Optimization Methods for Data Science

FILL OUT ACADEMIC ACTIVITY FORM

Course Webpage: Via Canvas

Grading: 50% HW
20% Midterm
30% Final

Rough course outline:

- Background
- Convexity / Convex opt.
- Gradient descent
- Variations: GD w/ momentum
  w/ acceleration

- Conjugate Gradient
- Newton's method

Nonconvex optimization

Why optimization for data science?

Let's start with an informal def'n of data science:

the science, or sometimes art, of extracting knowledge, from data.

Often, the goal is to make the best (optimal) decision based on, say, data.

Why this course?

Feynman posed question of one idea passed down if ever had cataclysm.

- In physics: "all things are made of particles ..."
- In ML/opt: "to learn a model, change the parameters in the direction that quickly reduces error"

Why not ChatGPT?

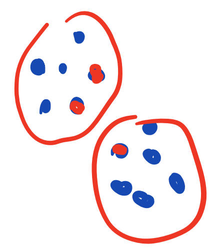
Policy: Cannot use (even edited version) for HW. Can use if you have question you may otherwise ask me/TA.

Can't blindly rely on LLMs:

- Context is critical (eg. do you need stability in min or argmin)
- No guide for when math errors occur
- If you can be imitated by LLM, you can be made redundant by LLM.
- LLMs are learned using techniques in course

General Examples of opt. problems

- optimize cost, revenue subject to some constraints
- partition data (cluster it) in some optimal way.



- classification: decide how to optimally assign objects to classes (e.g. sick vs healthy)

More Examples

- Building, say, fantasy sports teams given a salary cap.
- Recommender systems: making optimal recommendations (Netflix, Amazon)

- Image recognition
- Speech recognition
- Airline route planning.

- All deep learning applications
- Almost all machine learning has opt. at its core.

"Case Study" of how an optimization problem can arise out of a machine learning or data science application.

Classification (as an example of how an optimization problem can arise from a data-science or machine learning problem)

Somebody gives you accurately classified data:

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

$x_i \in \mathbb{R}^d$ e.g. $y_i \in \{-1, 1\}$

data point

label

so $(x_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}$

and $i = 1, \ldots, n$

Objective: Given $\{(x_i, y_i)\}_{i=1}^{n}$

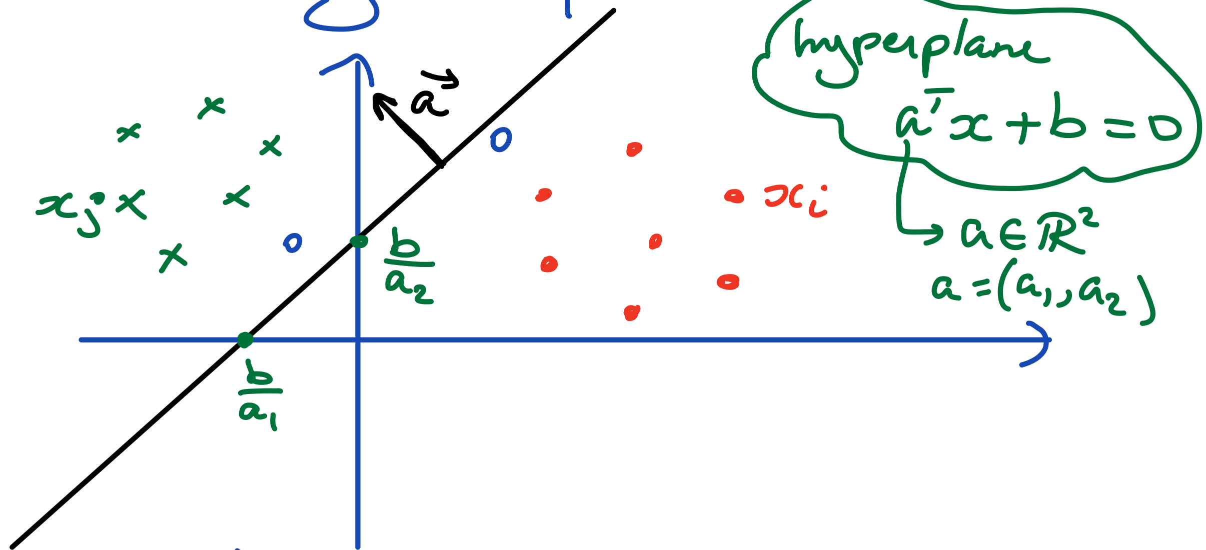
learn a function $f: \mathbb{R}^d \to \mathbb{R}$ so that when we get a new data point x, with true but unknown label y, we have that

$$f(x) > 0 \text{ when } y = +1$$

$$\& \ f(x) < 0 \text{ when } y = -1$$

There are many ways one might go about this. We'll look at one or two for now.

"Sketch of the problem"



One could seek the "best" hyperplane that separates the classes (to make the problem easier).

$$f(x) = a^T x + b$$

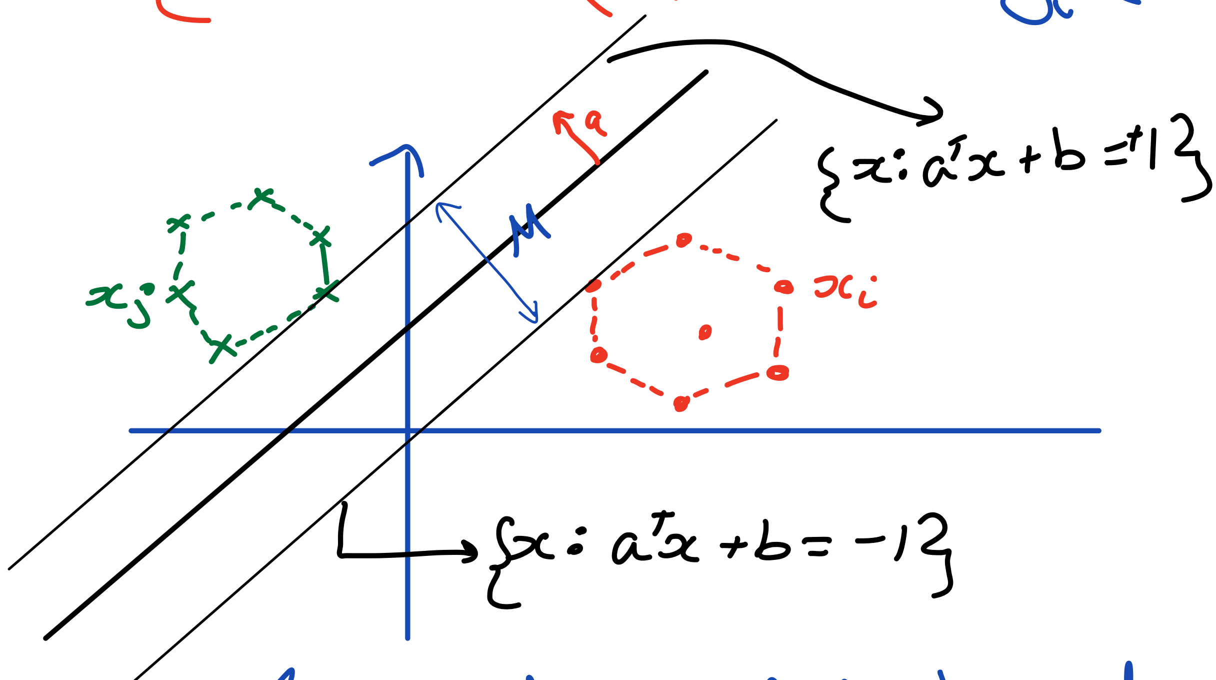
so we want to find $a \in \mathbb{R}^d$, $b \in \mathbb{R}$ so that

$$(*) \begin{cases} a^T \overset{\in \mathbb{R}^d}{x_i} + b > 0 & \text{when } y_i > 0 \\ a^T x_i + b < 0 & \text{when } y_i < 0 \end{cases}$$

$$\left(a^T z = \sum_{j=1}^{d} a_j z_j \right.$$

It turns out that it's more convenient to work with (**)

$$(**)\begin{cases} a^T x_i + b > +1 & \text{when } y_i > 0 \\ a^T x_i + b < -1 & \text{when } y_i < 0 \end{cases}$$



A reasonable goal is to make the distance between the "thin black hyperplanes", known as the margin, M, as big as possible

Fact: width of M is $\frac{2}{\|a\|}$

(why?)

Putting all this together:

we want to solve a constrained optimization problem

$$\max_{\substack{a \in \mathbb{R}^d \\ b \in \mathbb{R}}} \frac{2}{\|a\|} \quad \text{such that} \quad \begin{cases} \bullet\ a^T x_i + b_i > 1 \\ \quad \text{for all } i \\ \quad \text{for which } y_i = 1 \\ \bullet\ a^T x_i + b_i < -1 \\ \quad \text{for all } i \\ \quad \text{for which } y_i = -1 \end{cases}$$