Project Logistics

• Projects can be done in groups of 1–3 students. A list of suggested projects is given below. You’re welcome to suggest an idea for a project not on the list which would require approval from the teaching staff. Please submit your preference form by April 26, in which you should indicate your preference ranking among three of the suggested projects (1 represents “most prefer” and 3 represents “least prefer”), and whether you would like to propose your own project. If you would like to suggest your own project, please submit along with the form a brief proposal on your project (maximum two pages), including the title, abstract, and your ideas. For theoretical projects your proposal should also review the state-of-the-art, and for practical ones you should describe the algorithmic approaches you plan to explore, and the datasets you would like to use.

• Preference match: Since we do not want to see too many duplicate projects, we will assign the ultimate project topic to you after collecting your preference forms. We will try to meet your preference as much as possible, and also evaluate your proposal if you propose your own project. Our decisions will be sent to you via email by April 30.

• Milestone report: Each student group should submit a written report on the milestone progress of the project by May 21. The report should not exceed 3 pages in the two-column format, and should summarize the state-of-the-art on your chosen topic. Note that by that time point you should strive to complete approximately half of the tasks in your chosen project.

• Presentation: Each student group will give a short presentation of their project. The presentation will be on June 8. We will provide you with the presentation format and a schedule later.

• Final report: Each group should submit a written report on the project by June 14. The report should not exceed 4 pages in the two-column format. You should email your final report (and source codes if you are doing a practical project) to the course assistants. Note that the code should be well documented, and the results should be replicable. To achieve this goal, we strongly recommend that you use the Matlab publish mode or IPython notebook.

Project List

The following are some suggested projects. We provide an incomplete list of references under each project title to give a sense of the project idea. You’ll be digging up more references of relevance for the project you end up pursuing.

1. Deletion and substitution DUDE

   This project is dedicated to the development and application of a variation on the Discrete Universal Denoiser (will be covered in class) to accommodate, beyond substitution errors (for which DUDE was designed).

   References:

   • Chapter 8 of the 2013 Course Notes of EE378A.

2. Neural DUDE
This project is dedicated to understanding the recently introduced “Neural DUDE” and exploring possible variations on it.
References:

3. Denoiser competing with given set of denoisers
This project is about the combination of the denoising framework with a learning theoretic one of competition with a set of experts: given a set of denoisers, can we construct one that will do essentially as well as the best of them?
References:

4. Applications of directed information estimation
This project is about using the directed information estimator to infer the causal relationships in a network, e.g., the influence flow in neuro-scientific data or the financial market.
References:
• Codes for directed information estimation: http://web.stanford.edu/~tsachy/DIcode/

5. Applications of relative entropy estimation
This project is about using the relative entropy estimator to measure the discrepancy of two discrete distributions in practice, e.g., estimate the error exponent in the outlier detection, or do clustering based on the relative entropy distance.
References:
• Y. Han, J. Jiao, and T. Weissman, “Minimax Rate-Optimal Estimation of Divergences between Discrete Distributions”, available on arXiv.
• Codes for relative entropy estimation: http://web.stanford.edu/~tsachy/index_hjw.htm

6. Information theoretic gene regulatory network inference
This project is about using information theoretic functionals, such as the mutual information, to infer the structure of gene regulatory networks. You are expected to explore the use of the recently proposed minimax rate-optimal entropy and mutual information estimators in these important genetic applications, and compare them with existing approaches. A concrete list of references in genetics are available for download at the project page of the course website.

7. Fundamental limits in language modeling
This project is about estimating the fundamental limits in language modeling. As we shall argue in class, estimating fundamental limits is usually easier than achieving fundamental limits. You are expected to use the recently proposed minimax rate-optimal entropy estimators to estimate the entropy
rate of natural English languages, and to compare the results to the performance of existing language models. You may choose to use the PTB dataset, or the much larger One Billion Word Benchmark. The references provide detailed background information. Note that in language modeling, people usually use "perplexity" to measure the performance of language models, which is equivalent to the entropy rate up to a bijection.


8. Estimation of entropy rate

This project is dedicated to estimating the entropy rate of a stationary stochastic process. You are supposed to extend the idea used in the approximation-based estimator (will be covered in class) in the iid case. You are expected to compare the performances of various entropy rate estimators (e.g. see [http://bactra.org/notebooks/entropy-estimation.html](http://bactra.org/notebooks/entropy-estimation.html)) in simulated data, which may include various methods inspired by data compression, and methods that directly focus on estimating the entropy rate. For the latter category, you are expected to also compare these two approaches: estimating $H(X_0|X_{-k}^{-1})$ directly by conditioning on each realization of $X_{-k}^{-1}$, or estimating it via $H(X_0|X_{-k}^{-1}) = H(X_0^{-k}) - H(X_{-k}^{-1})$.