and then calculate $\boldsymbol{H}$.

LDPC Codes (2)

regular LDPC code An LDPC code with constant number of 0s per row and per column.

irregular LDPC code An LDPC code that is not regular.

Alphabet for LDPC codes: GF(2), GF(4), GF(8), GF(16),. . . . Generally: better performance with bigger alphabet.

(Pseudo-)randomness occurs in turbo and LDPC codes in the interleaver and in the parity check matrix, respectively.

Irregular LDPC codes in general perform better than regular LDPC codes.

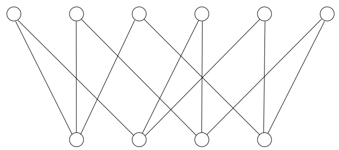
Tanner Graph

bipartite graph A graph with vertex set $V = V_1 \cup V_2$, where each edge has one endpoint in V_1 and one in V_2.

A **Tanner graph** of an LDPC code with parity check matrix $\boldsymbol{H}$ has one vertex in V_1 for each row of $\boldsymbol{H}$ and one vertex in V_2 for each column of $\boldsymbol{H}$, and there is an edge between two vertices i and j exactly when $h_{ij} \neq 0$.

Example: Tanner Graph

$$\boldsymbol{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$



Cycles of Tanner Graphs

Short cycles of Tanner graphs have a negative impact on decoding. Cycles necessarily have even length and length 2 is not possible.

Avoiding cycles of length 4: The intersection of positions in which two columns have nonzero values should be at most 1.

The requirement that a Tanner graph should not have short cycles is an intricate part in the construction of good LDPC codes.

Note. The degrading effect of short-length cycles diminishes as the code length increases and is strongly reduced with length > 1000 bits.

Obtaining Generator Matrices

To obtain the generator matrix, the parity check matrix is converted into systematic form—for example, using Gaussian elimination—after which the transformation is straightforward.

LDPC Code Types

LDPC codes can be divided into

- random LDPC codes and
- structured LDPC codes.

The best known codes are of the former types. Structured LDPC codes can be constructed from various types of combinatorial objects (designs, geometries,. . .).

Decoding LDPC Codes

Decoding LDPC codes is an iterative process of interchanging information between the two types of nodes of the corresponding Tanner graph. If

- at some point of the iterative process the syndrome of the estimated decoded vector is the all-zero vector, this result is output;
- the iterative process has not converged to a solution after a predetermined number of iterations, decoding failure is declared.

See [MF, Fig. 8.5] for the impact of the maximum number on iterations on the BER performance.

Calculating Estimates

The core part of the turbo decoding algorithm in the previous lecture is the BCJR algorithm. The core part of the LDPC decoding algorithm is the **sum-product algorithm**, or **belief propagation algorithm**.

These algorithms are maximum a posteriori (MAP) algorithm—recall that the Viterbi algorithm is a maximum likelihood (ML) algorithm. It is a matter of estimating symbols versus estimating codewords.

Some Practical Aspects of Decoding

- ▷ Polar format should be used instead of binary format.
- ▷ With logarithmic calculation, products and divisions are converted into additions and subtractions, respectively (cf. turbo coding slides).
- ▷ Look-up tables for parts of the logarithmic calculations save a lot of time and do not have a significant impact on the BER performance.

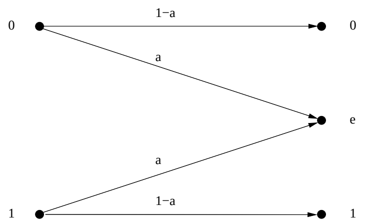
Turbo vs. LDPC Codes

Turbo codes: Very good BER performance for intermediate block length.

LDPC codes: Very good BER performance for long block length (for example, BER performance with length $n = 10,000$ that is less than 0.1 dB from the Shannon limit).

Erasure Channel

The code type to be discussed next is designed for the **erasure channel** rather than AWGN, BSC, and similar channels assumed so far.



Fountain Code

Consider a (binary) message consisting of K data packets of m bits each, that is, of total length Km. (It is common that communication protocols transmit information in packets.)

Fountain code: The transmitter continuously transmits packets of m bits that are obtained by XORing—that is, adding modulo 2—subsets of the packets. The receiver collects (just a little bit more than) K packets to retrieve the original message.

Decoding Fountain Codes

Decoding fountain codes means solving a system of equations. Call the blocks $B_1, B_2, \ldots B_K$.

Example. $K = 3$, $m = 4$. Assume that for the received blocks we have

$$\begin{aligned} B_1 + B_3 &= 0100, \\ B_2 + B_3 &= 1110, \\ B_1 + B_2 + B_3 &= 0000. \end{aligned}$$

Adding the second and the third equation gives $B_1 = 1110$, and then $B_3 = 1010$ and $B_2 = 0100$.

Random Fountain Codes

How to form the packets? One possibility: Random combinations/sums of packets. The indices of the packets involved must be known also by the receiver. When N such packets have been received

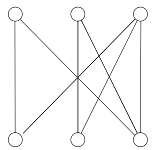
- if $N < K$, decoding is not possible,
- if $N = K$, decoding is possible with probability about 0.289,
- if $N = K + \Delta$, decoding is possible with probability at least $1 - 2^{-\Delta}$.

Bipartite Graphs for Fountain Codes

Decoding fountain codes is about solving a system of equations, which can be rather time-consuming if K is large.

As with LDPC codes, a bipartite graph may be useful in the decoding process, with one set of nodes corresponding to the blocks (variables) and the other to the received words.

Example. (cont.)



Luby Transform Codes

Luby transform (LT) codes are improved random fountain codes.

- ▷ Random combinations/sums have only a few packets.
- ▷ The number of packets in the sums are given by an optimized distribution function.
- ▷ Decoding is straightforward due to equations of the form $B_i =?$ throughout the calculations.