Form groups of 2–3 persons and work on a research-oriented project on information and coding theory. You need to culminate your work during the application period. In week 8 you need to make a presentation and provide us with final report as well as a source code.

In what follows we list possible project areas, initial project descriptions, bibliography in the area and our expectations. You need to express interest in up to 3 of these areas, or come up with your own suggestion, by March 8 and send them by e-mail to TAs. Groups will be formed after your suggestions are received.

1 Channel Coding Rate in the Finite Blocklength Regime

Fundamental results of information and coding theory (Shannon's theorem) claim that if the rate is less than the channel capacity, then there exists a code of large length, reaching an arbitrarily small probability of error. Unfortunately, things change in the case of a finite (small) length. Over the past few years significant progress was obtained in this area of research. Let us mention here the paper, where the bound on achievable rates for a fixed (small) length (N) and a given probability of wrong decoding (P_w) was obtained.

Our expectations:

1. Implement a tool to plot the bound on the achievable coding rate
2. Plot 2 curves of coding rate in dependence of SNR:
   (a) BI-AWGN channel, $N = 100$, $P_w = 10^{-4}$;
   (b) BI-AWGN channel, $N = 10000$, $P_w = 10^{-4}$;

References and Resources:

- Y. Polyanskiy, H. V. Poor and S. Verdu, "Channel Coding Rate in the Finite Blocklength Regime," in IEEE Trans. on Inf. Theory, vol. 56, no. 5, pp. 2307-2359, May 2010.

2 Soft-Decision Decoding of Reed–Solomon Codes

Reed–Solomon codes are the most popular and beautiful algebraic codes. A powerful hard-decision Guruswami–Sudan algorithm allows to decode errors beyond half of the minimum distance. It is well known, that soft decision decoders have better performance in comparison to hard decision ones. Koetter and Vardy found a way how to modify Guruswami–Sudan algorithm, such that it can use soft information (log likelihood ratios).

Our expectations:

1. Find the implementation of Guruswami–Sudan algorithm (or contact TAs to obtain the source code);
2. Modify Guruswami–Sudan in accordance to the article of Koetter and Vardy;
3. Compare the performance of Guruswami–Sudan and Koetter–Vardy decoders, i.e. plot probability of wrong decoding (P_w) in dependence of SNR for BI-AWGN channel. Consider RS codes with parameters:

(a) $RS(N = 63, K = 56)$ (high rate regime, $R = 8/9$)
(b) $RS(N = 63, K = 7)$ (low rate regime, $R = 1/9$)

References and Resources:

- R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," in IEEE Trans. Inf. Theory, vol. 49, no. 11, pp. 2809-2825, Nov. 2003.

3 Sum-Product Algorithm for LDPC Codes

Sum-Product algorithm (SPA) is a message passing algorithm, which works on the Tanner graph, corresponding to LDPC code. It is known, that SPA is equivalent to maximum a posteriori probability (MAP) decoder if a Tanner graph is a tree. Unfortunately, tree codes have minimum distance equal to 2 and bad error correcting capabilities. So SPA is usually applied for loopy Tanner graphs.

Our expectations:

1. Implement an SPA;
2. TAs will provide you with LDPC parity-check matrix and encoding function (written in MATLAB);
3. Choose a digit from MNIST database
http://yann.lecun.com/exdb/mnist/
4. Encode the digit, modulate ($0 \to +1$, $1 \to -1$), add Gaussian noise (choose an SNR value yourself, such that the decoding is possible);
5. Use the decoder to recover the original digit. Show the "evolution" of the decoding process for iterations $1 \ldots 15$.

References and Resources:

- F. R. Kschischang, B. J. Frey and H. A. Loeliger, "Factor graphs and the sum-product algorithm," in IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 498-519, Feb 2001.

4 Density Evolution for LDPC Codes

Irregular LDPC codes are known to have better thresholds (waterfalls) than regular LDPC codes. Density evolution is a method, that allows to calculate an asymptotic threshold given polynomials $\rho(x)$ (row weight distribution) and $\lambda(x)$ (column weight distribution). Gaussian approximation allows to reduce the complexity of this method.

Our expectations:

1. Implement density evolution method with use of Gaussian approximation;
2. Find optimal (in sense of threshold) distributions for LDPC codes of rate $R = \{0.9, 0.5, 0.2\}$.

References and Resources:

- S. Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of Sum-Product Decoding of Low-Density Parity-Check Codes Using a Gaussian Approximation", IEEE Trans. Inf. Theory, Vol. 47, No. 2, pp. 657-670, Feb. 2001

Cont.

5 ACE Constrained Progressive Edge-Grows Algorithm

LDPC codes are known to suffer from high error floor. The reasons of error floors are pseudo codewords (or trapping sets). Trapping sets always contain short cycles. There are numerous algorithms how to eliminate short cycles in the Tanner graph. We suggest you to implement one of them.

Our expectations:

1. Implement the algorithm, explained in the paper;
2. TAs will provide you with optimal degree distributions for LDPC of rate 0.9;
3. Construct a parity-check matrix of LDPC code with $R = 0.9$ with use of implemented algorithm;
4. TAs will provide you with LDPC simulation platform;
5. Perform a simulation and check error floor;

References and Resources:

- D. Vukobratovic and V. Senk, "Generalized ACE Constrained Progressive Edge-Growth LDPC Code Design," in IEEE Communications Letters, vol. 12, no. 1, pp. 32-34, January 2008.

6 List decoding of Polar Codes

Polar codes are a new and interesting class of codes proposed by E. Arikan in 2009. The usual way to decode polar codes is to apply successive cancellation algorithm. An improvement was suggested by Tal and Vardy. The idea is to consider a list of possible codewords at each step. List decoder significantly improves the performance.

Our expectations:

1. Implement the list decoder;
2. TAs will provide you with an encoding function (written in MATLAB);
3. Choose a digit from MNIST database
   http://yann.lecun.com/exdb/mnist/
4. Encode the digit, modulate ($0 \to +1$, $1 \to -1$), add Gaussian noise (choose an SNR value yourself, such that the decoding is possible);
5. Use the decoder to recover the original digit. Consider list size $L = 1$ and $L = 8$;

References and Resources:

- I. Tal and A. Vardy, "List Decoding of Polar Codes," in IEEE Trans. on Inf. Theory, vol. 61, no. 5, pp. 2213-2226, May 2015.

The End.