Lecture 6 Machine Learning and Optimization

- Taylor Series review
- Oracle Methods of Optimization
- Convex sets & Functions; Convex Optimization Problems

Oracle Methods of Optimization

To solve a general optimization problem numerically, we use iterative methods:

$x_0 \leftarrow$ initial guess of the solution

repeat until "convergence":

- compute $f(x_i)$, and/or $\nabla f(x_i)$, and/or $\nabla^2 f(x_i)$

  zeroth, first, and second-order oracles

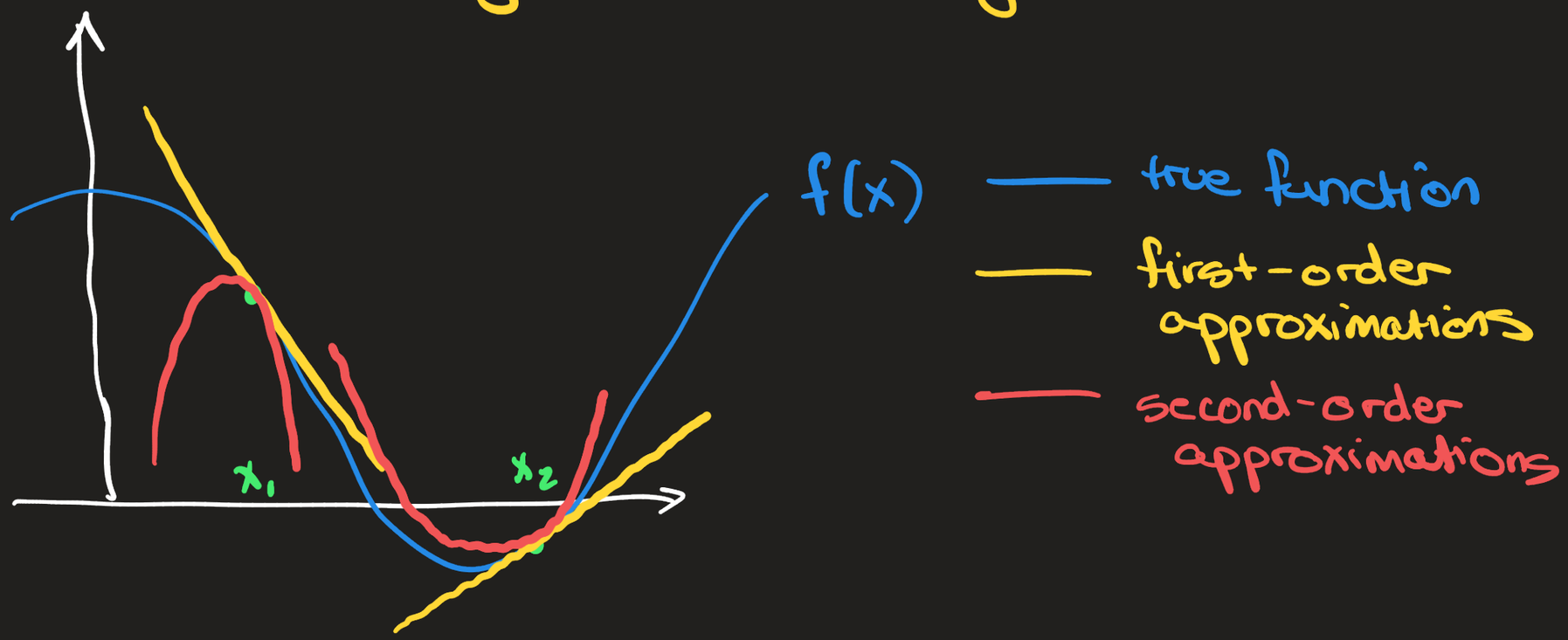
- form a local model
- move to a new point x_{i+1} based on the local model

We do not have a global view of the function, so model it locally, using multivariate Taylor Series

Taylor Series

A T.S. is a local polynomial approximation, taken at a point, of a differentiable function f.

Intuition: if I know a function's value at x, $f(x)$, its rate of change there, $f'(x)$, and its acceleration, $f''(x)$, then locally I can accurately model the function.



1st-order TS

Approximates f locally around x_1 with a linear polynomial:

$$f(x) \approx f(x_1) + f'(x_1)(x - x_1)$$

In fact, we are guaranteed the error of this approximation decreases faster than the rate at which $x \to x_1$:

$$\lim_{x \to x_1} \frac{|f(x) - f(x_1) - f'(x_1)(x - x_1)|}{|x - x_1|} = 0$$

equivalently,

$$f(x) = f(x_1) + f'(x_1)(x - x_1) + o(|x - x_1|)$$

2nd-order TS

Approximates f locally around x_1 with a quadratic polynomial

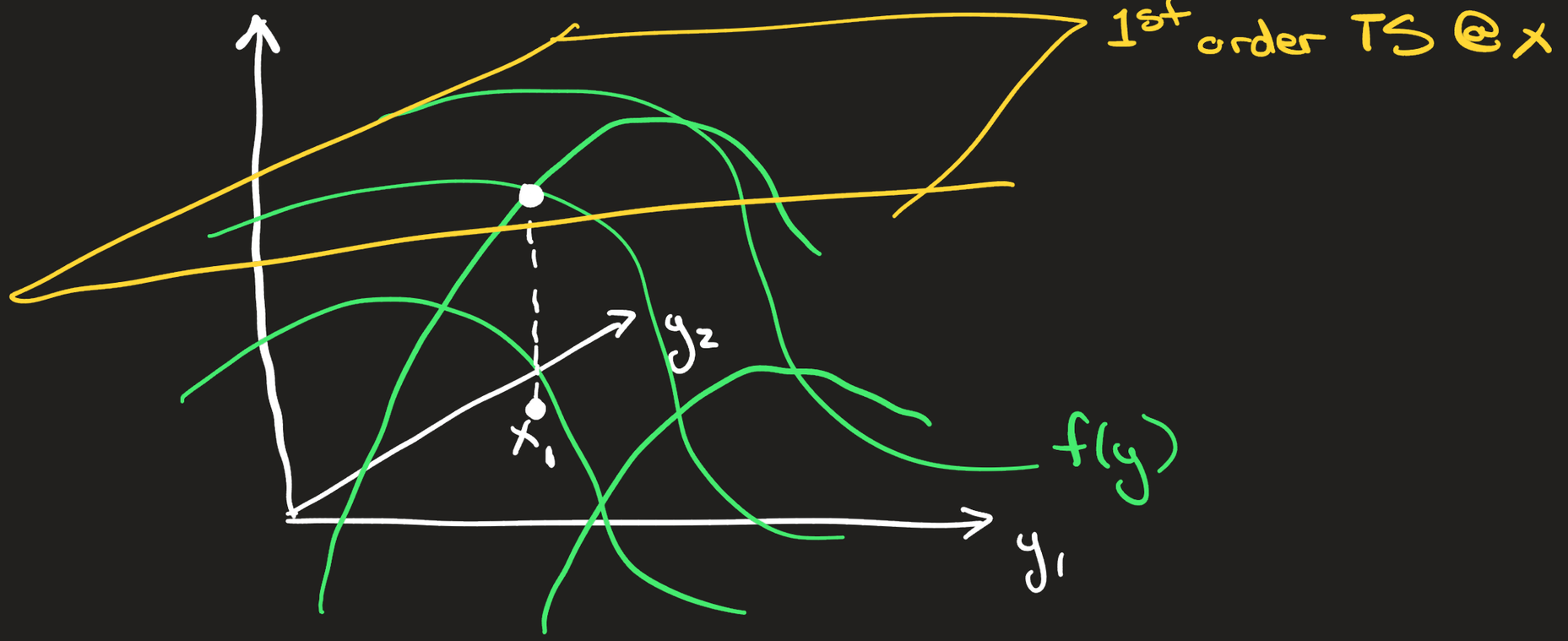
$$f(x) \approx f(x_1) + f'(x_1)(x-x_1) + \frac{f''(x_1)}{2}(x-x_1)^2$$

Because we have more information on how f behaves, the approximation error decreases superquadratically fast,

$$f(x) = f(x_1) + f'(x_1)(x-x_1) + \frac{f''(x_1)}{2}(x-x_1)^2 + o(|x-x_1|^2)$$

Multivariate TS

Same idea, but now a new point $y \in \mathbb{R}^d$ can differ from x in d different coordinates



1st-order TS

$$f(y) \approx f(x) + \sum_{i=1}^{d} \frac{\partial f}{\partial y_i}(x)(y_i - x_i)$$

$$= f(x) + \langle \nabla f(x), y - x \rangle$$

where $\nabla f(x) = \begin{bmatrix} \partial f/\partial y_1 (x) \\ \vdots \\ \partial f/\partial y_d (x) \end{bmatrix} \in \mathbb{R}^d$ is the gradient of f at x

The error is again superlinear in the distance of y from x:

$$f(y) = f(x) + \langle \nabla f(x), y - x \rangle + o(\|y - x\|_2)$$

2nd-order TS

Now the function can exhibit curvature in each pair of dimensions. This curvature is captured by the Hessian matrix of f at x.

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial^2 y_1}(x) & \cdots & \frac{\partial^2 f}{\partial y_1 \partial y_d}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial y_d \partial y_1}(x) & \cdots & \frac{\partial^2 f}{(\partial y_d)^2}(x) \end{bmatrix} = \left[\frac{\partial^2 f}{\partial y_i \partial y_j}(x) \right]_{i,j=1..d}$$

Note:

- $\nabla^2 f(x) \in \mathbb{R}^{d \times d}$
- $\nabla^2 f(x)$ is symmetric

Ex

Consider the negative log likelihood of a logistic regression model $\theta \in \mathbb{R}^d$ assigned by a single point (x, y)

$$\operatorname*{argmin}_\theta \log\left(1 + e^{-y\theta^T x}\right) = \ell(\theta)$$

$$\nabla \ell(\theta) = \begin{bmatrix} \partial \ell / \partial \theta_1 (\theta) \\ \vdots \\ \partial \ell / \partial \theta_d (\theta) \end{bmatrix} = \begin{bmatrix} \dfrac{-y x_1 e^{-y\theta^T x}}{1 + e^{-y\theta^T x}} \\ \vdots \\ \dfrac{-y x_d e^{-y\theta^T x}}{1 + e^{-y\theta^T x}} \end{bmatrix}$$

$$= \frac{-y e^{-y\theta^T x}}{1 + e^{-y\theta^T x}} \cdot x$$

Recall that $\nabla \ell(\theta) = \frac{y e^{-y\theta^T x}}{1 + e^{-y\theta^T x}} \cdot x$

$\nabla^2 \ell(\theta) \in \mathbb{R}^{d \times d}$ satisfies

$$\left[\nabla^2 \ell(\theta)\right]_{ij} = \frac{\partial^2 \ell}{\partial \theta_i \partial \theta_j}(\theta) = \frac{\partial}{\partial \theta_i}\left[\frac{\partial \ell}{\partial \theta_j}(\theta)\right]$$

$$= \frac{\partial}{\partial \theta_i}\left[\left[\nabla \ell(\theta)\right]_j\right]$$

$$= \frac{\partial}{\partial \theta_i}\left[\frac{y e^{-y\theta^T x}}{1 + e^{-y\theta^T x}} \cdot x_j\right] = ?$$ (exercise)

Recall for the Bernoulli noise model (logistic regression)

$$p(y=1|x) = \sigma(\theta^T x)$$
$$p(y=-1|x) = 1 - \sigma(\theta^T x) = \sigma(-\theta^T x)$$

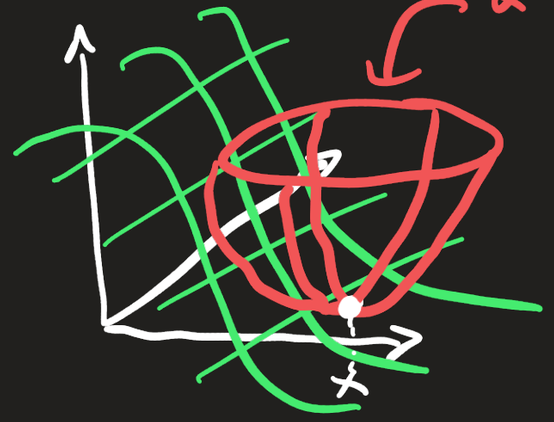
So

$$p(y|x) = \sigma(y\theta^T x) = \frac{1}{1+e^{-y\theta^T x}}$$

$$\Rightarrow \quad 1 - p(y|x) = \frac{e^{-y\theta^T x}}{1+e^{-y\theta^T x}}$$

$$\Rightarrow \quad \nabla \ell(\theta) = (1 - p_\theta(y|x)) \cdot yx$$

If $y|x$ is likely, small gradient, otherwise large gradient



Aside: If $v \in \mathbb{R}^d$, $w \in \mathbb{R}^p$, and $M \in \mathbb{R}^{d \times p}$, then

$$v^T M w = \sum_{i=1}^{d} v_i (Mw)_i$$

$$= \sum_{i=1}^{d} v_i \left(\sum_{j=1}^{p} M_{ij} w_j \right)$$

$$= \sum_{i=1}^{d} \sum_{j=1}^{p} M_{ij} v_i w_j$$

$$\boxed{v^T M w = \sum_{i=1}^{d} \sum_{j=1}^{p} M_{ij} v_i w_j}$$

if $v = w$ then $v^T M v = \sum_{i=1}^{d} \sum_{j=1}^{d} M_{ij} v_i v_j$

$$f(y) \approx f(x) + \langle \nabla f(x), y-x \rangle + \frac{1}{2} \sum_{i=1}^{d} \sum_{j=1}^{d} \frac{\partial^2 f(x)}{\partial y_i \partial y_j} (y_i - x_i)(y_j - x_j)$$

accounts for curvature in every pair of directions

$$= f(x) + \langle \nabla f(x), y-x \rangle + \frac{1}{2}(y-x)^T \nabla^2 f(x)(y-x)$$

and in fact the error decreases superquadratically:

$$f(y) = f(x) + \langle \nabla f(x), y-x \rangle + \frac{1}{2}\langle y-x, \nabla^2 f(x)(y-x) \rangle + o(\|y-x\|_2^2)$$

Because the multivariate TS have strong local approximation guarantees, we can use them to design optimization algorithms in the oracle model:

- Zeroth-order methods: use $x_0, \dots, x_i$ and $f(x_0), \dots, f(x_i)$ to determine x_{i+1}

- First-order methods use $x_0, \dots, x_i, f(x_0), \dots, f(x_i)$, and $\nabla f(x_0), \dots, \nabla f(x_i)$ to determine x_{i+1}

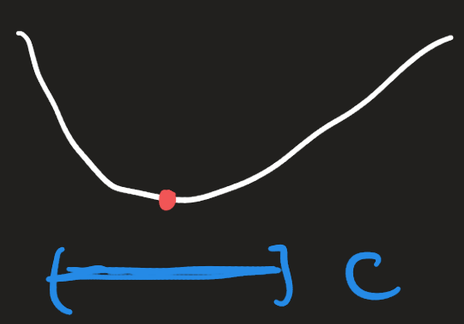
- Second-order methods use $x_0, \dots, x_i, f(x_0), \dots, f(x_i)$, $\nabla f(x_0), \dots, \nabla f(x_i), \nabla^2 f(x_0), \dots, \nabla^2 f(x_i)$ to determine x_{i+1}

Convex Optimization

In general, optimization is NP-Hard so we restrict ourselves to convex optimization problems

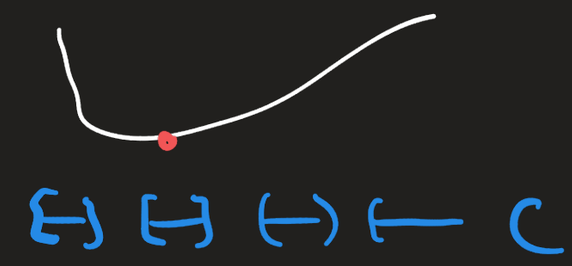
$$x^* = \underset{x \in C}{\operatorname{argmin}} f(x) \quad \text{where} \quad \begin{array}{l} C - \text{convex set} \\ f - \text{convex function} \end{array}$$

convex optimization

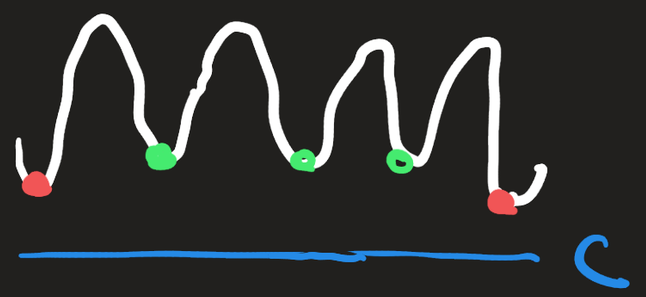


Local minima of convex functions are global minima.

nonconvex optimization



C nonconvex requires searching over many sets



. local minima are not necessarily global minima

Why convex optimization?

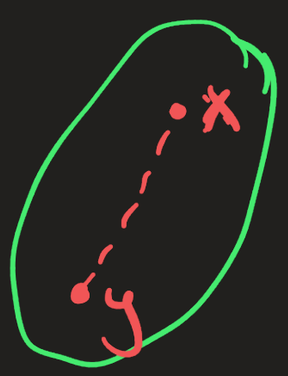
- very expressive model class. Contains all the MLE problems for exponential family statistical models; in particular: linear and logistic regression, multinomial logistic regression, and Poisson regression
- they are generally tractable: we can efficiently find local minima using algorithms not highly tailored to the specific optimization problem.
- local minima are global minima! so reasonable iterative algorithms are guaranteed to find a global minima
- we can say a lot about solutions to a problem using Convexity

What is convexity?

A set C is convex iff whenever $x, y \in C$, the line segment

$$[x, y] = \{\alpha \in [0,1] : \alpha x + (1-\alpha) y\}$$

is entirely contained in C



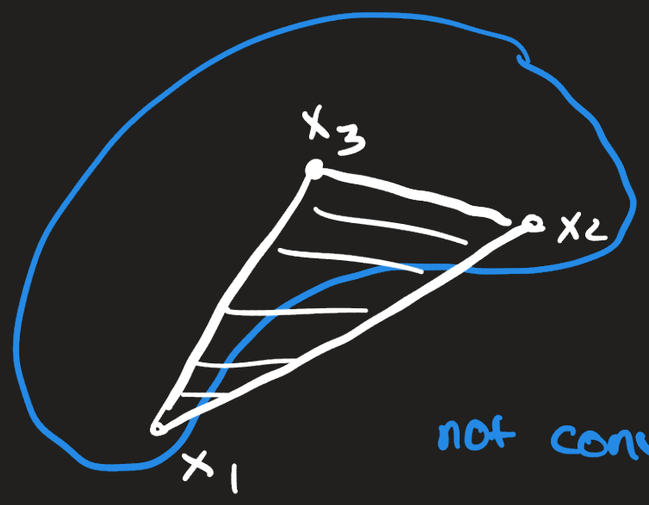
convex

nonconvex

Fact A set C is convex iff for all $x_1, \dots, x_k \in C$ and any numbers $\theta_1, \dots, \theta_k$ s.t. $\theta_i \geq 0$ for all i and $\sum_{i=1}^{k} \theta_i = 1$, the convex comb

Ex

convex



not convex

Given $x_1, \dots, x_k$ the set of all convex combinations of these points is called their convex hull. This is a convex set.

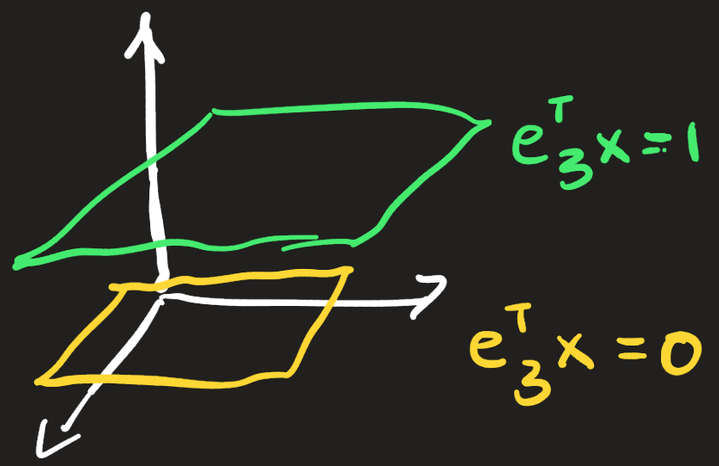
Fact

If C is a convex set and X is a r.v. taking values in C, then $\mathbb{E}X \in C$

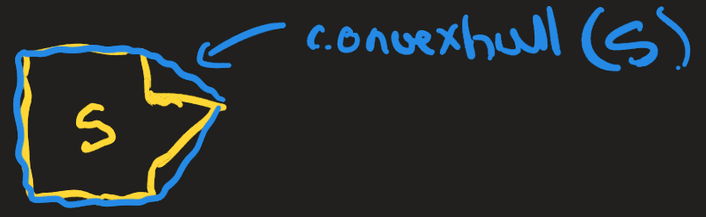
Examples of convex sets

- empty set $\emptyset$
- $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{R}^d$, rays, lines, line segments
- hyperplanes

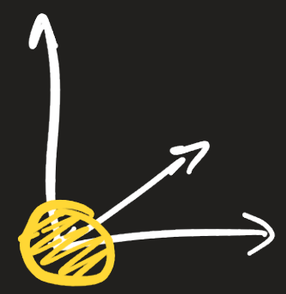
$$\{x : a^T x = b\}$$



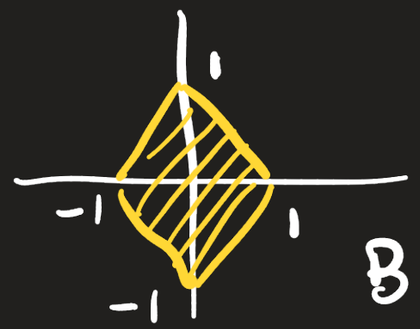
- convex hull of an arbitrary set S, i.e. the smallest convex set containing S.



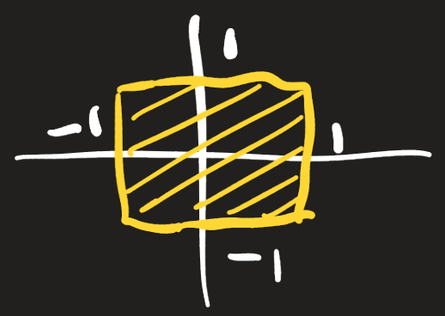
- The unit Euclidean ball $\{x \mid \|x\|_2 \leq 1\} = B_{\|\cdot\|_2}(1)$



- Norm balls $B_{\|\cdot\|}(x,r) = \{y \mid \|x-y\| \leq r\}$



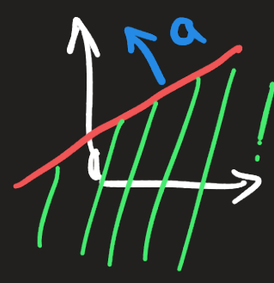
$B_{\|\cdot\|_1}(0,1)$



$B_{\|\cdot\|_\infty}(0,1)$

- polyhedra: a polyhedron is the set of solutions to a finite number of linear equalities and inequalities

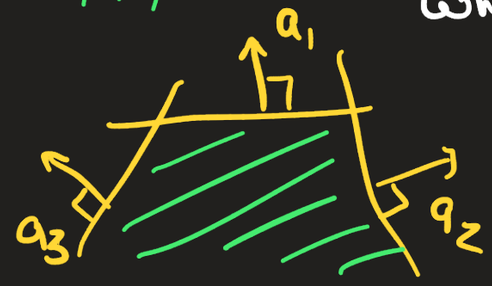
$\{x : a^T x \leq b\}$ is a half-space, so is convex



The intersection (a polyhedron)

$$\{x : a_1^T x \leq b_1, \cdots, a_n^T x \leq b_n\} := \{x : Ax \leq b\}$$

where $A = \begin{bmatrix} a_1^T \\ \vdots \\ a_n^T \end{bmatrix}$, $b = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$ is convex



$$\begin{bmatrix} a_1^T \\ a_2^T \\ a_3^T \end{bmatrix} x \leq \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

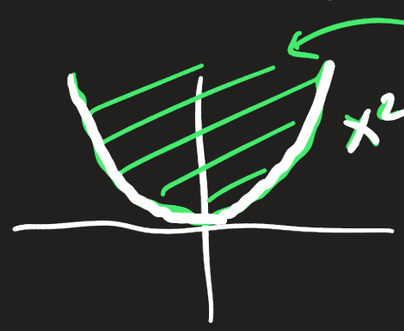
- Intersection of an arbitrary (even infinite) collection of convex sets is convex: if C_i is convex for all $i \in I$, then $\bigcap_{i \in I} C_i$ is convex

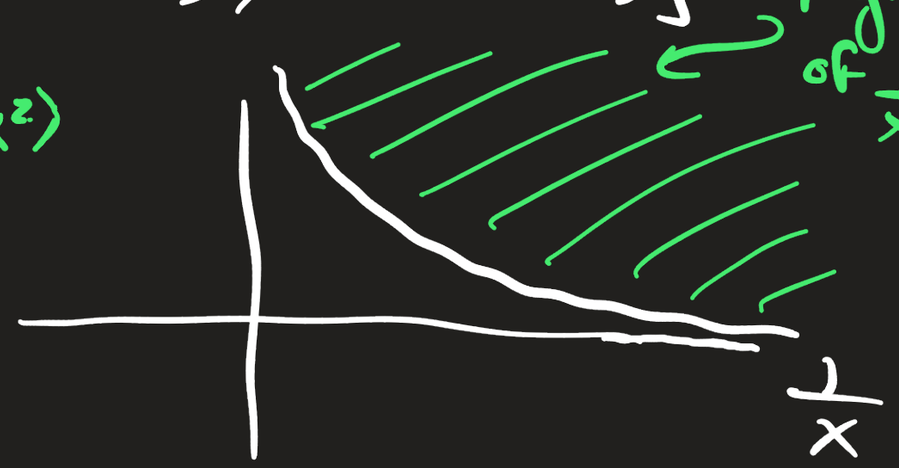
Convex function

A function f is convex on its domain, if:

1) $dom(f)$ is convex

2) $epigraph(f) = \{(x,t) : t \geq f(x), \; x \in dom(f)\}$ is convex



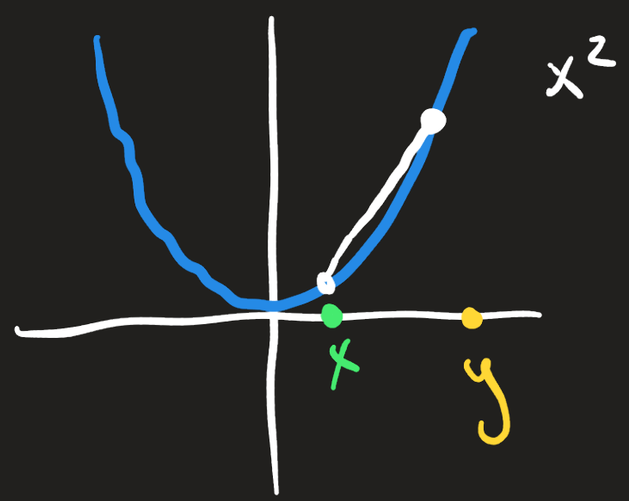


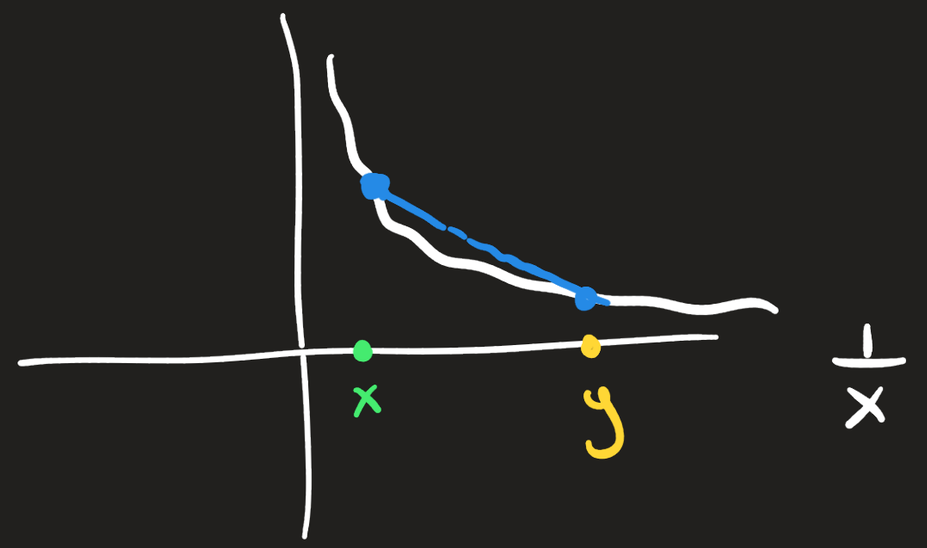
Equivalently, f is convex iff it satisfies

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y)$$

for all $\alpha \in [0,1]$ and $x, y \in dom(f)$

Visually, this means





f is convex iff it lies **below** all its secants/chords.

Note that this property is preserved if we take $f(x) = \infty$ whenever $x \notin \text{dom} f$. So, we often define convex functions $f: \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ with

$$f(x) = \begin{cases} \infty & \text{otherwise} \\ f(x) & \text{if } x \in \text{dom}(f) \end{cases}$$