**10-425/625: Introduction to Convex Optimization** **(Fall 2023)**

# Lecture 1: Overview of Optimization

*Instructor:*[1] *Matt Gormley* *August 28, 2023*

## 1.1 Syllabus Highlights

The syllabus is required reading, but here are some of the highlights:

- 1 Exam: in-class, Wed, Nov-08 (20%)
- 6 Quizzes: in-class, lowest score dropped (10%)
- 5-6 Homeworks: five for 425, six for 625 (40%)
  - 6 grace days for homework assignments
  - Late submissions: 75% day 1, 50% day 2, 25% day 3
  - No submissions accepted after 3 days w/o extension
  - Extension requests: for emergencies, see syllabus
- 1 Project: teams of 1-2, apply opt. to large scale ML problem (25%)
- Recitations: Fridays, same place and time as lecture
- Technologies: Piazza (discussion), Gradescope (homework), Google Forms (out-of-class surveys/polls)
- Office Hours: posted on Google Calendar on "Office Hours" page

## 1.2 What the course is about

The course is broadly about optimization. Despite the fact that it's a course in the ML department – this is not solely a course about optimization for

[1]These notes were originally written by Siva Balakrishnan for 10-725 Spring 2023 (original version: here) and were edited and adapted for 10-425/625.

deep learning. At least the first half of the course will primarily focus on *convex* optimization. These ideas are extremely broadly useful.

There are lots of different motivations for the topics we will learn about in this course. Optimization problems are everywhere in ML, Statistics, and tons of other disciplines.

1. A deep understanding of optimization will aid in designing algorithms to solve different types of optimization problems, and in understanding their relative merits. This will be our primary focus in this course.

2. Just formulating an optimization problem often gives a much deeper understanding of the problem at hand – for instance, the statistical analysis of most estimators crucially builds on insights (and characterizations) obtained by formulating the estimator as a solution to an optimization problem.

3. Finally, knowing the tricks of the optimization trade often aids in creating new optimization problems (ones with better algorithmic properties – i.e. are easier to solve, or better statistical properties).

The second half of the course will focus on advanced techniques for optimization that are the workhorses of modern machine learning. This will include both advanced methods of convex optimization, as well as techniques that can help the types of nonconvex optimization problems that abound in deep learning. Further, we will explore techniques used for (efficient) distributed optimization.

Today's lecture will focus on introducing optimization problems, and convex optimization problems. This is all from Chapter 1 of the Boyd-Vandenberghe (henceforth BV) book.

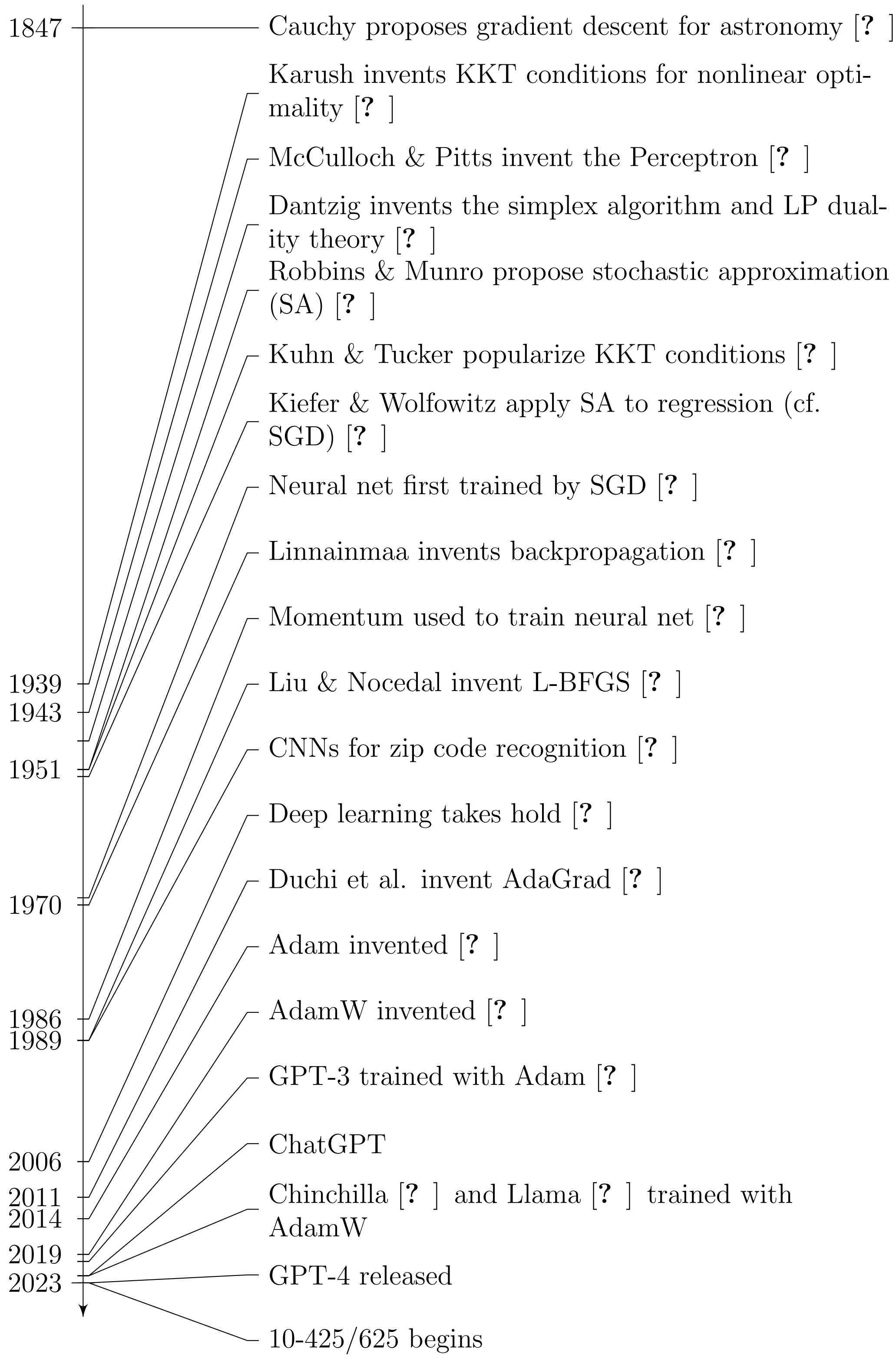
## 1.3 Timeline of Optimization and Machine Learning

> Just about the only aspect of my work in machine translation from five years ago that is still relevant today is the survey I wrote on *optimization* for MT—plus some datasets and metrics.
>
> Graham Neubig, associate professor, LTI, CMU

This quote[2] captures an important shift that occurred in machine learning over the last half decade: all the models and many of the associated inference and learning algorithms we used have since gone by the wayside. However, convex optimization remains as the bedrock for training even the largest, most complex models we deal with today.

Below, we consider a (woefully incomplete and a little bit arbitrary) timeline of some important events in optimization and machine learning.

[2]With apologies to Graham for what is surely an egregious paraphrase of what he actually said.


1847
Cauchy proposes gradient descent for astronomy [? ]
Karush invents KKT conditions for nonlinear optimality [? ]
McCulloch & Pitts invent the Perceptron [? ]
Dantzig invents the simplex algorithm and LP duality theory [? ]
Robbins & Munro propose stochastic approximation (SA) [? ]
Kuhn & Tucker popularize KKT conditions [? ]
Kiefer & Wolfowitz apply SA to regression (cf. SGD) [? ]
Neural net first trained by SGD [? ]
Linnainmaa invents backpropagation [? ]
Momentum used to train neural net [? ]
Liu & Nocedal invent L-BFGS [? ]
CNNs for zip code recognition [? ]
Deep learning takes hold [? ]
Duchi et al. invent AdaGrad [? ]
Adam invented [? ]
AdamW invented [? ]
GPT-3 trained with Adam [? ]
ChatGPT
Chinchilla [? ] and Llama [? ] trained with AdamW
GPT-4 released
10-425/625 begins
1939
1943
1951
1970
1986
1989
2006
2011
2014
2019
2023

## 1.4 (Mathematical) optimization problems

An optimization problem of the form,

$$\begin{aligned} \min\ & f_0(x) \\ \text{subject to } & f_i(x) \leq b_i, \quad i \in \{1, \ldots, m\}. \end{aligned}$$

Just some terminology:

1. **Optimization variables:** $x \in \mathbb{R}^d$.
2. **Objective function:** $f_0 : \mathbb{R}^d \mapsto \mathbb{R}$.
3. **Constraint functions:** $f_i : \mathbb{R}^d \mapsto \mathbb{R}$.
4. **Feasible solution:** $x$ satisfies all the constraints.
5. **Optimal solution:** $x^*$, has smallest value of $f_0$ amongst all vectors which satisfy constraints.
6. **Optimal value:** $p^* = \inf\{f_0(x) : f_i(x) \leq 0, i \in \{1, \ldots, m\}\}$.
   - $p^*$ may not be attained, i.e. there may not be an $x^*$ for which $f_0(x^*) = p^*$.
   - $p^* = \infty$ if problem is infeasible.
   - $p^* = -\infty$ if problem is unbounded from below.

**Background:** The infimum (i.e., inf) of a set is useful to us because it may exist even if the minimum (i.e., min) does not. The inf of a function $f(x) \in \mathbb{R}$ is the greatest value of $x \in \mathbb{R}$ that lower bounds $f$.

**Example 1.1.** Consider the function $f(x) = 1/x$. What is $\inf_{x \in \mathbb{R}: x > 0} f(x)$? What is $\min_{x \in \mathbb{R}: x > 0} f(x)$?

### 1.4.1 Examples

It is worth keeping in mind some examples of optimization problems, just so we have some concrete places to map the terminology we will learn. Here are some of my favorite optimization problems:

1. Maximum likelihood
2. Least squares
3. Empirical risk minimization
4. Optimal Transport

**Least Squares** Let's consider the Least Squares problem in more detail: Suppose we have a full rank matrix $A \in \mathbb{R}^{m \times n}$, and a vector $b \in \mathbb{R}^m$ such that $b$ is not a linear combination of the rows of $A$. We can't find $x \in \mathbb{R}^n$ such that $Ax = b$, so instead we want:

$$\min_x ||Ax - b||_2^2$$

We can expand out the objective to better inspect it using the fact that for any vector $v$, $||v||_2^2 = v^T v$.

$$\begin{aligned} ||Ax - b||_2^2 &= (Ax - b)^T (Ax - b) \\ &= (x^T A^T Ax) - 2b^T Ax + b^T b \end{aligned}$$

The $x$ that minimizes this objective is $x^* = (A^T A)^{-1} A^T b$.

### 1.4.2 Standard form

It is not significantly different, but some authors (particularly BV), refer to programs in standard form as also additionally allowing equality constraints, i.e.

$$\begin{aligned} \min \ & f_0(x) \\ \text{subject to } & f_i(x) \leq b_i, \quad i \in \{1, \ldots, m\} \\ & h_i(x) = 0, \quad i \in \{1, \ldots, p\}. \end{aligned}$$

### 1.4.3 Implicit versus explicit constraints

The above optimization problems have some explicit (inequality and equality) constraints. It is worth noting that in general they also have *implicit constraints*, i.e. that,

$$x \in \mathcal{D} = \text{dom}(f_0) \cap \bigcap_{i=1}^{m} \text{dom}(f_i) \cap \bigcap_{i=1}^{p} \text{dom}(h_i).$$

That is to say, these functions may not be defined everywhere, in which case our optimization problem is implicitly only over vectors where all the criterion and constraint functions are defined.

If we wanted to be more explicit we might write the standard form optimization problem as:

$$\begin{aligned} \min_{x \in \mathcal{D}} \ & f_0(x) \\ \text{subject to } & f_i(x) \le b_i, \quad i \in \{1, \ldots, m\} \\ & h_i(x) = 0, \quad i \in \{1, \ldots, p\}. \end{aligned}$$

### 1.4.4 Convex Optimization Problems – Standard Form

**Background:** In real vector space, a **linear function** can be written as $f(x) = Ax$ for $A \in \mathbb{R}^{m \times n}$ and $x \in \mathbb{R}^n$. An **affine function** includes a translation of a linear function and so can be written as $g(x) = Ax + b$ for $b \in \mathbb{R}^m$.

A problem of the form,

$$\begin{aligned} \min_{x \in \mathcal{D}} \ & f_0(x) \\ \text{subject to } & f_i(x) \le b_i, \quad i \in \{1, \ldots, m\} \\ & h_i(x) = 0, \quad i \in \{1, \ldots, p\}, \end{aligned}$$

where

1. $\mathcal{D}$ is a convex set.
2. $f_0, f_1, \ldots, f_m$ are convex functions.
3. $h_i(x) = a_i^T x + b_i$, are affine functions.

To make sense of this definition we'll need to understand what convex sets are, and what convex functions are. This will be what we will spend most of this and the next lecture on.

For now it is worth noting (and re-visiting once the definitions are in place), that the explicit constraints define a convex set, and their intersection with the domain $\mathcal{D}$ is also a convex set. If we denote this convex set $\mathcal{C}$ then our convex optimization problem can be equivalently, succinctly described as:

$$\min_{x \in \mathcal{C}} f_0(x),$$

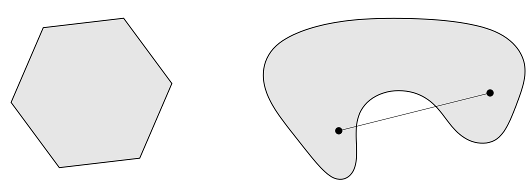
i.e. a *convex optimization problem* is simply the problem of minimizing a *convex function* over a *convex set.*

### 1.4.5 The Key Feature of Convex Optimization Problems

The most important structural feature of convex optimization problems is that *every local minima is a global minima.* This in turn makes local search algorithms effective for convex optimization.

We'll need to define some things in order to make sense of this claim. First, lets briefly define convex sets and functions:

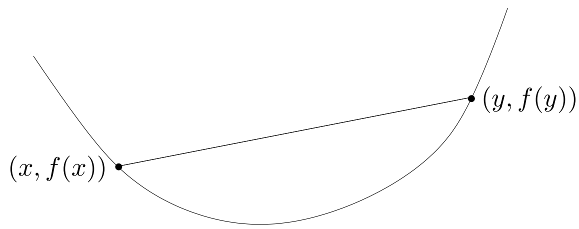
**Definition 1.2** **(Convex Set).** *A set $C$ is convex, if for every $x_1, x_2 \in C$ and $0 \leq \theta \leq 1$ we have that, $\theta x_1 + (1-\theta)x_2 \in C$.*



**Definition 1.3** **(Convex Function).** *A function $f : \mathbb{R}^d \mapsto \mathbb{R}$ is a convex function if,*

1. *dom$(f)$ is a convex set,*
2. *for every $x, y \in$ dom$(f)$, and $0 \leq \theta \leq 1$ we have that,*

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y).$$




**Definition 1.4** (**Nonconvex Function**)**.** *A function $f$ is said to be nonconvex if it is* not *convex.*

**Segue...** Next time we'll see that the key property of convex sets is this: for a convex optimization problem, a local optimum is also a global optimum.