Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Department of Computer Science
Institute of Theoretical Computer Science
Bernd Gärtner and Niao He

Optimization for Data Science **Learning Goals: All Lectures** **FS23**

This document summarizes the key concepts taught in all lectures, and it provides concrete learning goals: things that you should be able to do after studying the material in these lectures. The weekly exercise sheets, clicker questions, **and in particular the graded assignments** are designed to check the learning goals of the lectures they pertain to. In the final exam, we may check any of the published learning goals, but we will *only* check learning goals that have been published.

Introduction

Key concepts

Learning problem, Data source, Hypothesis, Loss function, Expected risk, Expected risk minimization, Empirical risk, Empirical risk minimization, PAC learning, Worst-case complexity, Average-case complexity, Estimation error, Optimization error, Estimation-optimization tradeoff.

Learning goals

You should know ...

- what learning problems and algorithms are, and how they differ from standard optimization problems and algorithm as typically taught in introductory computer science courses;
- why worst-case complexity is not an appropriate measure for learning algorithms, and what the alternatives are;
- what the major sources of errors are in solving a learning problem.

You should be able to ...

- mathematically compute expected and empirical risk in a given application;
- use known sufficient conditions for PAC learning in a given application;
- argue why empirical risk minimization fails to solve the learning problem in a given application.

Theory of Convex Functions

Key concepts

Convex function, Differentiable function and gradient, First-order characterization of convexity, Twice differentiable function and Hessian, Second-order characterization of convexity, Operations preserving convexity, Strictly convex function, Necessary and sufficient optimality conditions, Optimization problem (in standard form), Convex program, Weak and strong Lagrange duality, Karush-Kuhn-Tucker conditions.

Learning goals

You should know ...

- what convex optimization problems are, and why they are attractive for theoretical studies;
- that there are a number of standard tools for proving that an optimization problem is convex, and that an optimal solution has been found;
- that differentiability of the objective function is a prerequisite for most of these tools to work;
- that convex functions occur in real-world applications.

You should be able to ...

- compute gradients and Hessians of given functions;
- prove (strict) convexity or non-convexity of given functions;
- mathematically compute (constrained) minimizers of (convex) functions;
- compute and argue about the Lagrange dual of a given optimization problem;
- use the Karush-Kuhn-Tucker conditions in a given application.

Gradient Descent

Key concepts

Gradient descent algorithm, Vanilla analysis, Lipschitz convex functions in $O(1/\varepsilon^2)$ steps, Smooth convex functions in $O(1/\varepsilon)$ steps, Sufficient decrease, Smooth and strongly convex functions in $O(1/\log(\varepsilon))$ steps, Accelerated gradient descent in $O(1/\sqrt{\varepsilon})$ steps on smooth functions, Characterizations of smooth and strongly convex functions.

Learning goals

You should know . . .

- why gradient descent is a natural approach to convex optimization;
- that all theoretical analyses of gradient descent require assumptions beyond convexity on the function to be optimized;
- that theoretical analyses provide good guidance for optimization algorithm design but cannot explain all practical success stories.

You should be able to . . .

- prove smoothness or non-smoothness of a given function;
- prove strong convexity or its absence for a given function;
- analyze gradient descent and variations of it on a given function or class of functions.

Projected Gradient Descent

Key concepts

Constrained optimization, projected gradient descent algorithm, projected sufficient decrease, constrained vanilla analysis, smooth convex functions in $O(1/\varepsilon)$ steps, the projection step

Learning goals

You should know. . .

- that projected gradient descent requires the same number of steps as gradient descent on the function classes that we have studied;
- that projected gradient descent requires a nontrivial primitive to be solved in each step (projection onto the feasible region);
- that this is efficient for a number of feasible regions, including affine spaces, ℓ_2-balls, and ℓ_1-balls, but may require more work in other cases.

You should be able to. . .

- prove smoothness or non-smoothness of a given function over a set X;
- prove (strong) convexity or its absence for a given function over a set X;
- analyze projected gradient descent and variations of it on a given function or class of functions;

- analyze the projecton step for a given set X.

Coordinate Descent

Polyak-Łojasiewicz inequality as a weaker version of strong convexity, convergence analysis via PL inequality and smoothness, coordinate-wise smoothness, coordinate descent algorithms, coordinate-wise sufficient decrease, randomized coordinate descent, importance sampling, steepest coordinate descent, strong convexity w.r.t. ℓ_1-norm, Steeper coordinate descent, Greedy coordinate descent and failure to converge, separable functions, Lagrange dual version of LASSO

Learning goals

You should know...

- that the PL inequality is implied by strong convexity and allows for a simple analysis of gradient descent;
- that the speedup in coordinate descent (smaller cost per iteration) is generally counterbalanced by a correspondingly higher number of iterations;
- that there are some scenarios where coordinate descent actually leads to net speedups;
- that greedy coordinate descent may fail to converge in the non-differentiable case, but that it converges in the separable case.

You should be able to...

- prove coordinate-wise smoothness of a given function;
- prove the PL inequality for a given function or class of functions;
- analyze coordinate descent algorithms and variations of them on a given function or class of functions.

Nonconvex Functions

Key concepts

Nonconvex function, concave function, smoothness and bounded Hessians, "convergence to a critical point" ($||\nabla f(x)||^2 \to 0$), hardness of local optimality, trajectory analysis

Learning goals

You should know ...

- that stepwise methods such as gradient descent cannot be expected to find the global minimum of a nonconvex function;
- that a number of useful tools such as sufficient decrease and PL inequality do not require convexity;
- that "convergence" in the nonconvex regime typically just means that the gradient norms tend to zero;
- that it is computationally hard (co-NP complete) to check whether a given critical point is a local minimum;
- that for particular nonconvex functions and particular algorithms, convergence to a global minimum can be proved, by exploiting specific properties of the function and/or the algorithm.

You should be able to ...

- compute or bound gradients, Hessians, smoothness and related parameters of (nonconvex) functions over given sets X;
- analyze gradient descent or other stepwise optimization algorithms on nonconvex functions, using information that is available to you in the given application;
- interpret convergence results w.r.t. their usefulness and limitations.

The Frank-Wolfe Algorithm

Key concepts

Linear minimization oracle (LMO), atoms, efficiently solvable instances of LMO, duality gap, duality gap based bound for the optimality gap, smooth convex functions in $O(1/\varepsilon)$ steps, stepsize variants, affine invariance, curvature constant, affine invariant analysis, sparse solutions

Learning goals

You should know ...

- that the Frank-Wolfe algorithm reduces convex optimization over X to linear optimization over X;
- that linear optimization over X can be easy (and easier than projection onto X) in relevant cases;

- that the Frank-Wolfe algorithm is affine invariant;
- that the solutions returned are convex combinations of atoms, one atom per step of the method;
- that the optimality gap of the current iterate can be bounded via the duality gap, a quantity available in each step.

You should be able to ...

- bound smoothness, diameter, or curvature constant of a given convex optimization problems (f, X);
- Analyze the Frank-Wolfe algorithm or related methods on given convex functions f and feasible regions X;
- Provide algorithms for and analyses of the linear minimization oracle in given applications.

Newton's Method and Quasi-Newton methods

Key concepts

Newton-Raphson method for zero-finding, Babylonian method for root-finding, Newton's method for optimization, alternative views (adaptive gradient descent, minimizing second-order Taylor approximation in each step), local convergence in $O(\log\log(1/\varepsilon))$ steps, secant method, secant condition, Quasi-Newton method, Greenstadt's family, BFGS method, L-BFGS method

Learning goals

You should know ...

- that under suitable conditions, Newton's method is very fast when starting close to optimality, but that general global convergence results are unknown;
- that the steps of Newton's method are computationally expensive for large d;
- that Quasi-Newton methods are Hessian free-approximations of Newton's method, in the same way that the secant method is a derivative-free approximation of the Newton-Raphson method;
- that Greenstadt's family is derived from the idea of keeping the approximations of the Hessians as stable as possible over the iterations;
- that BFGS and L-BFGS are popular Quasi-Newton methods, and that BFGS is a member of Greenstadt's family;

- that convergence results for Quasi-Newton methods are known only in very specific settings.

You should be able to ...

- bound inverse Hessians, and Lipschitz parameters of Hessians for given functions over given sets X;
- analyze Newton's method and related methods on given functions;
- check whether a given method is Quasi-Newton, and develop Quasi-Newton methods yourself, based on given requirements;
- prove basic facts about Newton's method or a given Quasi-Newton method.

Subgradient Methods

Smooth vs nonsmooth functions, subgradient and subdifferential, properties of subgradients, subgradient descent, asymptotic convergence under different stepsizes, convergence rate analysis for convex and strongly convex objectives, lower complexity bound

Learning goals

You should be able to...

- compute the subgradient or subdifferential set of a given function;
- analyze subgradient methods and variations of them on a given function or class of functions;
- explain when and why subgradient methods converge

Stochastic Gradient Descent

Stochastic optimization vs finite-sum optimization, stochastic gradient descent, adaptive stochastic methods (Momemtum SGD, AdaGrad, Adam, AMSGrad), convergence rate analysis of SGD for convex and strongly convex objectives, strong growth condition, lower complexity bounds, nonconvergence of Adam

Learning goals

You should be able to...

- construct unbiased stochastic gradient of a given function and analyze its variance;
- understand the key differences among GD, SGD, adaptive methods and identify their pros and cons;
- understand the design principles behind adaptive stochastic methods;
- explain when and why (adaptive) stochastic gradient methods converge;
- analyze SGD and its variations of on a given function or class of functions.

Stochastic Variance-reduced Methods

finite-sum optimization, complexities of GD vs SGD, variance reduction techniques, stochastic variance reduced methods (SAG, SAGA, SVRG, etc.), convergence rate analysis of SVRG for smooth strongly convex objectives

Learning goals

You should be able to...

- identify the fundamental differences between finite-sum problems and stochastic optimization;
- explain the differences among GD, SGD, variance-reduced methods, their pros and cons;
- identify key features of difference variance-reduced methods and their limitations;
- understand variance reduction techniques and explain why and how variance-reduced methods work;
- analyze SVRG and its variations of on a given function or class of functions.

Modern Second-order Methods and Nonconvex Optimization

Key concepts

cubic regularization, local vs. global convergence, first-order vs. second-order stationary points, strict saddle points, convergence to local optimality

Learning goals

You should know ...

- that Newton method does not always enjoy global convergence;
- that even for strongly-convex smooth objectives, global analysis of Newton method can be worse than GD;
- that many techniques are useful to overcome the local nature of Newton method such as line-search, damping, regularization, trust region, etc.
- that cubic regularization is a provably globally convergent second-order method with guaranteed convergence to second-order stationary points
- that nonconvex functions can have exponentially many local minima
- that GD/SGD only converges to first-order stationary points, which can be local minima, local maxima, or saddle points
- that for particular nonconvex functions and particular algorithms, convergence to a global minimum can be proved, by exploiting specific properties of the function and/or the algorithm.

You should be able to ...

- prove properties for functions with Lipschitz gradients and Lipschitz Hessians
- understand the global analysis of cubic regularization
- analyze convergence of SGD and variants
- interpret convergence results w.r.t. their usefulness and limitations.
- identify where does the algorithm converge to
- be aware of techniques for escaping from saddle points such as random initialization, noisy perturbation

Modern Nonsmooth Optimization

Key concepts

general norms, dual norm, general smoothness/strong convexity, convex conjugate, Fenchel inequality, Bregman divergence, Mirror Descent, Nesterov's smoothing, Yosida-Moreau smoothing, randomized smoothing

Learning goals

You should know ...

- that leveraging non-Euclidean geometry is often useful to improve the dimension dependence
- that Mirror descent is a generalization of gradient descent, projected subgradient descent
- that smoothing techniques are common to efficiently handle nonsmooth problems through smooth approximations
- that there exists a tradeoff between the smoothness, approximation accuracy and computation efficiency

You should be able to ...

- prove key properties of Bregman divergence;
- analyze the convergence of Mirror Descent and interpret the bounds compared to GD
- analyze the smoothness and approximation accuracy of different smoothing techniques
- identify the usefulness and limitations of different smoothing techniques

Min-Max Optimization

Key concepts

c saddle point (Nash equilibrium), global minimax points (Stackelberg equilibrium), minimax theorem, existence of saddle points, duality gap, gradient descent ascent (GDA), extragradient method, proximal point algorithm, concave games, variational inequalities with monotone operator, algorithms for VIs

Learning goals

You should know ...

- that min-max optimization problems are ubiquitous in machine learning
- that variational inequalities are more general forms for minimization, min-max optimization, or games.
- that saddle point may not always exist, but for convex-concave problems with bounded domains, saddle point exists.

- that existence of saddle point implies minimax theorem and vice versa.
- that simple GDA will not always converge with constant stepsize
- that extradient method are optimal for solving min-max problems for several specific settings
- that algorithms for min-max optimization can be generalized to solve VIs with monotone operators

You should be able to ...

- prove existence or non-existence of saddle points for a specific example
- prove basic facts about GDA, Extragradient method, proximal point method, or a given variant
- identify the strength and limitations of different first-order methods for min-max optimization
- be aware of the fundamental differences between convex and nonconvex settings