IE 510 Applied Nonlinear Programming

Lecture 0: Introduction

Ruoyu Sun

Jan, 2022

Outline

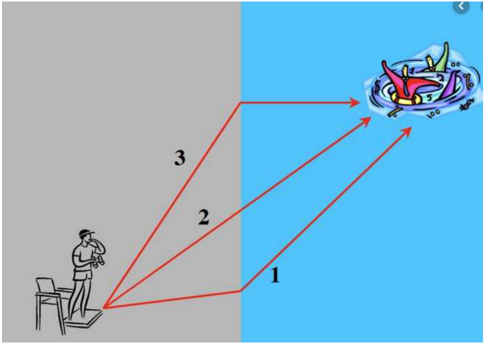
Introduction: What is Optimization

Course Introduction

Examples in Machine Learning

History of Optimization: Fermat's Principle

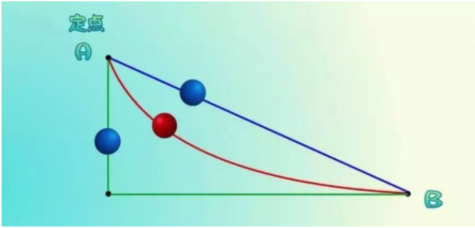
Fermat's principle: light travel in shortest path



Nature is searching for "optimum"!

History of Optimization: Brachistochrone

Bernolli's chllenge 1696: Brachistochrone (shortest time curve).



Solved by John Bernolli, Netwon, Leibniez, etc.

Euler invented calculus of variations.

History of Optimization: Early Methods

17th and 18th centry: **Fermat** (born in 1601) and **Lagrange** (born in 1736) first found calculus-based formulae for identifying optima.

18th and 19th centry: **Newton** and **Gauss** (1824) first proposed iterative methods to search for an optimum.

- Newton method
- Gauss-Seidel method

19th century: Steepest descent method (rooted in unpublished notes of **Riemann** in 1863.

We will learn them in this class (major contents of first half).

Reference: https://empowerops.com/en/blogs/2018/12/6/brief-history-of-optimization.

History of Optimization: since 1900

Linear programming: **Kantorovich** (1939, Nobel prize); George **Dantzig** (1947, Stanford); **John von Neumann**.

1940s-1970s: classic optimization approaches were developed rapidly and peaked in the 1970s.

1980-2000: advanced nonlinear algorithms (BB, BFGS, ALM, ADMM, etc.); interior point methods

After 2010: large-scale optimization

- Revisiting CD, SGD, ADMM

What is Optimization (in general)

What is optimization?

Example 1: physics. Light travels in shortest path

Example 2: AI. Is our brain doing optimization?

Example 3: Operations of companies.
McKincy efficiency manual: to improve the efficiency, you need to optimize the process/allocation/etc.

What is Optimization (in general)

What is optimization?

Example 1: physics. Light travels in shortest path

Example 2: AI. Is our brain doing optimization?

Example 3: Operations of companies.
McKincy efficiency manual: to improve the efficiency, you need to optimize the process/allocation/etc.

What is Optimization (mathematically)

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in X \end{array} \tag{1}$$

- x: optimization variable or decision variable (discrete, continuous)
- f: objective function (differentiable, convex, linear...)
- X: feasible region (convex, nonempty,...)

Three key elements of an optimization problem.

Key questions:

- **Existence**: Does the optimal solution exist?
- **Checking**: How to determine whether a feasible x is an optimal solution?
- **Algorithm**: How to find an optimal solution? (not by exhaustive search)

Main Topics

- **Unconstrained Optimization**
  1. Optimality Conditions
  2. Gradient Descent (GD) Methods and Momentum
  3. Coordinate Descent Methods (CD)
  4. Stochastic Gradient Descent (SGD)
  5. BB and BFGS
- **Constrained Optimization**
  1. Optimality Conditions (KKT Conditions)
  2. Gradient Projection Methods
  3. Penalty Method
  4. ALM (Augmented Lagrangian Methods), a.k.a., Method of Multipliers
  5. ADMM (Alternating Direction Method of Multipliers)

Main Topics (continued)

Other topics:

- Understanding convergence of first-order methods: importance of spectrum
- DFO (derivative free optimization)
- min-max optimization and bi-level optimization
- How to apply the knowledge? Applications in:
  - Machine Learning
  - Signal Processing

Difference With Other Courses

Difference with other courses:

Linear programming course: focus on linear programming. Won't talk about general gradient methods in detail

Convex optimization: focus on convex problems, especially linear programming and conic programming. Not so much on nonlinear problems (most machine learning problems are nonlinear)

Machine learning course: won't talk about convergence issue, and constrained problems.

My comment: if you want to understand machine learning algorithms, this course is the best fit (as introductory)

How To Use Optimization: Model v.s. Method

Questions

1. How to model the problem by optimization?

   This is not just a math problem.

   It's about understanding the core aspects, extract the key elements, figure out the logic.

   This is the most challenging part, if you work for a company.
2. How to solve it?
   Focus of traditional optimization course.

It is hard to teach modeling, but we will provide examples to help a bit.

We will teach methods. Knowing methods is a great help for modeling.

How To Use Optimization

Regular Questions

1. **Choice** of formulation: Is my problem easy to solve? (if no, don't spend time on it! Choose another formulation)
2. **Choice** of algorithm: Which algorithm should I use?
3. **Efficiency**: How fast should I get my results?
4. **Efficacy**: How good is my result?

Each question may require a sub-area. Many of them unknown! But knowing the basics helps a lot.

Answer (partially): This course

Outline

Introduction: What is Optimization

Course Introduction

Examples in Machine Learning

Organization

Instructor:
Ruoyu Sun (`ruoyus at illinois.edu`)
Assistant professor, ISE, ECE (affiliated) and CSL (affiliated)
209D Transportation Building

Administrative Details:
Lecture time/location: Tu and Th: 2:00pm - 3:20PM, TB112 & zoom
Office hour: Wed 8-9am (may change later), zoom, or by appointment
Zoom info: see course website (don't distribute)

Course info page of UIUC: `https://courses.illinois.edu/schedule/2022/spring/IE/510`

TA: Kangcheng Lin (`klin14@illinois.edu`)
Office hour: TBD
Location: TBD

Organization

Textbook

D. Bertsekas, *Nonlinear Programming*, Main Textbook

Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006, Important reference.

Luenberger and Y. Ye. *Linear and nonlinear programming*, Reference

Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course* Reference

Other relevant materials will be distributed and discussed throughout the course

Syllabus

1. **Course website**: https://wiki.illinois.edu/wiki/display/IE510SP22 (also found in my homepage $\rightarrow$ teaching $\rightarrow$ 2022 Spring IE510)
2. **Media Space**: contain pre-recorded lecture videos (search IE510 Spring 2022)
3. **Gradescope**: homework/exam submission. `https://www.gradescope.com`
   - Gradescope entry code for the course: **8633BN**
   - Instruction videos on how to use gradescope: `https://www.gradescope.com/get_started`
4. **Piazza**: for certain announcement, discussion of the course contents and logistics: sign up at piazza.com/illinois/spring2022/ie510

Components of the course

1. Numerical grade = homework (35 $\%$) + 1 mid-term exam (35 $\%$) + class project (30$\%$) + bonus points (up to 10$\%$)
2. Homework will be assigned regularly, some of them are mathematical, some of them requires programming
3. Course slides will be distributed on the course website regularly
4. One **mid-term exam** (around Mar 24, after Spring break)
5. Class project report
6. **No final exam**

Homework and Project Policy

- **Homework submission:** electronically via **gradescope** (not via email)
- You are given 3 "grace days" (self-granted extensions) for homeworks
- Instructor-granted extensions are only considered after all grace days are used and only given in exceptional situations.
- Late hw submission (after running out of grace days) leads to 20% penalty per day
- Hard deadline of 3 days past the original due date. Late submissions after the hard deadline (penalty or not) lead to ZERO point.
- Bonus points = bonus problems in homework/exam and/or excellent project
- Late project report submission is NOT accepted

More on the course

1. **Pre-requisite**: basic knowledge about linear algebra and calculus is required
2. Helpful knowledge: probability, numerical linear algebra, complexity theory, machine learning
3. **Theoretical or practical**? Mixture. Lots of theoretical analysis, but will try to provide insights/discussion/applications whenever possible

Project

1. There is a class project (30% of the numerical grade)
2. One-person, or multiple-people projects (indicate who did what)
3. **Time line**: 1-page proposal by (roughly) Mar 1;
   full report due on early May
4. **Option 1**: Apply optimization to practical problems
   Examples: recommendation system, object detection, beamforming design, reinforcement learning, GAN.
5. **Option 2**: study of optimization algorithms. It doesn't need to involve rigorous proofs, but also not pure applications.
   - How is AdaGrad compared to gradient descent?
   - When is cyclic coordinate descent slower than randomized coordinate descent?
   - When does Lagrangian multiplier method diverge?
6. **Option 3**: work on a theoretical question

Academic Misconduct

1. There is zero tolerance on academic misconduct. Individuals suspected of committing academic dishonesty will be reported to the university.
   - Penalty for academic misconduct (up to 100%).

2. Collaboration is encouraged, but not copying each others' homeworks

Outline

Introduction: What is Optimization

Course Introduction

Examples in Machine Learning

Key questions to keep in mind

- **Formulation**: How to set up an optimization problem?
  - What are the optimization variables here?
  - What is the optimization objective?
- **Analysis**: How to analyze the formulation? (optimal solutions, etc.)
- **Algorithm:** How to solve the resulting problem?

Toy Example

Problem: Solve equation $2x = 3$.

Optimization problem: $\min_{x\in\mathbb{R}}(2x-3)^2$.

Problem: Solve a system of equations $Aw = b$.

Optimization problem: $\min_{w\in\mathbb{R}^n} \|Aw - b\|^2$.

Principle: to make sth. equal 0, we can minimize a non-negative quantity

Problem 1: Linear Regression I

1. **Training data sets** (n data points, $\mathbf{a}_i$ is the feature, and b_i is the label)

$$\{\mathbf{a}_i, b_i\}_{i=1,\cdots,n}, \ \mathbf{a}_i \in \mathbb{R}^d, \ b_i \in \mathbb{R}$$

2. **Problem**: Learn a function f, such that $f(\mathbf{a}_i) \approx b_i, \forall i.$
3. Linear regression: let f be a linear function parameterized by $\mathbf{x} \in \mathbb{R}^d, x_0 \in \mathbb{R}$,

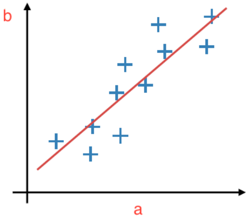
$$f(\mathbf{a}; \mathbf{x}) = \mathbf{x}^T \mathbf{a} + x_0 = \tilde{\mathbf{x}}^T \tilde{\mathbf{a}}, \quad \text{with} \quad \tilde{\mathbf{x}} := [\mathbf{x}, x_0], \tilde{\mathbf{a}} := [\mathbf{a}, 1]$$

4. We have absorbed x_0 in $\mathbf{x}$ and augmenting $\mathbf{a}_i$'s with extra 1
5. **Minimization problem:** minimize

$$F(\tilde{\mathbf{x}}) = \underbrace{\frac{1}{2}\sum_{i=1}^{n}(\tilde{\mathbf{x}}^T \tilde{\mathbf{a}}_i - b_i)^2}_{\text{squared loss}} = \|\mathbf{A}^T \tilde{\mathbf{x}} - \mathbf{b}\|^2$$

Problem 1: Linear Regression II

- 1d visualization: when $d = 1$, the data points are $\{a_i, b_i\}$, where $a_i, b_i \in \mathbb{R}$.



- A lot of aspects (including modeling and algorithm) can be improved when the data matrix $\mathbf{A}$ becomes "large"

Problem 1: Linear Regression III

- What if we also want to select a few key features that are most important?
- What if the dimension of the variable is huge ($\mathbf{x}$ very long, lots of features), while the data is scarce (n is small)?
- What if the data sets are distributed at different locations?

Problem 2: Classification I

- Example: Cat v.s. dog classification
  1. Given images of cat or dog
  2. Let the computer classify each image

Figure: Classification of cats and dogs

Problem 2: Classification II

- Data set $\{\mathbf{a}_i, b_i\}_{i=1}^n$, $\mathbf{a}_i \in \mathbb{R}^d$, $b_i \in \{-1,\ +1\}$
  - e.g. $b_i = +1$ represents “cat”; $b_i = -1$ represents “dog”
- Problem: We want to learn a f, such that $f(\mathbf{a}_i) \approx b_i$.
- **Optimization problem** (informal): find f such that dist$(f(\mathbf{a}_i), b_i)$ is minimized for certain “distance”
- **Optimization variable** and feasible set: $f(\mathbf{a}_i) = \mathbf{w}^T\mathbf{a}_i, \mathbf{w} \in \mathbb{R}$.
  - Often, there is a bias term $f(\mathbf{a}_i) = \mathbf{w}^T\mathbf{a}_i + w_0$; for simplicity, we skip w_0 in this lecture
  - f can take more general form; here, only consider linear function

Problem 2: Classification III

- **Optimization problem**: Find $\mathbf{w}$ s.t. dist$(\mathbf{w}^T\mathbf{a}_i, b_i), \forall i$ is small.
- First **objective function**: 0-1 loss function

$$\min_{\mathbf{w}} \sum_i \text{dist}(\mathbf{w}^T\mathbf{a}_i, b_i) = \sum_{i=1}^n \not\Vdash(b_i\mathbf{w}^T\mathbf{a}_i > 0),$$

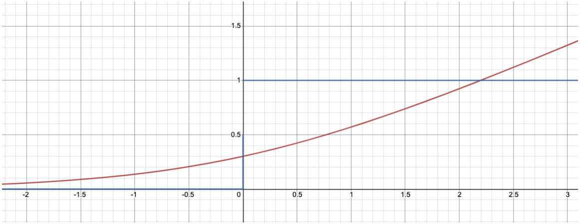
where $\not\Vdash(t > 0) = \begin{cases} 0 & t > 0 \\ 1 & t \le 0. \end{cases}$.

- Second **objective function** (binary cross entropy, or logit): dist$(y, \hat{y}) = \log(1 + \exp(-y\hat{y}))$

$$\min_{\mathbf{w}} \sum_{i=1}^n \text{dist}(\mathbf{w}^T\mathbf{a}_i, b_i) = \sum_{i=1}^n \log(1 + \exp(-b_i\mathbf{w}^T\mathbf{a}_i)).$$

Comparison of Two Loss Functions

Ideal $0-1$ loss function (as a function of $y\hat{y}$)
v.s.
surrogate loss function $\log(1+\exp(-z))$



Problem 3: Neural Nets I

Finding nonlinear classification boundary?

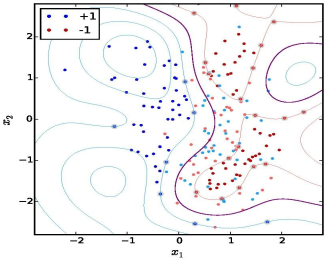


Figure: Nonlinear Classification [Wikipedia]

Neural Networks, especially Deep Neural Networks (DNN) become increasingly popular for various machine learning tasks

Problem 3: Neural Nets II

Data set $\{\mathbf{a}_i, b_i\}_{i=1}^n$, $\mathbf{a}_i \in \mathbb{R}^d$, $b_i \in \mathcal{Y}$.

Optimization problem (informal): find f such that dist$(f(\mathbf{a}_i), b_i)$ is minimized for certain "distance"

Define a neural-net function
$f(\theta; x) = W^L \phi(W^{L-1}\phi(W^{L-2} \cdots \phi(W^1 x) \dots))$, where $\phi : \mathbb{R} \to \mathbb{R}$ is a certain nonlinear function.

Optimization variable: $\theta = (W^L, \dots, W^1)$.

Optimization problem: $\min_\theta \sum_{i=1}^n \ell(f(\theta; \mathbf{a}_i), b_i)$, where ℓ is a certain loss function.

- Quadratic loss: $\ell(y, \hat{y}) = (y - \hat{y})^2$
- BCE loss: $\ell(y, \hat{y}) = \log(1 + \exp(-y\hat{y}))$

Lessons

Why do we discuss the formulations above?

First, formulation is important in practice.

- Follow the pipeline: set up “optimizing sth”; identify opt variable; identify objective.

Second, keep these examples in mind.

- Examples inspire questions, and inspire theory to address the questions
- Apply theory/algorithm to examples
- **General advice**: When learning abstract things (theory, algorithms, etc.), it is greatly helpful to keep some examples

Summary

History of optimization.

- Scattered results 16th-19th century
- Fast development since 1950s
- Received huge interest after 2010s (largely due to machine learning and big data)

Logistics of the course.

- **Course website**: `https://wiki.illinois.edu/wiki/display/IE510SP22`

Prototype examples of optimization.

- Linear regression
- Logistic regression
- Neural nets for regression and classification