Time, Place, Formalities

The exam will take place on Wednesday, August 8, 9:00-11:00 in HIL F61 (ETH Hönggerberg). No material is allowed.

Exam Style

The exam will be similar in style to the exams of the course Algorithms, probability and Computing (APC). On the website of the Fall 2017 edition, you find a collection of old APC exams: https://www.ti.inf.ethz.ch/ew/courses/APC17/index.html

We are not interested in checking whether you are able to memorize all definitions and proofs. We expect you to know some basic definitions and techniques that have frequently been used in the course and the exercises, and to be able to apply them in similar and related contexts (see the learning goals below).

To this end, we will ask you to provide small proofs and analyze simple examples. Whenever the questions require material beyond basic definitions and techniques, we will provide you with this material.

There might also be multiple-choice questions for examining concept understanding.

Generally, we put more emphasis on a high-level understanding and a good overview of the course material, rather than on technical details. There will be technical questions, but they can be mastered using the basic definitions and techniques, the extra material that we may provide, and good general math and computer science skills.

Mastery of the exercises handed out during the course is an excellent preparation for the exam. The exam might actually contain questions that are very similar to exercises that have been handed out.
Learning Goals

You should

- know the definitions of a convex set and a convex function, the first- and second-order characterizations of convex functions, tools to construct convex functions from others, as well as criteria for the existence of a (unique) minimizer;
- be able to prove (non-)convexity of concrete functions;
- know the definition of the (projected) Gradient Descent algorithm, and its runtime bounds for the case of bounded gradients, smooth as well as strongly convex functions. There is no need to memorize proofs of these bounds;
- be able to analyze the behavior of the (projected) Gradient Descent algorithm on simple example functions;
- know the definitions of the subgradient and stochastic Gradient Descent algorithms and understand their purposes;
- know the definition of Newton’s Method and its convergence behavior near a minimizer (no need to memorize proofs); understand the advantages and disadvantages of Newton’s Method, compared to Gradient Descent;
- be able to analyze the behavior of Newton’s method on simple example functions;
- know the definition of a Quasi-Newton Method and the rationale behind Quasi-Newton methods;
- understand the main ideas that lead to the BFGS and the l-BFGS methods (no need to memorize the details of these methods);
- know the general setup for statistical estimation problems (some keywords: parameterized distribution, estimation error, measurement, noise);
- know basic properties of Gaussian distributions (e.g., the facts that Gaussian distributions are determined by their mean and their covariance and that Gaussian distributions are preserved under linear transformations);
- know the definition of maximum likelihood estimation and understand the challenges around it, e.g., lack of universal guarantees (at least in the high dimensional case) and computational intractability (even in simple cases);
- know the definition of linear regression as a statistical estimation problem; the relationship of least-squares optimization and likelihood maximization for linear regression; understand the statistical guarantees of least squares and its analysis;
- understand lower bounds for estimation errors for linear regression (e.g., linear unbiased estimators and using Bayesian priors);
- know the definition of sparse linear regression as a statistical estimation problem; know about best-subset selection and LASSO estimators; know about their statistical guarantees and their analyses;
• know the definitions of low-rank estimation problems and estimators for them based on singular value decompositions; know about their statistical guarantees and their analysis;
• be able (in simple cases) to design estimators for statistical estimation problems and analyze their statistical guarantees.