Lecture 15
Convex Optimization

Lei Li, **Yu-Xiang Wang**

(some slides from my convex optimization class, originally taught by Ryan Tibshirani in CMU)

Announcements

- Modification to the schedule
  - Two lectures on statistical learning theory replaced by Reinforcement Learning.
  - Now three lectures on RL.
  - No more lectures on theory of deep learning (because it depends on statistical learning theory)

Plan today

- Review of what we have learned so far
- An optimization view to ML
  - Modeling with optimization
- Convex optimization basics
  - Convex Set
  - Convex functions
  - Examples

Review: We have learned a lot of concepts in ML from this course

- MLP
- Transformers
- VAE
- LSTM
- ConvNet
- Decision Trees
- Linear classifier
- Linear regression
- Logistic regression
- K-means
- Gaussian Mixture Models

Models

- PCA
- Probabilistic PCA
- CRF
- Linear dynamical systems
- Directed Graphical Model
- Undirected graphical models

models

learning

inference algs.

- Gradient descent
- Kalman filter
- Expectation Maximization
- Regularization
- Loss function
- Risk
- Empirical risk
- Sample complexity
- Iteration complexity
- Holdout
- Cross Validation

learning

eval

Review: machine learning basics

- Data $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$
- Hypothesis $h : \mathcal{X} \to \mathcal{Y}$ from $\mathcal{H}$
- Loss function $\ell(h, (x, y))$
- Learning algorithms: How to solve ERM or empirical risks minimization.

regularized

Review: Modeling --- formulate a problem to be solved by ML

- Feature engineering

  $\rightarrow \cancel{X}, Y$

- Discriminative modeling: specifying hypothesis class

  $\mathcal{H}$ { linear classifier

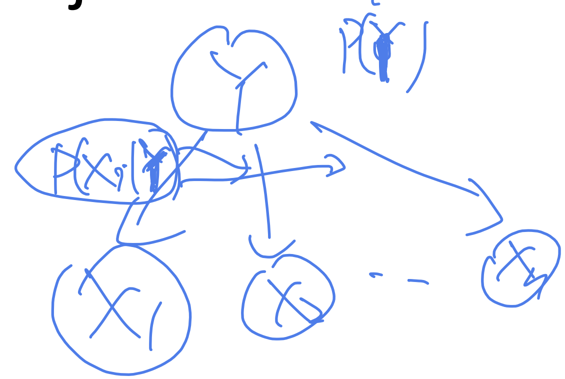
  Decision trees

  X

- Generative modeling: specifying the joint distribution

  $P_\theta(x, y)$

  Inference $\rightarrow$ $\boxed{P_\theta(y \mid x)}$



Quiz: Are these ML models discriminative or generative?

- MLP D
- Transformers D
- VAE G
- LSTM D/G
- ConvNet D
- Decision Trees D
- Linear classifier D
- Linear regression D
- Logistic regression D
- K-means D
- Gaussian Mixture Models G

- PCA D
- Probablistic PCA G
- CRF G
- Linear dynamical systems G
- Directed Graphical Model G
- Undirected graphical models G

Review: Discriminative vs Generative Modeling

	Discriminative / deterministic	Generative / Probabilistic
Modeling	Specify $\mathcal{H}$, loss	$P_\theta(x,y)$
Learning	ERM: $\hat{h} = \arg\min_{h \in \mathcal{H}} \sum_i loss(z_i, h)$	MLE $\hat{\theta} = \arg\max_\theta \sum_{i=1}^n \log P_\theta(x_i, y_i)$
Inference	$\hat{h}(x)$	$P_\theta(y \mid x)$

Supervised $\ell(y, f_\theta(x)) = CE(y, f_\theta(x))$

Does this unification work for unsupervised learning too?

$$\ell(z, h) = \min_{\mu \in h} \|z - h\|^2$$

$$h = \{\mu_1, \mu_2, \dots, \mu_k\}$$

Regularization vs Prior?

$$\frac{1}{n}\sum_i loss_i(h) + \lambda \|h\|_2^2$$

Gaussian Prior

$$\exp\left(\frac{-\|\theta\|^2}{2\sigma}\right)$$

$$\log\left(\left(\prod_{i=1}^n P_\theta(z_i)\right) \cdot \pi(\theta)\right)$$

$$= \sum_{i=1}^n \log P_\theta(z_i) + \log \pi(\theta)$$

$$\log \pi(\theta) = -\frac{1}{2\sigma}\|\theta\|^2$$

One way of another, we are dealing with optimization problems at the end of the day.

- What we learned so far is mostly about how we translate conceptual ideas into a rigorous optimization problem.

x^* optimal solution

$f^* := f(x^*)$ optimal obj val.

obj func.

translate  into $$P : \underline{\min_{x \in D}} \; f(x)$$

Conceptual idea

argument / variable

Domain

Optimization problem

Constraint set

- Two thoughts:
  1. How to solve these optimization problems?
  2. Why not model with optimization directly?

Why not directly use off-the-shelf optimization packages (e.g., cplex,gurobi, scipy.optimize)?

$$P \;:\; \min_{x \in D} \; f(x)$$

You need to know whether they are applicable.
You need to know whether they are guaranteed to find the solutions.
You need to know how quickly they find the solution, so as to set hyperparameters.

1. Different algorithms can perform better or worse for different problems P (sometimes drastically so)
2. Studying P through an optimization lens can actually give you a deeper understanding of the statistical procedure
3. Knowledge of optimization can actually help you create a new P that is even more interesting/useful

Advantages of modeling with optimization

- No need to deal with probabilities / MLE / conditional independences
- Directly optimize quantities of interest
- Encode structures /domain knowledge / design choices as part of the optimization problem
  - Design loss functions
  - Design regularization functions

$\min_{\theta} \; \text{loss}(\theta, y)$ s.t. $\sum_{(i,j)\in E} |\theta_i - \theta_j| \le K$

Example: Image denoising

$\lambda = 0$

$\theta_i = y_i$?

The 2d fused lasso or 2d total variation denoising problem:

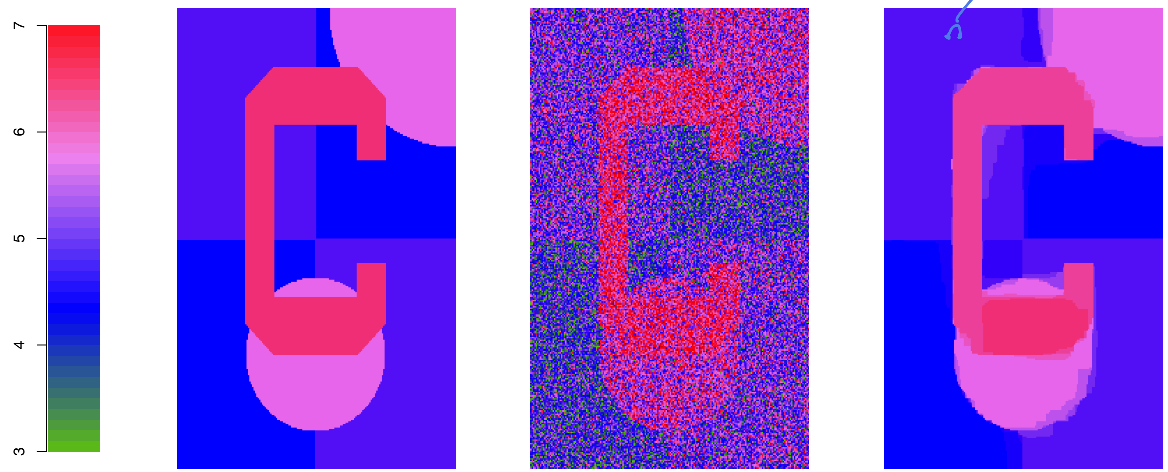
$\theta_i^* + \text{Noise}$

$$\min_{\theta} \; \underbrace{\frac{1}{2}\sum_{i=1}^{n}(y_i - \theta_i)^2}_{} + \underbrace{\lambda \sum_{(i,j)\in E} |\theta_i - \theta_j|}_{}$$

$\cancel{1(\theta_i \neq \theta_j)}$

This fits a piecewise constant function over an image, given data y_i, $i = 1, \ldots, n$ at pixels. Here $\lambda \geq 0$ is a tuning parameter

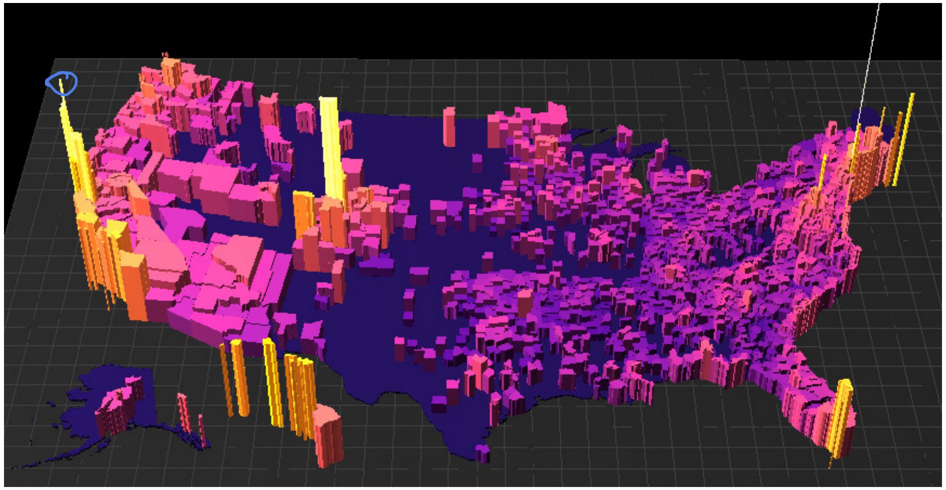
$\in E$



True image Data Solution

Example: Housing price prediction on a map

- Intuition:
  - Maybe neighbors on the map are likely to have similar housing prices?



https://www.visualcapitalist.com/interactive-map-price-per-square-foot-us-housing-markets/

(V, E) obs y_i $i \in \Omega \subset V$

$$\min_{\theta_i \, \forall i \in V} \underline{\sum_{i \in \Omega} (\theta_i - y_i)^2} + \lambda \underline{\sum_{(j,\ell) \in E} (\theta_j - \theta_\ell)^2}$$

Laplacian Smoothing

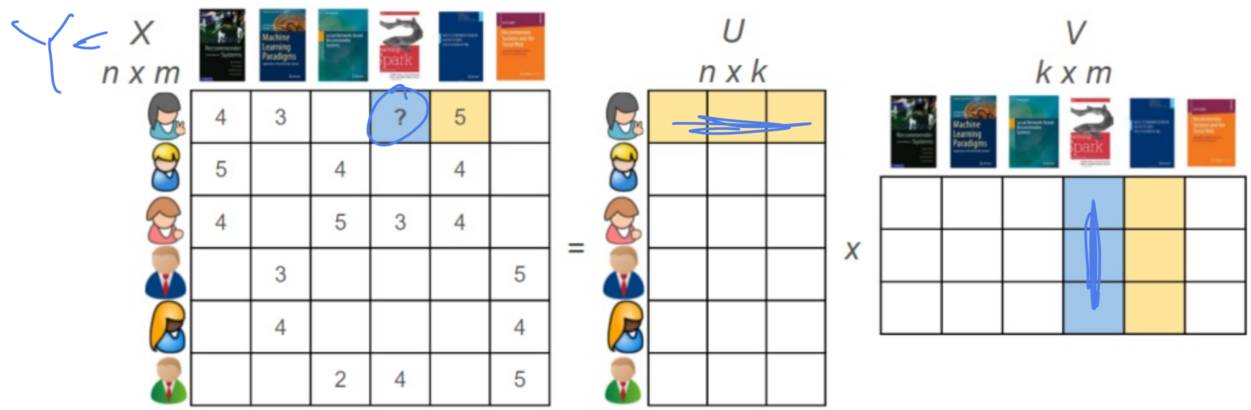
$$\sum_{i \in \Omega} |\theta_i - y_i|$$

$2 \to 1$

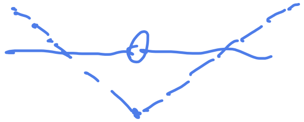
$\underline{|\theta_j - \theta_\ell|}$

Graph fused lasso

Example: Movie Recommendation



$$\min_{U,V} \sum_{(i,j)\in\Omega} (u_i^T v_j - y_{ij})^2 = \|P_\Omega(Y - UV^T)\|_F^2$$



$\|x\|_0 = \sum \mathbb{1}(x_i \neq 0)$

$\|x\|_1 = \sum |x_i|$

perfect

$Y = X + E$

Example: Robust PCA

$X \in \mathbb{R}^{d \times n}$

Corruption

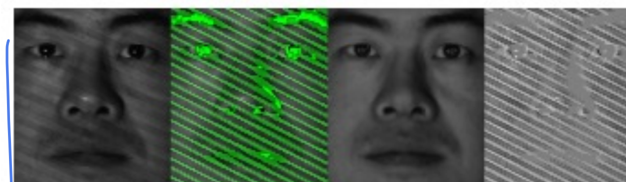
$Y = X + E + \text{Noise}$

lowrank Sparse

Column Sparse

$$\min_{X,E} \|Y - X - E\|_F^2 + \lambda_1 \operatorname{rank}(X) + \lambda_2 \|E\|_0$$

(a) Cast shadow and attached shadow are recovered. Region of cast shadow is now visible, and attached shadow is also filled with meaningful negative values.



(c) Rare corruptions in image acquisition are recovered.

$\operatorname{rank}(X) \to \cancel{\operatorname{trace}(X^T X)}$

$\|E\|_0 \to \|E\|_1$ $\|X\|_* = \sum \sigma_i(X)$

Frame

backgrnd foreground

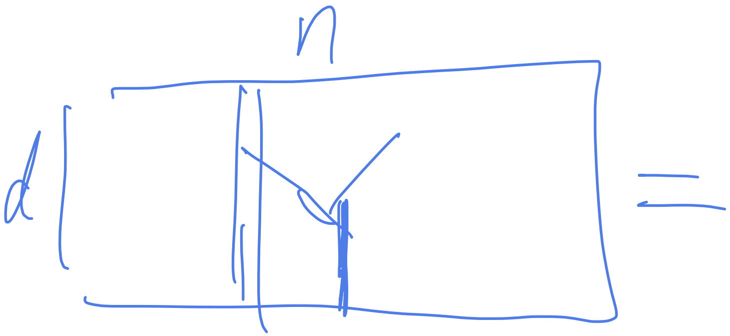
$Y = [\text{Video}] = X + E$

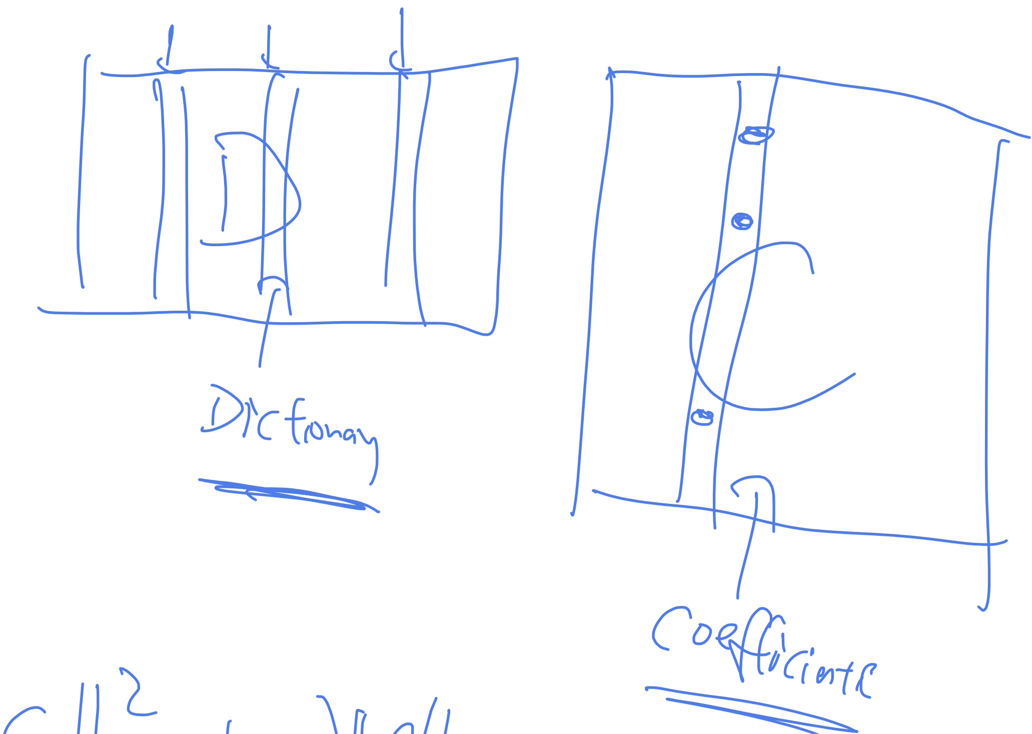
$t=1$

X E

Example: Dictionary Learning

K-SVD





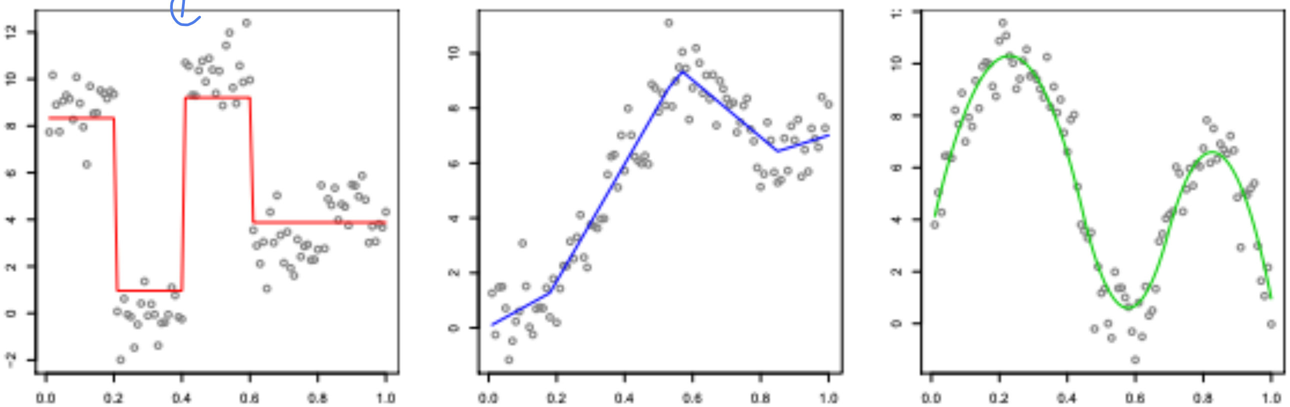
$$y_j = \sum_l D_{:l} C_{lj}$$

$$\min_{D,C} \|Y - DC\|_F^2 + \lambda \|C\|_{1,1}$$

s.t. D is orthonormal

Example: L1 Trend filtering

fused lasso

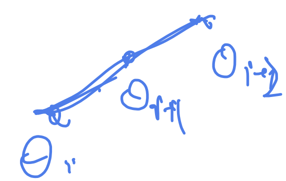


$\theta_i = f(x_i)$

θ_1 —— θ_d

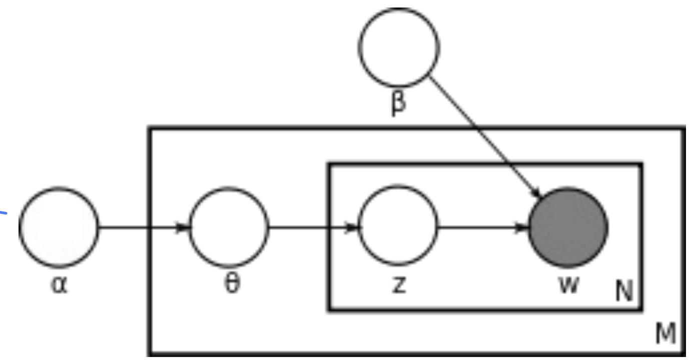
- How to design regularization terms that promote piecewise polynomial structures with a small number of knots?

$$\min_{\theta_{1:d}} \sum_{i\in[d]} (\theta_i - y_i)^2 + \lambda \sum_{i=1}^{d-2} \left| \theta_i - 2\theta_{i+1} + \theta_{i+2} \right|$$



Example: Topic models

- Latent Dirichlet Allocation



- From an optimization point-of-view

$$\underbrace{Y}_{d \times n} = \underbrace{U}_{d \times k}\ \underbrace{V}_{k \times n}$$

Document

topic

$U[:,i] \in \Delta^{d-1}$

How to solve these optimization problems?

- If **convex**, there are generic tools, and many algorithms with guarantees

- If not-convex:
  - Or we can try solving it anyways with greedy local search algorithms

    Greed is good: Algorithmic results for sparse approximation | 4129 | 2004
    JA Tropp
    IEEE Transactions on Information theory 50 (10), 2231-2242

  - There are often "convex relaxation"

    Just relax: Convex programming methods for identifying sparse signals in noise | 1692 * | 2006
    JA Tropp
    IEEE transactions on information theory 52 (3), 1030-1051

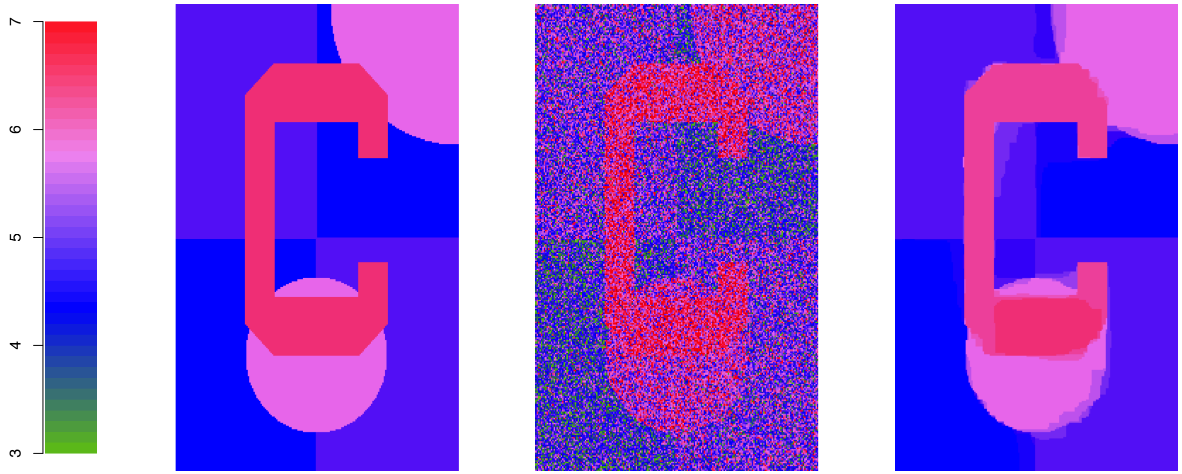
Revisit the example: What are some algorithms for solving it

Example: algorithms for the 2d fused lasso

The 2d fused lasso or 2d total variation denoising problem:

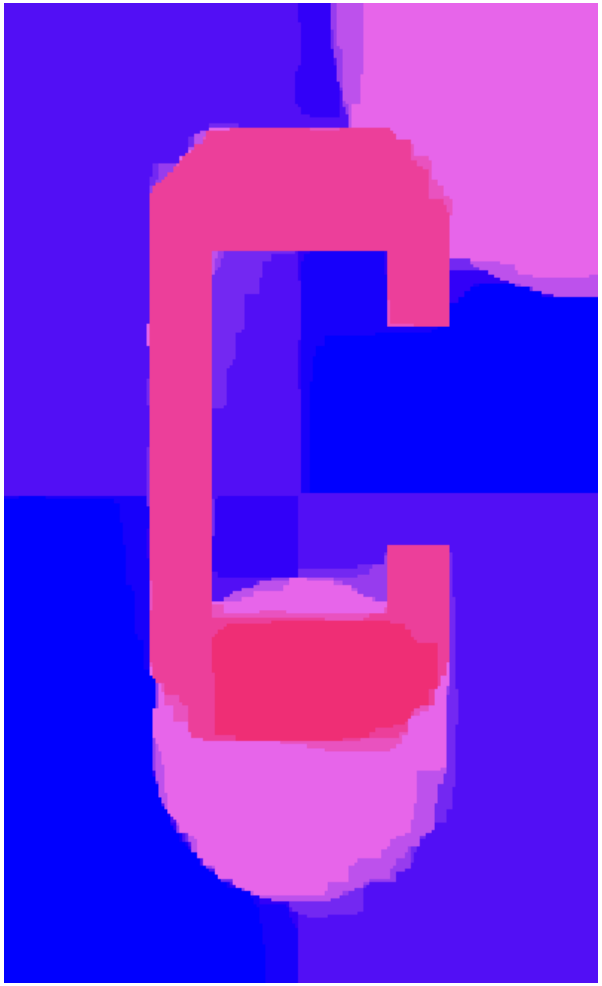
$$\min_{\theta} \ \frac{1}{2}\sum_{i=1}^{n}(y_i - \theta_i)^2 + \lambda \sum_{(i,j)\in E} |\theta_i - \theta_j|$$

This fits a piecewise constant function over an image, given data y_i, $i = 1, \ldots, n$ at pixels. Here $\lambda \geq 0$ is a tuning parameter



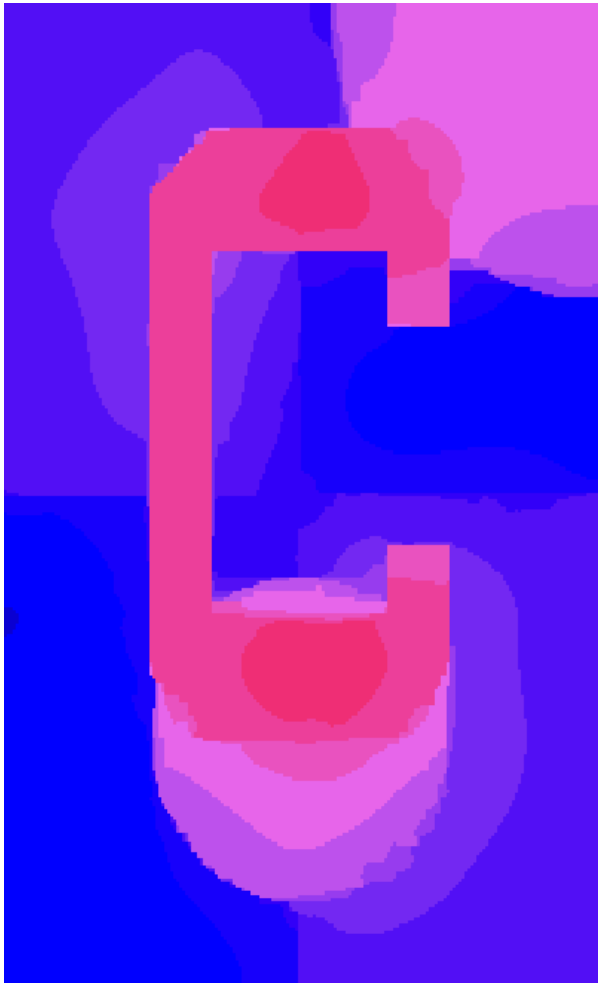
True image Data Solution

Our problem: $$\min_{\theta} \frac{1}{2} \sum_{i=1}^{n} (y_i - \theta_i)^2 + \lambda \sum_{(i,j) \in E} |\theta_i - \theta_j|$$



Specialized ADMM, 20 iterations

Our problem: $$\min_{\theta} \frac{1}{2}\sum_{i=1}^{n}(y_i - \theta_i)^2 + \lambda \sum_{(i,j)\in E} |\theta_i - \theta_j|$$

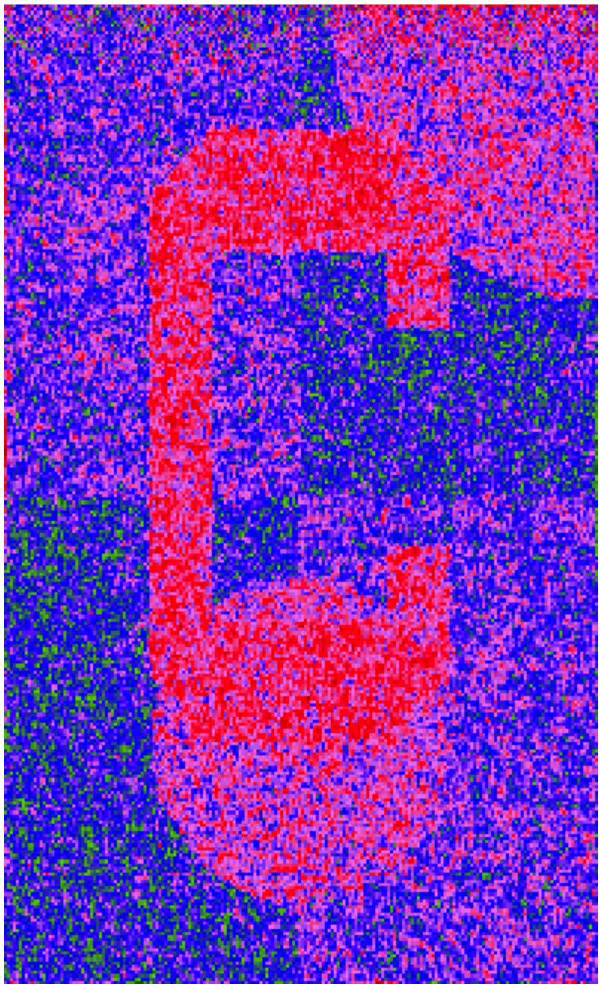


Specialized ADMM, 20 iterations

Proximal gradient descent, 1000 iterations

Our problem: $$\min_{\theta} \; \frac{1}{2} \sum_{i=1}^{n} (y_i - \theta_i)^2 + \lambda \sum_{(i,j) \in E} |\theta_i - \theta_j|$$

$x = i$

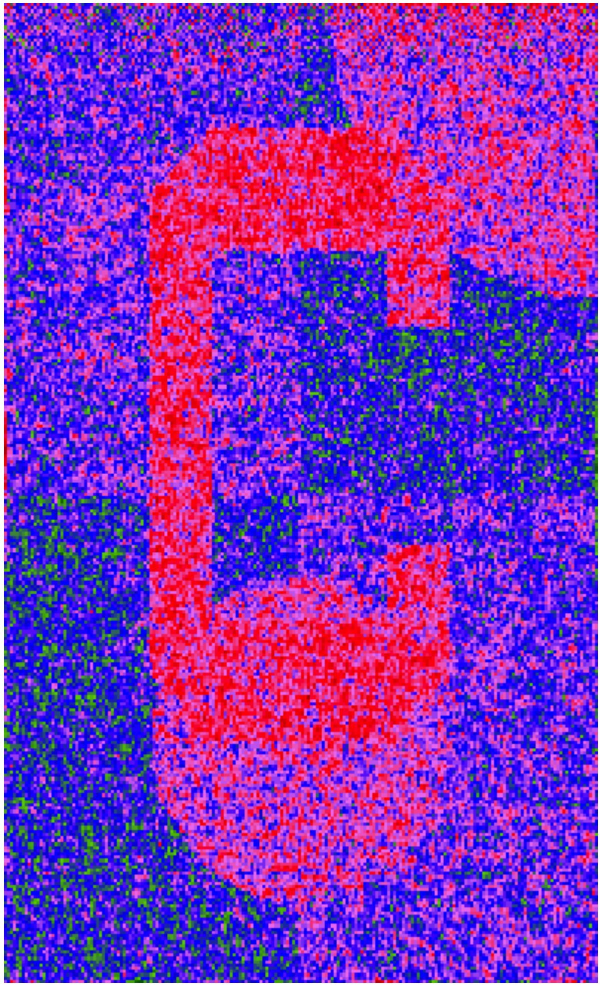


Specialized ADMM, 20 iterations

Proximal gradient descent, 1000 iterations

Coordinate descent, 10K cycles

Our problem: $$\min_{\theta} \frac{1}{2}\sum_{i=1}^{n}(y_i - \theta_i)^2 + \lambda \sum_{(i,j)\in E} |\theta_i - \theta_j|$$



Specialized ADMM, 20 iterations

Proximal gradient descent, 1000 iterations

Coordinate descent, 10K cycles

(Last two from the dual)

What is our conclusion here?

- Is the "Alternating Direction Method of Multipliers" (ADMM) a better method than proximal gradient descent or coordinate descent?

- In fact, different algorithms perform better / worse in different situations.

  In the 2d fused lasso problem:
  - Special ADMM: fast (structured subproblems)
  - Proximal gradient: slow (poor conditioning)
  - Coordinate descent: slow (large active set)

- I won't be able to teach you all of these. But if I offer convex optimization again at some point, you should consider registering.

Plan today

- Review of what we have learned so far

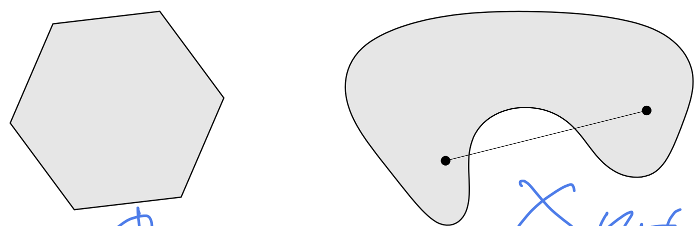
- An optimization view to ML
  - Modeling with optimization

- Convex optimization basics
  - Convex Set
  - Convex functions
  - Examples

Convex sets and functions

Convex set: $C \subseteq \mathbb{R}^n$ such that

$$x, y \in C \implies tx + (1-t)y \in C \text{ for all } 0 \le t \le 1$$

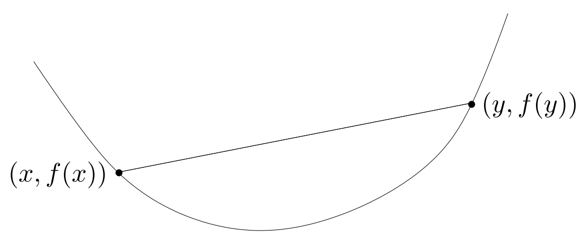


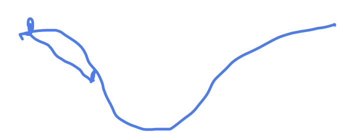
✗ not cvx

Convex function: $f : \mathbb{R}^n \to \mathbb{R}$ such that $\mathrm{dom}(f) \subseteq \mathbb{R}^n$ convex, and

$$f(tx + (1-t)y) \le tf(x) + (1-t)f(y) \quad \text{for all } 0 \le t \le 1$$

and all $x, y \in \mathrm{dom}(f)$





Convex optimization problems

Optimization problem:

$a^T x + b = 0$

"affine"

$$\begin{aligned} \min_{x \in D} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \le 0, \; i = 1, \ldots m \\ & h_j(x) = 0, \; j = 1, \ldots r \end{aligned}$$

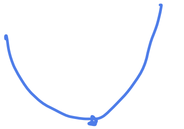
R^d

Here $D = \mathrm{dom}(f) \cap \bigcap_{i=1}^m \mathrm{dom}(g_i) \cap \bigcap_{j=1}^p \mathrm{dom}(h_j)$, common domain of all the functions

This is a convex optimization problem provided the functions f and $g_i, i = 1, \ldots m$ are convex, and $h_j, j = 1, \ldots p$ are affine:

$$h_j(x) = a_j^T x + b_j, \quad j = 1, \ldots p$$

Quick refresh of your memory on your knowledge from high school



$$\min_{x\in\mathbb{R}} x^2 - 4x + 9$$

$4 - 8 + 9 = 5$

$\nabla f(x) = 2x - 4 = 0$

$x^* = \frac{4}{2} = 2$

- What is the objective function?

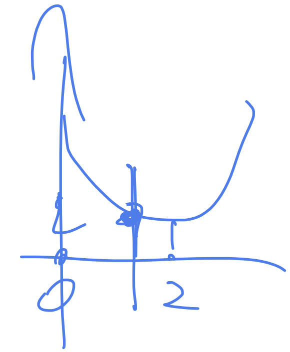
- What is the optimal objective function value?

$f^* = 5$

- What is the optimal solution?

$x^* = 2$

What about?



$$\min_{x \in [0,1]} \; x^2 - 4x + 9$$

- What is the optimal solution? How to work it out?

$x^* = 1$

- Can we reformulate it in a standard form?

$\min \; f(x)$

s.t. $\quad x \leq 1 \qquad g_1(x) \leq 1$

$\qquad -x \leq 0 \qquad g_2(x) \leq 0$

$\Rightarrow$

Optimality Condition

$\nabla f(x^*) \neq 0$?

Local minima are global minima

For convex optimization problems, local minima are global minima

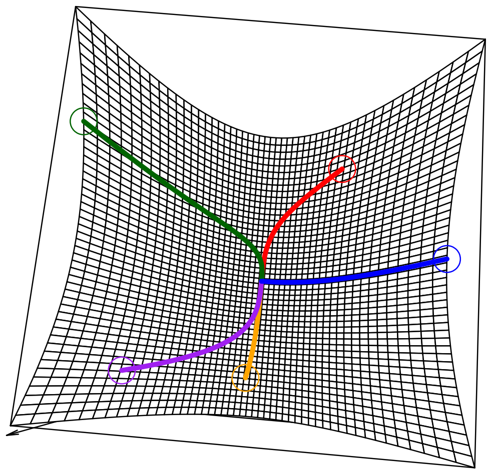
Formally, if x is feasible—$x \in D$, and satisfies all constraints—and minimizes f in a local neighborhood,

$$f(x) \leq f(y) \text{ for all feasible } y, \ \|x - y\|_2 \leq \rho,$$

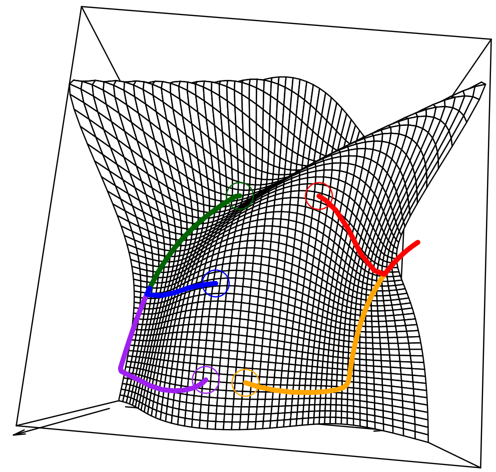
then

$$f(x) \leq f(y) \text{ for all feasible } y$$

This is a very useful fact and will save us a lot of trouble!



Convex

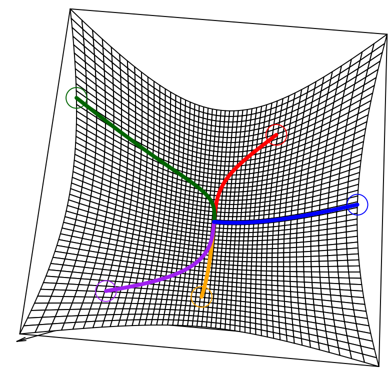


Nonconvex

In summary: why convexity?

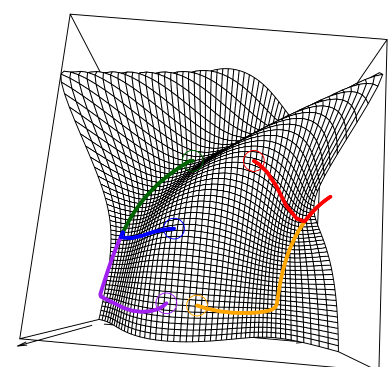
Why convexity? Simply put: because we can broadly understand and solve convex optimization problems

Nonconvex problems are mostly treated on a case by case basis



Reminder: a convex optimization problem is of the form

$$
\begin{aligned}
\min_{x \in D} \quad & f(x) \\
\text{subject to} \quad & g_i(x) \leq 0, \ i = 1, \dots m \\
& h_j(x) = 0, \ j = 1, \dots r
\end{aligned}
$$



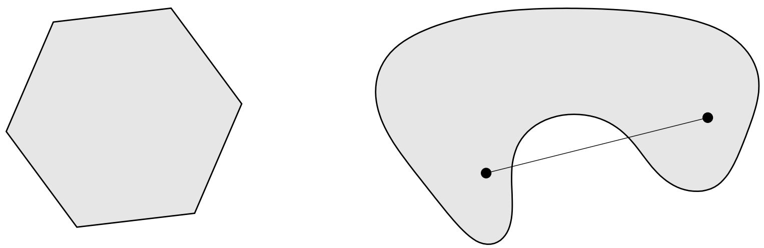
where f and g_i, $i = 1, \dots m$ are all convex, and h_j, $j = 1, \dots r$ are affine. Special property: any local minimizer is a global minimizer

Convex sets

Convex set: $C \subseteq \mathbb{R}^n$ such that

$$x, y \in C \implies tx + (1-t)y \in C \text{ for all } 0 \leq t \leq 1$$

In words, line segment joining any two elements lies entirely in set



Convex combination of $x_1, \ldots x_k \in \mathbb{R}^n$: any linear combination

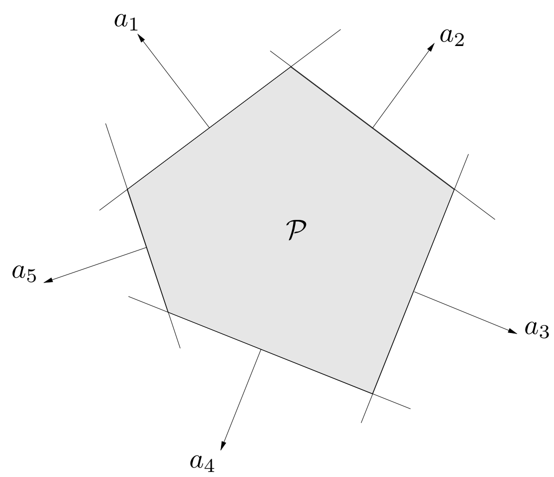
$$\theta_1 x_1 + \ldots + \theta_k x_k$$

with $\theta_i \geq 0$, $i = 1, \ldots k$, and $\sum_{i=1}^k \theta_i = 1$. Convex hull of a set C, $\mathrm{conv}(C)$, is all convex combinations of elements. Always convex

Examples of convex sets

- Trivial ones: empty set, point, line
- Norm ball: $\{x : \|x\| \leq r\}$, for given norm $\|\cdot\|$, radius r
- Hyperplane: $\{x : a^T x = b\}$, for given a, b
- Halfspace: $\{x : a^T x \leq b\}$
- Affine space: $\{x : Ax = b\}$, for given A, b

- Polyhedron: $\{x : Ax \le b\}$, where inequality $\le$ is interpreted componentwise. Note: the set $\{x : Ax \le b, Cx = d\}$ is also a polyhedron (why?)



- Simplex: special case of polyhedra, given by $\mathrm{conv}\{x_0, \ldots x_k\}$, where these points are affinely independent. The canonical example is the probability simplex,

$$\mathrm{conv}\{e_1, \ldots e_n\} = \{w : w \ge 0,\ 1^T w = 1\}$$

Operations preserving convexity

- Intersection: the intersection of convex sets is convex
- Scaling and translation: if C is convex, then

  $$aC + b = \{ax + b : x \in C\}$$

  is convex for any a, b
- Affine images and preimages: if $f(x) = Ax + b$ and C is convex then

  $$f(C) = \{f(x) : x \in C\}$$

  is convex, and if D is convex then

  $$f^{-1}(D) = \{x : f(x) \in D\}$$

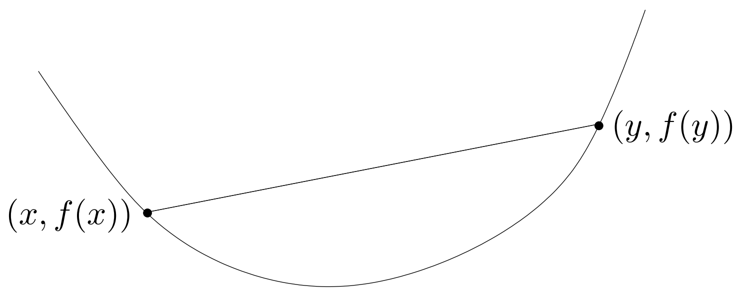
  is convex

Convex functions

Convex function: $f : \mathbb{R}^n \to \mathbb{R}$ such that $\mathrm{dom}(f) \subseteq \mathbb{R}^n$ convex, and

$$f(tx + (1-t)y) \le tf(x) + (1-t)f(y) \quad \text{for } 0 \le t \le 1$$

and all $x, y \in \mathrm{dom}(f)$



In words, function lies below the line segment joining $f(x), f(y)$

Concave function: opposite inequality above, so that

$$f \text{ concave} \iff -f \text{ convex}$$

Important modifiers:

- Strictly convex: $f\big(tx + (1-t)y\big) < tf(x) + (1-t)f(y)$ for $x \neq y$ and $0 < t < 1$. In words, f is convex and has greater curvature than a linear function

- Strongly convex with parameter $m > 0$: $f - \frac{m}{2}\|x\|_2^2$ is convex. In words, f is at least as convex as a quadratic function

Note: strongly convex $\Rightarrow$ strictly convex $\Rightarrow$ convex

(Analogously for concave functions)

Examples of convex functions

- Univariate functions:
  - Exponential function: e^{ax} is convex for any a over $\mathbb{R}$
  - Power function: x^a is convex for $a \geq 1$ or $a \leq 0$ over $\mathbb{R}_+$ (nonnegative reals)
  - Power function: x^a is concave for $0 \leq a \leq 1$ over $\mathbb{R}_+$
  - Logarithmic function: $\log x$ is concave over $\mathbb{R}_{++}$

- Affine function: $a^T x + b$ is both convex and concave

- Quadratic function: $\frac{1}{2} x^T Q x + b^T x + c$ is convex provided that $Q \succeq 0$ (positive semidefinite)

- Least squares loss: $\|y - Ax\|_2^2$ is always convex (since $A^T A$ is always positive semidefinite)

- Norm: $\|x\|$ is convex for any norm; e.g., ℓ_p norms,

$$\|x\|_p = \left(\sum_{i=1}^{n} x_i^p\right)^{1/p} \quad \text{for } p \geq 1, \quad \|x\|_\infty = \max_{i=1,\ldots n} |x_i|$$

and also operator (spectral) and trace (nuclear) norms,

$$\|X\|_{\mathrm{op}} = \sigma_1(X), \quad \|X\|_{\mathrm{tr}} = \sum_{i=1}^{r} \sigma_r(X)$$

where $\sigma_1(X) \geq \ldots \geq \sigma_r(X) \geq 0$ are the singular values of the matrix X

- Indicator function: if C is convex, then its indicator function

$$I_C(x) = \begin{cases} 0 & x \in C \\ \infty & x \notin C \end{cases}$$

is convex

- Support function: for any set C (convex or not), its support function

$$I_C^*(x) = \max_{y \in C} \; x^T y$$

is convex

- Max function: $f(x) = \max\{x_1, \ldots x_n\}$ is convex

Key properties of convex functions

- A function is convex if and only if its restriction to any line is convex

- Epigraph characterization: a function f is convex if and only if its epigraph

$$\mathrm{epi}(f) = \{(x,t) \in \mathrm{dom}(f) \times \mathbb{R} : f(x) \leq t\}$$

  is a convex set

- Convex sublevel sets: if f is convex, then its sublevel sets

$$\{x \in \mathrm{dom}(f) : f(x) \leq t\}$$

  are convex, for all $t \in \mathbb{R}$. The converse is not true

- First-order characterization: if f is differentiable, then f is convex if and only if $\mathrm{dom}(f)$ is convex, and

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

  for all $x, y \in \mathrm{dom}(f)$. Therefore for a differentiable convex function $\nabla f(x) = 0 \iff x$ minimizes f

- Second-order characterization: if f is twice differentiable, then f is convex if and only if $\mathrm{dom}(f)$ is convex, and $\nabla^2 f(x) \succeq 0$ for all $x \in \mathrm{dom}(f)$

- Jensen's inequality: if f is convex, and X is a random variable supported on $\mathrm{dom}(f)$, then $f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$

Operations preserving convexity

- Nonnegative linear combination: $f_1, \ldots f_m$ convex implies $a_1 f_1 + \ldots + a_m f_m$ convex for any $a_1, \ldots a_m \geq 0$

- Pointwise maximization: if f_s is convex for any $s \in S$, then $f(x) = \max_{s \in S}\ f_s(x)$ is convex. Note that the set S here (number of functions f_s) can be infinite

- Partial minimization: if $g(x, y)$ is convex in x, y, and C is convex, then $f(x) = \min_{y \in C}\ g(x, y)$ is convex

Example: distances to a set

Let C be an arbitrary set, and consider the maximum distance to C under an arbitrary norm $\|\cdot\|$:

$$f(x) = \max_{y \in C} \; \|x - y\|$$

Let's check convexity: $f_y(x) = \|x - y\|$ is convex in x for any fixed y, so by pointwise maximization rule, f is convex

Now let C be convex, and consider the minimum distance to C:

$$f(x) = \min_{y \in C} \; \|x - y\|$$

Let's check convexity: $g(x, y) = \|x - y\|$ is convex in x, y jointly, and C is assumed convex, so apply partial minimization rule

More operations preserving convexity

- Affine composition: if f is convex, then $g(x) = f(Ax + b)$ is convex

- General composition: suppose $f = h \circ g$, where $g : \mathbb{R}^n \to \mathbb{R}$, $h : \mathbb{R} \to \mathbb{R}$, $f : \mathbb{R}^n \to \mathbb{R}$. Then:
  - f is convex if h is convex and nondecreasing, g is convex
  - f is convex if h is convex and nonincreasing, g is concave
  - f is concave if h is concave and nondecreasing, g concave
  - f is concave if h is concave and nonincreasing, g convex

  How to remember these? Think of the chain rule when $n = 1$:

$$f''(x) = h''(g(x))g'(x)^2 + h'(g(x))g''(x)$$

- Vector composition: suppose that

$$f(x) = h\big(g(x)\big) = h\big(g_1(x), \ldots g_k(x)\big)$$

where $g : \mathbb{R}^n \to \mathbb{R}^k$, $h : \mathbb{R}^k \to \mathbb{R}$, $f : \mathbb{R}^n \to \mathbb{R}$. Then:

  - f is convex if h is convex and nondecreasing in each argument, g is convex
  - f is convex if h is convex and nonincreasing in each argument, g is concave
  - f is concave if h is concave and nondecreasing in each argument, g is concave
  - f is concave if h is concave and nonincreasing in each argument, g is convex

Example: log-sum-exp function

Log-sum-exp function: $g(x) = \log(\sum_{i=1}^{k} e^{a_i^T x + b_i})$, for fixed a_i, b_i, $i = 1, \ldots k$. Often called "soft max", as it smoothly approximates $\max_{i=1,\ldots k} (a_i^T x + b_i)$

How to show convexity? First, note it suffices to prove convexity of $f(x) = \log(\sum_{i=1}^{n} e^{x_i})$ (affine composition rule)

Now use second-order characterization. Calculate

$$\nabla_i f(x) = \frac{e^{x_i}}{\sum_{\ell=1}^{n} e^{x_\ell}}$$

$$\nabla_{ij}^2 f(x) = \frac{e^{x_i}}{\sum_{\ell=1}^{n} e^{x_\ell}} 1\{i = j\} - \frac{e^{x_i} e^{x_j}}{(\sum_{\ell=1}^{n} e^{x_\ell})^2}$$

Write $\nabla^2 f(x) = \mathrm{diag}(z) - zz^T$, where $z_i = e^{x_i} / (\sum_{\ell=1}^{n} e^{x_\ell})$. This matrix is diagonally dominant, hence positive semidefinite

Next lecture: Support Vector Machines

- You will learn about why is SVM
  - “Max-margin”
  - The notorious “Kernel trick” in ML

- Also some hammers from convex optimization
  - Optimality (KKT) conditions
  - Lagrange Duality