CPSC 540: Machine Learning

Convex Optimization

Mark Schmidt

University of British Columbia

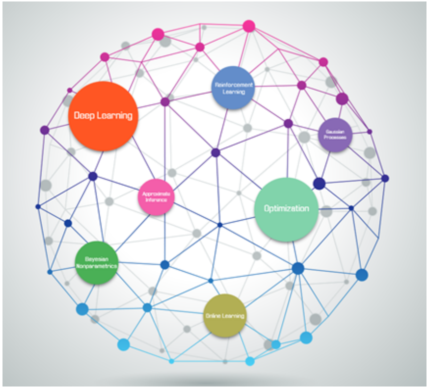
Winter 2018

Admin

- Audditting/registration forms:
  - Submit them at end of class, pick them up end of next class.
  - I need your prereq form before I'll sign registration forms.
  - I wrote comments on the back of some forms.
- Website/Piazza:
  - `https://www.cs.ubc.ca/~schmidtm/Courses/540-W19`.
  - `https://piazza.com/ubc.ca/winterterm22018/cpsc540`.
- Tutorials: start today after class.
- Office hours: start Wednesday after class.
- Assignment 1 due Friday.
  - All questions now posted, see Piazza update thread for changes.

Current Hot Topics in Machine Learning

- Graph of most common keywords among ICML papers in 2015:



- Why is there so much focus on deep learning and optimization?

Why Study Optimization in CPSC 540?

- In machine learning, training is typically written as an optimization problem:
  - We optimize parameters w of model, given data.
- There are some exceptions:
  1. Methods based on counting and distances (KNN, random forests).
     - See CPSC 340.
  2. Methods based on averaging and integration (Bayesian learning).
     - Later in course.

  But even these models have parameters to optimize.
- But why study optimization? Can't I just use optimization libraries?
  - "\", linprog, quadprog, CVX, MOSEK, and so.

The Effect of Big Data and Big Models

- Datasets are getting huge, we might want to train on:
  - Entire medical image databases.
  - Every webpage on the internet.
  - Every product on Amazon.
  - Every rating on Netflix.
  - All flight data in history.
- With bigger datasets, we can build bigger models:
  - Complicated models can address complicated problems.
  - Regularized linear models on huge datasets are standard industry tool.
  - Deep learning allows us to learn features from huge datasets.

The Effect of Big Data and Big Models

- But optimization becomes a bottleneck because of time/memory.
  - We can't afford $O(d^2)$ memory, or an $O(d^2)$ operation.
  - Going through huge datasets hundreds of times is too slow.
  - Evaluating huge models many times may be too slow.
- Next class we'll start large-scale machine learning.
- Today we'll discuss problems that have "off the shelf" optimization methods.

Least Squares and Linear Equalities

- In 340 we showed that solving least squares optimization problem,

$$\operatorname*{argmin}_{w \in \mathbb{R}^d} \|Xw - y\|^2.$$

  is equivalent to solving the normal equations,

$$(X^\top X)w = X^\top y.$$

- This is a special case of solving a set of linear equalities, $Aw = b$.
  - Set of equalities of the form $a_i^\top w = b_i$ for vectors a_i and scalaras b_i.

- There exists reliable "off the shelf" software for solving linear equalities.

Linear Inequalities and Linear Programs

- We can also solve linear inequalities $Aw \leq b$ (instead of $Aw = b$).
  - A set of inequalities of the form $a_i^T w \leq b_i$ for vectors a_i and scalars b_i.

- More generally, there are "off the shelf" codes for solving linear programs:

$$\underset{w}{\text{argmin}}\, w^\top c, \quad \text{among the } w \text{ satisfying } Aw \leq b,$$

  which minimize a linear cost function and linear constraints.

- Another common problem class with "off the shelf" tools is quadratic programs.
  - Minimize a quadratic cost function with linear constraints.
  - For example, non-negative least squares minimizies $\|Xw - y\|^2$ subject to $w \geq 0$.

Robust Regression as Linear Program

- Consider regression with the absolute error as the loss,

$$\underset{w \in \mathbb{R}^d}{\text{argmin}} \sum_{i=1}^{n} |w^\top x^i - y^i|.$$

- In CPSC 340 we argued that this is more robust to outliers than least squares.
- This problem can be turned into a linear program.
  - You can then solve it with "off the shelf" linear programming software.
- Our first step is re-writing absolute value using $|\alpha| = \max\{\alpha, -\alpha\}$,

$$\underset{w \in \mathbb{R}^d}{\text{argmin}} \sum_{i=1}^{n} \max\{w^\top x^i - y^i, y^i - w^\top x^i\}.$$

Robust Regression as a Linear Program

- So we've show that L1-regression is equivalent to

$$\underset{w\in\mathbb{R}^d}{\text{argmin}} \sum_{i=1}^{n} \max\{w^\top x^i - y^i, y^i - w^\top x^i\}.$$

- Second step: introduce n variables r_i that upper bound the max functions,

$$\underset{w\in\mathbb{R}^d, r\in\mathbb{R}^n}{\text{argmin}} \sum_{i=1}^{n} r_i, \quad \text{with} \quad r_i \geq \max\{w^\top x^i - y^i, y^i - w^\top x^i\}, \forall i.$$

- This is a linear objective in terms of the parameters w and r.

- Problems are equivalent: solutions must have $r_i = |w^\top x^i - y^i|$.
  - If $r_i < |w^\top x^i - y^i|$, then one of the constraints are not satisfied (not a solution).
  - If $r_i > |w^\top x^i - y^i|$, then we could decrease r_i and get lower cost (not a solution).

Robust Regression as a Linear Program

- So we've show that L1-regression is equivalent to

$$\underset{w\in\mathbb{R}^d, r\in\mathbb{R}^n}{\text{argmin}} \sum_{i=1}^{n} r_i, \quad \text{with} \quad r_i \geq \max\{w^\top x^i - y^i, y^i - w^\top x^i\}, \forall i,$$

  which has a linear cost function but non-linear constraints.

- Third step: split max constraints into individual linear constraints,

$$\underset{w\in\mathbb{R}^d,\, r\in\mathbb{R}^n}{\text{argmin}} \sum_{i=1}^{n} r_i, \quad \text{with} \quad r_i \geq w^\top x^i - y^i,\ r_i \geq y^i - w^\top x^i, \forall i.$$

- Being greater than the max is equivalent to being greater than each.

Minimizing Absolute Values and Maxes

- We've shown that L1-norm regression can be written as a linear program,

$$\underset{w\in\mathbb{R}^d,\, r\in\mathbb{R}^n}{\text{argmin}} \sum_{i=1}^n r_i, \quad \text{with} \quad r_i \geq w^\top x^i - y^i,\; r_i \geq y^i - w^\top x^i, \forall i,$$

- For medium-sized problems, we can solve this with Julia's *linprog*.
  - Linear programs are solvable in polynomial time.

- A general approach for minimizing absolute values and/or maximums:
  1. Replace absolute values with maximums.
  2. Replace maximums with new variables, constrain these to bound maixmums.
  3. Transform to linear constraints by splitting the maximum constraints.

Example: Support Vector Machine as a Quadratic Program

- The SVM optimization problem is

$$\operatorname*{argmin}_{w\in\mathbb{R}^d} \sum_{i=1}^{n} \max\{0, 1 - y^i w^\top x^i\} + \frac{\lambda}{2}\|w\|^2,$$

- Introduce new variables to upper-bound the maxes,

$$\operatorname*{argmin}_{w\in\mathbb{R}^d, r\in\mathbb{R}^n} \sum_{i=1}^{n} r_i + \frac{\lambda}{2}\|w\|^2, \quad \text{with} \quad r_i \geq \max\{0, 1 - y^i w^\top x^i\}, \forall i.$$

- Split the maxes into separate constraints,

$$\operatorname*{argmin}_{w\in\mathbb{R}^d, r\in\mathbb{R}^n} \sum_{i=1}^{n} r_i + \frac{\lambda}{2}\|w\|^2, \quad \text{with} \quad r_i \geq 0,\ r_i \geq 1 - y^i w^\top x^i,$$

which is a quadratic program (quadratic objective with linear constraints).

General Lp-norm Losses

- Consider minimizing the regression loss

$$f(w) = \|Xw - y\|_p,$$

with a general Lp-norm, $\|r\|_p = \left(\sum_{i=1}^n |r_i|^p\right)^{\frac{1}{p}}$.
- With $p = 2$, we can minimize the function as a linear system.
  - Raise to the power of 2 and set gradient to zero.
- With $p = 1$, we can minimize the function using linear programming.
- With $p = \infty$, we can also use linear programming (using same trick).
- For $1 < p < \infty$, we can turn this into a convex optimization problem.
  - By raising it to the power p (next topic).
- If we use $p < 1$ (which is not a norm), minimizing f is NP-hard.

Outline

1 Minimizing Maxes of Linear Functions

2 Convex Functions

Convex Optimization

- Consider an optimization problem of the form

$$\min_{w \in \mathcal{C}} f(w).$$

where we are minimizing a function f subject to w being in the set $\mathcal{C}$.

- We say that this is a convex optimization problem if:
  - The set $\mathcal{C}$ is a convex set.
  - The function f is a convex function.

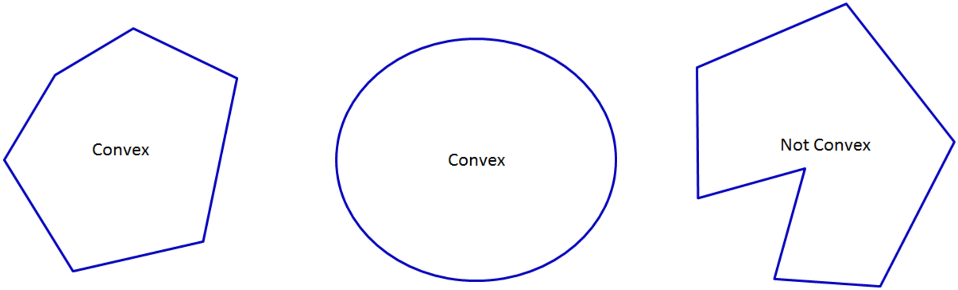
- Linear programming is a special case of convex optimization.

Convex Optimization

- Key property of convex optimization problems:
  - All local optima are global optima.
- Convexity is usually a good indicator of tractability:
  - Minimizing convex functions is usually easy.
  - Minimizing non-convex functions is usually hard.
- Off-the-shelf software solves many classes of convex problems (*MathProgBase*).

Definition of Convex Sets

- A set $\mathcal{C}$ is convex if the line between any two points stays also in the set.



Definition of Convex Sets

- To formally define convex sets, we use notion of convex combination:
  - A convex combination of two variables w and v is given by

$$\theta w + (1-\theta)v \quad \text{for any} \quad 0 \leq \theta \leq 1,$$

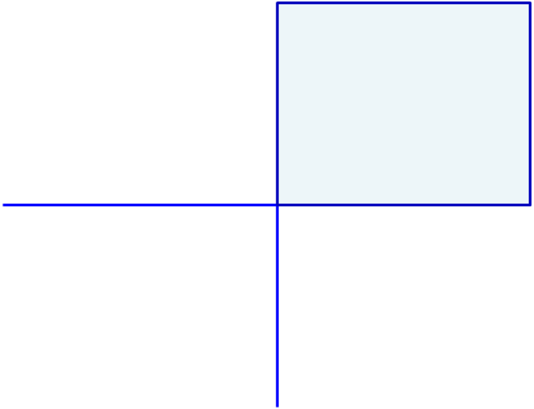
    which characterizes the points on the line between w and v.

- A set $\mathcal{C}$ is convex if convex combinations of points in the set are also in the set:
  - For all $w \in \mathcal{C}$ and $v \in \mathcal{C}$ we have $\underbrace{\theta w + (1-\theta)v}_{\text{convex comb}} \in \mathcal{C}$ for $0 \leq \theta \leq 1$.

- This definition allows us to prove the convexity of many simple sets.

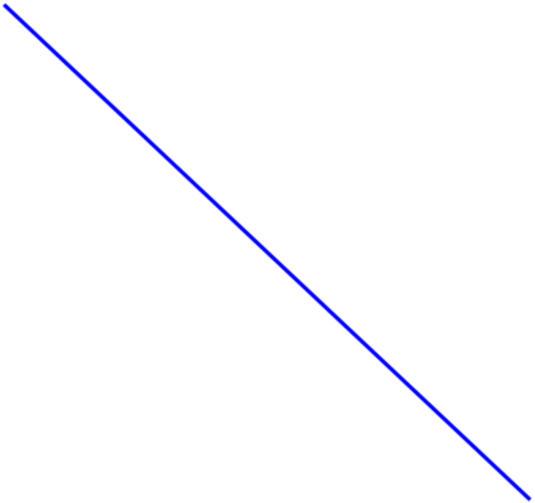
Examples of Simple Convex Sets

- Real space $\mathbb{R}^d$.
- Positive orthant $\mathbb{R}^d_+ : \{w \mid w \geq 0\}$.
- Hyper-plane: $\{w \mid a^\top w = b\}$.
- Half-space: $\{w \mid a^\top w \leq b\}$.
- Norm-ball: $\{w \mid \|w\|_p \leq \tau\}$.
- Norm-cone: $\{(w, \tau) \mid \|w\|_p \leq \tau\}$.



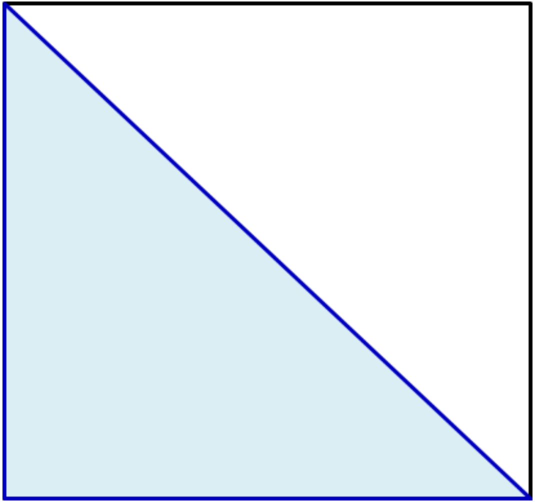
Examples of Simple Convex Sets

- Real space $\mathbb{R}^d$.
- Positive orthant $\mathbb{R}^d_+ : \{w \mid w \geq 0\}$.
- Hyper-plane: $\{w \mid a^\top w = b\}$.
- Half-space: $\{w \mid a^\top w \leq b\}$.
- Norm-ball: $\{w \mid \|w\|_p \leq \tau\}$.
- Norm-cone: $\{(w, \tau) \mid \|w\|_p \leq \tau\}$.



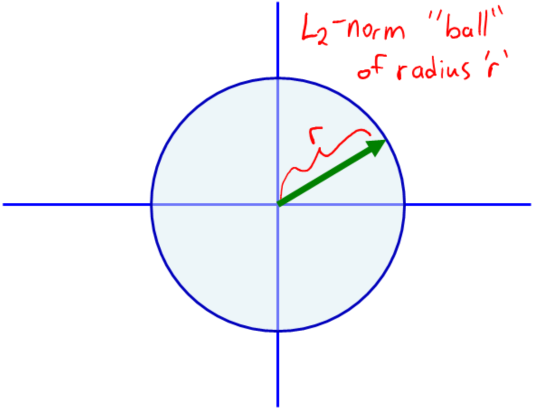
Examples of Simple Convex Sets

- Real space $\mathbb{R}^d$.
- Positive orthant $\mathbb{R}^d_+ : \{w \mid w \geq 0\}$.
- Hyper-plane: $\{w \mid a^\top w = b\}$.
- Half-space: $\{w \mid a^\top w \leq b\}$.
- Norm-ball: $\{w \mid \|w\|_p \leq \tau\}$.
- Norm-cone: $\{(w, \tau) \mid \|w\|_p \leq \tau\}$.



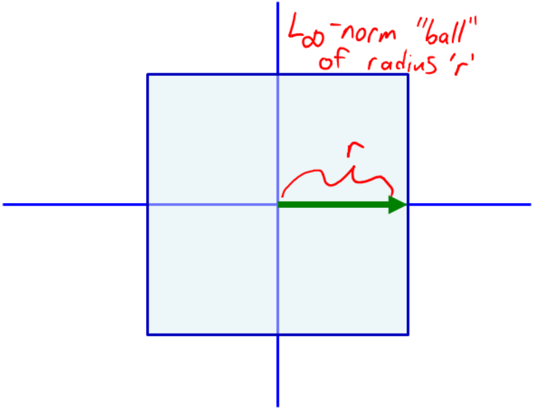
Examples of Simple Convex Sets

- Real space $\mathbb{R}^d$.
- Positive orthant $\mathbb{R}^d_+ : \{w \mid w \geq 0\}$.
- Hyper-plane: $\{w \mid a^\top w = b\}$.
- Half-space: $\{w \mid a^\top w \leq b\}$.
- Norm-ball: $\{w \mid \|w\|_p \leq \tau\}$.
- Norm-cone: $\{(w, \tau) \mid \|w\|_p \leq \tau\}$.



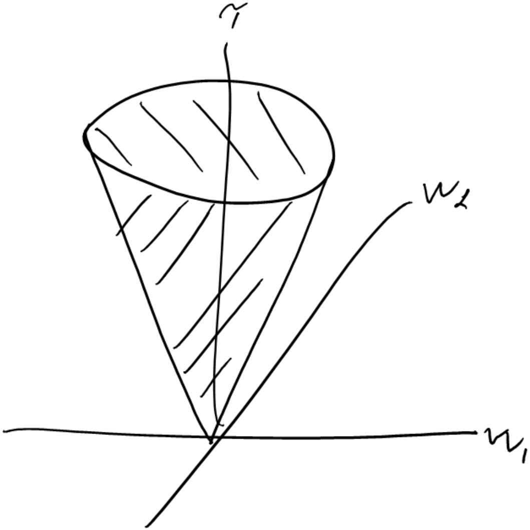
Examples of Simple Convex Sets

- Real space $\mathbb{R}^d$.
- Positive orthant $\mathbb{R}^d_+ : \{w \mid w \geq 0\}$.
- Hyper-plane: $\{w \mid a^\top w = b\}$.
- Half-space: $\{w \mid a^\top w \leq b\}$.
- Norm-ball: $\{w \mid \|w\|_p \leq \tau\}$.
- Norm-cone: $\{(w, \tau) \mid \|w\|_p \leq \tau\}$.



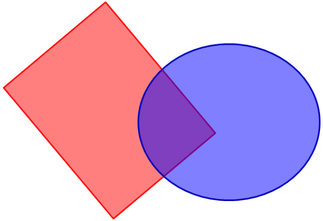
Examples of Simple Convex Sets

- Real space $\mathbb{R}^d$.
- Positive orthant $\mathbb{R}^d_+ : \{w \mid w \geq 0\}$.
- Hyper-plane: $\{w \mid a^\top w = b\}$.
- Half-space: $\{w \mid a^\top w \leq b\}$.
- Norm-ball: $\{w \mid \|w\|_p \leq \tau\}$.
- Norm-cone: $\{(w, \tau) \mid \|w\|_p \leq \tau\}$.



Showing a Set is Convex from Intersections

- The intersection of convex sets is convex.



- We can prove convexity of a set by showing it's an intersection of convex sets.
- Example: linear programs have constraints of the form $Aw \leq b$.
    - Each constraints $a_i^\top b_i$ defines a half-space.
    - Half-spaces are convex sets.
    - So the set of w satisfying $Aw \leq b$ is the intersection of convex sets.

Showing a Set is Convex from a Convex Function

- The set $\mathcal{C}$ is often the intersection of a set of inequalities of the form

$$\{w \mid g(w) \leq \tau\},$$

for some function g and some number τ.

- Sets defined like this are convex if g is a convex function (see bonus).
  - This follows from the definition of a convex function (next topic).

- Example:
  - The set of w where $w^2 \leq 10$ forms a convex set by convexity of w^2.
  - Specifically, the set is $[-\sqrt{10}, \sqrt{10}]$.

Digression: k-way Convex Combinations and Differentiability Classes

- A convex combintion of k vectors $\{w_1, w_2, \ldots, w_k\}$ is given by

$$\sum_{c=1}^{k} \theta_c w_c \quad \text{where} \quad \sum_{c=1}^{k} \theta_c = 1, \ \theta_c \geq 0.$$

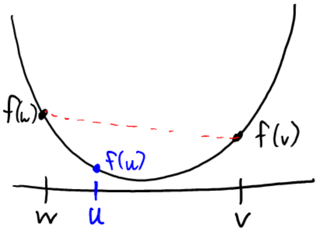
- We'll define convex functions for different differentiability classes:
  - C^0 is the set of continuous functions.
  - C^1 is the set of continuous functions with continuous first-derivatives.
  - C^2 is the set of continuous functions with continuous first- and second-derivatives.

Definitions of Convex Functions

- Four quivalent definitions of convex functions (depending on differentiability):
  1. A C^0 function is convex iff the area above the function is a convex set.
  2. A C^0 function is convex iff the function is always below its "chords" between points.
  3. A C^1 function is convex iff the function is always above its tangent planes.
  4. A C^2 function is convex iff it is curved upwards everwhere.
     - If the function is univariate this means $f''(w) \geq 0$ for all w.
- Univariate examples where you can show $f''(w) \geq 0$ for all w:
  - Quadratic $w^2 + bw + c$ with $a \geq 0$.
  - Linear: $aw + b$.
  - Constant: b.
  - Exponential: $\exp(aw)$.
  - Negative logarithm: $-\log(w)$.
  - Negative entropy: $w \log w$, for $w > 0$.
  - Logistic loss: $\log(1 + \exp(-w))$.

C^0 Definitions of Convex Functions

- A function f is convex iff the area above the function is a convex set.



- Equivalently, the function is always below its "chords" between points.

$$\underbrace{f(\theta w + (1-\theta)v)}_{\text{convex comb}} \leq \underbrace{\theta f(w) + (1-\theta)f(v)}_{\text{"chord"}}, \quad \text{for all } w \in \mathcal{C}, v \in \mathcal{C}, 0 \leq \theta \leq 1.$$

- Implies all local minima of convex functions are global minima.
  - Indeed, $\nabla f(w) = 0$ means w is a global minima.

Convexity of Norms

- The C^0 definition can be used to show that all norms are convex:
  - If $f(w) = \|w\|_p$ for a generic norm, then we have

$$\begin{aligned} f(\theta w + (1-\theta)v) &= \|\theta w + (1-\theta)v\|_p \\ &\leq \|\theta w\|_p + \|(1-\theta)v\|_p && \text{(triangle inequality)} \\ &= |\theta| \cdot \|w\|_p + |1-\theta| \cdot \|v\|_p && \text{(absolute homogeneity)} \\ &= \theta\|w\|_p + (1-\theta)\|v\|_p && (0 \leq \theta \leq 1) \\ &= \theta f(w) + (1-\theta)f(v), && \text{(definition of } f\text{)} \end{aligned}$$

    so f is always below the "chord".
- See course webpage notes on norms if the above steps aren't familiar.

- Also note that all squared norms are convex.
  - These are all convex: $|w|, \|w\|, \|w\|_1, \|w\|^2, \|w_1\|^2, \|w\|_\infty$,...

Operations that Preserve Convexity

- There are a few operations that preserve convexity.
  - Can show convexity by writing as sequence of convexity-preserving operations.

- If f and g are convex functions, the following preserve convexity:
  1. Non-negative scaling: $$h(w) = \alpha f(w).$$
  2. Sum: $$h(w) = f(w) + g(w).$$
  3. Maximum: $$h(w) = \max\{f(w), g(w)\}.$$
  4. Composition with affine map: $$h(w) = f(Aw + b),$$

     where an affine map $w \mapsto Aw + b$ is a multi-input multi-output linear function.
     - Like $g(w) = Aw + b$ which takes in a vector and outputs a vector.

- But note that composition $f(g(w))$ of convex f and g is not convex in general.

Convexity of SVMs

- If f and g are convex functions, the following preserve convexity:
  1. Non-negative scaling.
  2. Sum.
  3. Maximum.
  4. Composition with affine map.
- We can use these to quickly show that SVMs are convex,

$$f(w) = \sum_{i=1}^{n} \max\{0, 1 - y^i w^\top x^i\} + \frac{\lambda}{2}\|w\|^2.$$

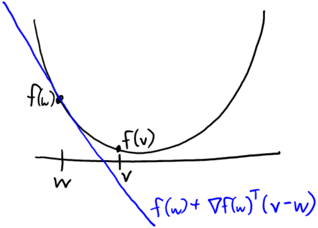
- Second term is squared norm multiplied by non-negative $\frac{\lambda}{2}$.
  - Squared norms are convex, and non-negative scaling perserves convexity.
- First term is sum(max(linear)). Linear is convex and sum/max preserve convexity.
- Since both terms are convex, and sums preserve convexity, SVMs are convex.

C^1 Definition of Convex Functions

- Convex functions must be continuous, and have a domain that is a convex set.
  - But they may be non-differentiable.

- A *differentiable* (C^1) function f is convex iff f is always above tangent planes.

$$f(v) \geq f(w) + \nabla f(w)^\top (v - w), \quad \forall w \in \mathcal{C}, v \in \mathcal{C}.$$



- Notice that $\nabla f(w) = 0$ implies $f(v) \geq f(w)$ for all v, so w is a global minimizer.

C^2 Definition of Convex Functions

- The multivariate C^2 definition is based on the Hessian matrix, $\nabla^2 f(w)$.
  - The matrix of second partial derivatives,

$$\nabla^2 f(w) = \begin{bmatrix} \frac{\partial}{\partial w_1 \partial w_1} f(w) & \frac{\partial}{\partial w_1 \partial w_2} f(w) & \cdots & \frac{\partial}{\partial w_1 \partial w_d} f(w) \\ \frac{\partial}{\partial w_2 \partial w_1} f(w) & \frac{\partial}{\partial w_2 \partial w_2} f(w) & \cdots & \frac{\partial}{\partial w_2 \partial w_d} f(w) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial w_d \partial w_1} f(w) & \frac{\partial}{\partial w_d \partial w_2} f(w) & \cdots & \frac{\partial}{\partial w_d \partial w_d} f(w) \end{bmatrix}$$

- In the case of least squares, we can write the Hessian for any w as

$$\nabla^2 f(w) = X^\top X,$$

see course webpage notes on the gradients/Hessians of linear/quadratic functions.

Convexity of Twice-Differentiable Functions

- A C^2 function is convex iff:

$$\nabla^2 f(w) \succeq 0,$$

  for all w in the domain ("curved upwards" in every direction).

- This notation $A \succeq 0$ means that A is positive semidefinite.

- Two equivalent definitions of a positive semidefinite matrix A:
  1. All eigenvalues of A are non-negative.
  2. The quadratic $v^\top A v$ is non-negative for all vectors v.

Convexity and Least Squares

- We can use twice-differentiable condition to show convexity of least squares,

$$f(w) = \frac{1}{2}\|Xw - y\|^2.$$

- The Hessian of this objective for any w is given by

$$\nabla^2 f(w) = X^\top X.$$

- So we want to show that $X^\top X \succeq 0$ or equivalently that $v^\top X^\top X v \geq 0$ for all v.
- We can show this by non-negativity of norms,

$$v^\top X^\top X v = \underbrace{(Xv)^\top (Xv)}_{u^\top u} = \underbrace{\|Xv\|^2}_{\|u\|^2} \geq 0,$$

so least squares is convex and solving $\nabla f(w) = 0$ gives *global minimum*.

Summary

- Converting non-smooth problems involving max to constrained smooth problems.
- Convex optimization problems are a class that we can usually efficiently solve.
- Showing functions and sets are convex.
  - Either from definitions or convexity-preserving operations.
- C^2 definition of convex functions that the Hessian is positive semidefinite.

- How many iterations of gradient descent do we need?

Showing that Hyper-Planes are Convex

- Hyper-plane: $\mathcal{C} = \{w \mid a^\top w = b\}$.
  - If $w \in \mathcal{C}$ and $v \in \mathcal{C}$, then we have $a^\top w = b$ and $a^\top v = b$.
  - To show $\mathcal{C}$ is convex, we can show that $a^\top u = b$ for u between w and v.

$$\begin{aligned} a^\top u &= a^\top(\theta w + (1-\theta)v) \\ &= \theta(a^\top w) + (1-\theta)(a^\top v) \\ &= \theta b + (1-\theta)b = b. \end{aligned}$$

- Alternately, if you knew that linear functions $a^\top w$ are convex, then $\mathcal{C}$ is the intersection of $\{w \mid a^\top w \leq b\}$ and $\{w \mid a^\top w \geq b\}$.

Convex Sets from Functions

- For sets of the form

$$\mathcal{C} = \{w \mid g(w) \leq \tau\},$$

If g is a convex function, then $\mathcal{C}$ is a convex set:

$$g(\underbrace{\theta w + (1-\theta)v}_{\text{convex comb}}) \leq \underbrace{\theta g(w) + (1-\theta)g(v)}_{\text{by convexity}} \leq \underbrace{\theta\tau + (1-\theta)\tau}_{\text{definition of } g} = \tau,$$

which means convex combinations are in the set.

More Examples of Convex Functions

- Examples of more exotic convex sets over matrix variables:
  - The set of positive semidefinite matrices $\{W \mid W \succeq 0\}$.
  - The set of positive definite matrices $\{W \mid W \succ 0\}$.

- Some more exotic examples of convex functions:
  - $f(w) = \log(\sum_{j=1}^{d} \exp(w_j))$ (log-sum-exp function).
  - $f(W) = -\log \det W$ for $W \succ 0$ (negative log-determinant over positive-definite matrices).
  - $f(W, v) = v^\top W^{-1} v$ for $W \succ 0$.