CPSC 540: Machine Learning

MAP Estimation, Convex Functions

Mark Schmidt

University of British Columbia

Winter 2017

Admin

- Auditting/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.
- Website/Piazza:
  - `https://www.cs.ubc.ca/~schmidtm/Courses/540-W17`.
  - `https://piazza.com/ubc.ca/winterterm22016/cpsc540`.
- Tutorials: start this Friday (4:00 in DMP 110).
- Assignment 1 due January 16.
  - 1 late day to hand it in January 18.
  - 2 late days to hand it in January 23.

Last Time: Loss Plus Regularizer Framework

- We discussed the typical "minimizing loss plus regularizer" framework,

$$f(w) = \underbrace{\sum_{i=1}^{n} f_i(w)}_{\text{data-fitting term}} + \underbrace{\lambda g(w)}_{\text{regularizer}}.$$

- Loss function f_i measures how well we fit example i with parameters w.
- Regularizer g measures how complicated the model is with parameters w.

- Regularization parameter $\lambda > 0$ controls strength of regularization:
  - Usually set by using a validation set or with cross-validation.

Last Time: L2-Regularized Least Squares

- One of the simplest examples is L2-regularized least squares:

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(w^Tx^i - y^i)^2 + \frac{\lambda}{2}\sum_{j=1}^{d} w_j^2,$$

- We showed how to write this in matrix and norm notation:

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

- We showed how to derive the gradient and minimum of quadratics,

$$\nabla f(w) = X^TXw - X^Ty + \lambda w, \quad w^* = (X^TX + \lambda I)^{-1}(X^Ty).$$

- We showed how to derive the Hessian and that it is positive-definite,

$$\nabla^2 f(w) = X^TX + \lambda I \succ 0.$$

- Today: a probabilistic perspective on the loss plus regularizer framework.

Logistic Regression for Binary y^i

- After squared error, second most common loss function is logistic loss,

$$f(w) = \sum_{i=1}^{n} \log(1 + \exp(-y^i w^T x^i)) + \frac{\lambda}{2}\|w\|^2,$$

  for binary $y^i \in \{-1, +1\}$ and where we make predictions using $\hat{y}^i = \text{sign}(w^T \hat{x}^i)$.
- This is not a norm, so where does it come from?

- When $\lambda = 0$, this is derived as a maximum likeilihood estimate (MLE).
- When $\lambda > 0$, this is derived as a maximum a posteriori (MAP) estimate.

Maximum Likelihood Estimation (MLE)

- MLE in an abstract setting:
  - We have a dataset D.
  - We want to pick a model h among a set of models $\mathcal{H}$.
  - We define the likelihood as the probability mass/density function $p(D|h)$.
  - We choose the model h^* that maximizes the likelihood,

$$h^* \in \underset{h \in \mathcal{H}}{\operatorname{argmax}}\, p(D|h).$$

- MLE has appealing "consistency" properties as $n \to \infty$ (take STAT 560/561).
- In the case of regression, we usually maximize the conditional likelihood,

$$p(y|X, w),$$

where we condition on the features X.

Minimizing the Negative Log-Likelihood

- To maximize the likelihood, usually we minimize the negative log-likelihood,

$$h^* \in \underset{h \in \mathcal{H}}{\text{argmax}}\, p(D|h) \equiv \underset{h \in \mathcal{H}}{\text{argmin}} - \log p(D|h),$$

- This yields same solution.
  - Logarithm is monotonic: if $\alpha > \beta$ then $\log(\alpha) > \log(\beta)$.
  - Changing sign flips max to min.

- See "Max and Argmax" notes on the webpage if the above seems strange.

Minimizing the Negative Log-Likelihood

- We use logarithm because it turns multiplication into addition,

$$\log(\alpha\beta) = \log(\alpha) + \log(\beta),$$

or more generally

$$\log\left(\prod_{i=1}^{n} a_i\right) = \sum_{i=1}^{n} \log(a_i).$$

- If data is n IID samples D_i then $p(D|h) = \prod_{i=1}^{n} p(D_i|h)$, and our MLE is

$$h^* \in \underset{h\in\mathcal{H}}{\text{argmax}} \prod_{i=1}^{n} p(D_i|h) \equiv \underset{h\in\mathcal{H}}{\text{argmin}} - \sum_{i=1}^{n} \log p(D_i|h).$$

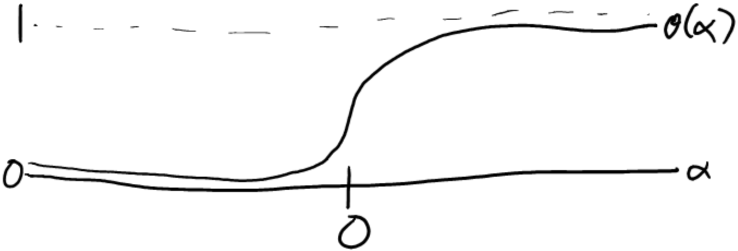
MLE Interpretation of Logistic Regression

- For IID regression problems the conditional negative log-likelihood can be written

$$-\log p(y|X,w) = -\log\left(\prod_{i=1}^n p(y^i|x^i,w)\right) = -\sum_{i=1}^n \log p(y^i|x^i,w).$$

- Logistic regression assumes conditional likelihood using sigmoid function σ,

$$p(y^i|x^i,w) = \sigma(y^i w^T x^i), \quad \text{where} \quad \sigma(\alpha) = \frac{1}{1+\exp(-\alpha)}.$$



MLE Interpretation of Logistic Regression

- For IID regression problems the conditional negative log-likelihood can be written

$$-\log p(y|X,w) = -\log\left(\prod_{i=1}^n p(y^i|x^i,w)\right) = -\sum_{i=1}^n \log p(y^i|x^i,w).$$

- Logistic regression assumes conditional likelihood using sigmoid function σ,

$$p(y^i|x^i,w) = \sigma(y^i w^T x^i), \quad \text{where} \quad \sigma(\alpha) = \frac{1}{1+\exp(-\alpha)}.$$

- Plugging in the sigmoid we get

$$f(w) = -\sum_{i=1}^n \log\left(\frac{1}{1+\exp(-y^i w^T x^i)}\right) = \sum_{i=1}^n \underbrace{\log(1+\exp(-y^i w^T x^i))}_{\text{logistic loss}},$$

using $\log(1) = 0$.
- Many loss functions are equivalent to negative log-likelihoods.

Least Squares as Conditional-Gaussian MLE

- Recall the Gaussian (normal) distribution,

$$p(\alpha|\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(\mu-\alpha)^2}{2\sigma^2}\right).$$

- Least squares is MLE assuming Gaussian conditional likelihood with mean w^Tx^i,

$$\begin{aligned} p(y^i|x^i,w,\sigma^2) &= \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(w^Tx^i-y^i)^2}{2\sigma^2}\right) \\ &\propto \exp\left(-\frac{(w^Tx^i-y^i)^2}{2\sigma^2}\right), \end{aligned}$$

  where for probabilities $\propto$ means "equal up to a constant not depending on y^i".

- Another way we'll write this assumption is

$$y^i \sim \mathcal{N}(w^Tx^i,\sigma^2),$$

  which is read "y^i is generated from a Gaussian with mean w^Tx^i and variance σ^2".

Least Squares as Conditional-Gaussian MLE

- Least squares is the MLE under our assumption that $y^i \sim \mathcal{N}(w^T x^i, \sigma^2)$,

$$
\begin{aligned}
w^* &\in \operatorname*{argmin}_{w \in \mathbb{R}^d} \sum_{i=1}^n -\log p(y^i | w, x^i) \\
&\equiv \operatorname*{argmin}_{w \in \mathbb{R}^d} \sum_{i=1}^n -\log\left(\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(w^T x^i - y^i)^2}{2\sigma^2}\right)\right) \\
&\equiv \operatorname*{argmin}_{w \in \mathbb{R}^d} \sum_{i=1}^n \left[-\log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) + \frac{(w^T x^i - y^i)^2}{2\sigma^2}\right].
\end{aligned}
$$

- Notice that constant doesn't depend on w so doesn't change argmin,

$$
\equiv \operatorname*{argmin}_{w \in \mathbb{R}^d} \frac{1}{2\sigma^2} \sum_{i=1}^n (w^T x^i - y^i)^2 \equiv \underbrace{\operatorname*{argmin}_{w \in \mathbb{R}^d} \frac{1}{2} \sum_{i=1}^n (w^T x^i - y^i)^2}_{\text{least squares}},
$$

where we note that $\sigma > 0$ doesn't change argmin.

Maximum Likelihood Estimation and Overfitting

- In our abstract setting with data D the MLE is

$$h^* \in \underset{h \in \mathcal{H}}{\operatorname{argmax}}\, p(D|h).$$

- But conceptually MLE is a bit weird:
  - "Find the h that makes D have the highest probability given h".
- And MLE often leads to overfitting:
  - Data could be very likely in some very unlikely model from family.
  - For example, a complex model overfits by memorizing the data.
- What we really want:
  - "Find the h that is the most likely given the data D"'.

Maximum a Posteriori (MAP) Estimation

- Maximum a posteriori (MAP) estimate maximizes the reverse probability,

$$h^* \in \underset{h \in \mathcal{H}}{\operatorname{argmax}}\, p(h|D).$$

- This is what we want: the probability of h given our data.
- MLE and MAP are connected by Bayes' rule,

$$\underbrace{p(h|D)}_{\text{posterior}} = \frac{p(D|h)p(h)}{p(D)} \propto \underbrace{p(D|h)}_{\text{likelihood}}\underbrace{p(h)}_{\text{prior}}.$$

- So MAP maximizes the likelihood $p(D|h)$ times the prior $p(h)$:
    - Prior is our "belief" that h is correct model before seeing data.
    - Prior can reflect that complex models are likely to overfit.

MAP Estimation and Regularization

- From Bayes rule the MAP estimate with IID examples D_i is

$$h^* \in \underset{h \in \mathcal{H}}{\text{argmax}}\, p(h|D) \equiv \underset{h \in \mathcal{H}}{\text{argmax}} \prod_{i=1}^{n} [p(D|h)]\, p(h).$$

- By again taking the negative logarithm we get

$$h^* \in \underset{h \in \mathcal{H}}{\text{argmin}} \sum_{i=1}^{n} \underbrace{-\log p(D_i|h)}_{\text{loss}} \underbrace{-\log p(h)}_{\text{regularizer}},$$

so we can view the negative log-prior as a regularizer.

  - Many regularizers are equivalent to negative log-priors.

L2-Regularization and MAP Estimation

- We obtain L2-regularization under an independent Gaussian assumption,
$$w_j \sim \mathcal{N}(0, 1/\lambda).$$
- This implies that
$$p(w) = \prod_{j=1}^d p(w_j|\lambda) \propto \prod_{j=1}^d \exp\left(-\frac{\lambda}{2}w_j^2\right) = \exp\left(-\frac{\lambda}{2}\sum_{j=1}^d w_j^2\right).$$
so we have that
$$-\log p(w) = -\log\exp\left(-\frac{\lambda}{2}\|w\|^2\right) = \frac{\lambda}{2}\|w\|^2.$$
- So the MAP estimate with IID training examples would be
$$w^* \in \underset{w\in\mathbb{R}^d}{\text{argmin}} -\log p(y|X,w) - \log p(w) \equiv \underset{w\in\mathbb{R}^d}{\text{argmin}} \sum_{i=1}^n -\log p(y^i|x^i,w) + \frac{\lambda}{2}\|w\|^2.$$

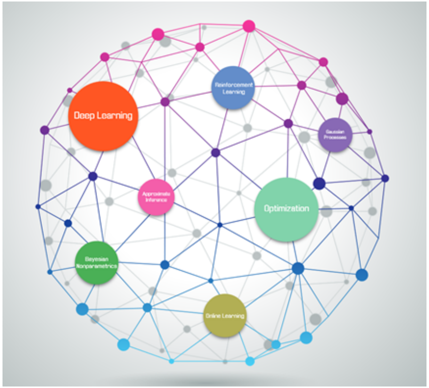
MAP Estimation Perspective

- Many of our loss functions and regularizers have probabilistic interpretations.
  - For example, Laplace likelihood leads to absolute error and L1-regularization.
- Probabilitic interpretation lets us define regression losses in non-standard settings:
  - Multi-label y^i.
  - Multi-class y^i.
  - Ordinal y^i.
  - Count y^i.
  - Survival time y^i.

Outline

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 optimization:
  - Numerically optimize parameters w of model, given data.
- There are some exceptions:
  1. Counting- and distance-based methods (KNN, random forests).
     - See CPSC 340.
  2. Integration-based methods (Bayesian learning).
     - Later in course.

  Although you still need to tune parameters in those models.
- But why study optimization? Can't I just use optimization libraries?
  - "\", linprog, quadprog, fminunc, fmincon, CVX, 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.

- 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.
  - But first we'll show how to use some "off the shelf" optimization methods.

Robust Regression in Matrix Notation

- Regression with the absolute error as the loss,

$$f(w) = \sum_{i=1}^{n} |w^T x^i - y^i|.$$

- In CPSC 340 we argued that this is more robust to outliers.
- We can write this in matrix notation as

$$f(w) = \|Xw - y\|_1.$$

where recall that the L1-norm of a vector r of length n is

$$\|r\|_1 = \sum_{i=1}^{n} |r_i|.$$

- This objective is not quadratic, but can be minimized as a linear program.
  - Minimizing a linear function with linear constraints.

Robust Regression as a Linear Program

- L1-norm regression in summation notation,

$$\operatorname*{argmin}_{w\in\mathbb{R}^d} \sum_{i=1}^{n} |w^T x^i - y^i|.$$

- Re-write absolute value using $|\alpha| = \max\{\alpha, -\alpha\}$,

$$\operatorname*{argmin}_{w\in\mathbb{R}^d} \sum_{i=1}^{n} \max\{w^T x^i - y^i, y^i - w^T x^i\}.$$

- Introduce n variables r_i that upper bound the max functions,

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

- This is a linear objective with non-linear constraints.
- Note that we have $r_i = |w^T x^i - y^i|$ at the solution.
  - Otherwise, either the constraints are violated or we could decrese r_i.

Robust Regression as a Linear Program

- L1-norm regression in summation notation,

$$\operatorname*{argmin}_{w\in\mathbb{R}^d} \sum_{i=1}^{n} |w^T x^i - y^i|.$$

- Re-write absolute value using $|\alpha| = \max\{\alpha, -\alpha\}$,

$$\operatorname*{argmin}_{w\in\mathbb{R}^d} \sum_{i=1}^{n} \max\{w^T x^i - y^i, y^i - w^T x^i\}.$$

- Introduce n variables r_i that upper bound the max functions,

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

- Having r_i bound the max is equivalent to r_i bounding max arguments.

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

Robust Regression as a Linear Program

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

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

  or in matrix notation as

$$\underset{w \in \mathbb{R}^d,\, r \in \mathbb{R}^n}{\text{argmin}} \; 1^T r, \quad \text{with} \quad r \geq Xw - y,\; r \geq y - Xw,$$

  where 1 is a vector containinng all ones and inequalities are element-wise.
- For medium-sized problems, we can solve this with Matlab's *linprog*.

Minimizing Absolute Values and Maxes

- A general approach for minimizing absolute values and/or maximums:
  1. Introduce maximums over linear functions to replace minimizing absolute values.
  2. Introduce new variables that are constrained to bound the maximums.
  3. Transform to linear constraints by splitting the maximum constraints.

- For example, we can write minimizing support vector machine (SVM) objective,

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

as a quadratic program (quadratic objective with linear constraints).

Support Vector Machine as a Quadratic Program

- The SVM optimization problem is

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

- Introduce new variables to upper-bound the maxes,

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

- Split the maxes into separate constraints,

$$\operatorname*{argmin}_{w \in \mathbb{R}^d, r \in \mathbb{R}^n} 1^T r + \frac{\lambda}{2}\|w\|^2, \quad \text{with} \quad r \geq 0,\ r \geq YXw,$$

where Y is a diagonal matrix with the y^i values along the diagonal.
  - This means YX is X with each row scaled by the corresponding y^i.

Outline

General Lp-norm Losses

- Consider minimizing the regression loss

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

where $\|\cdot\|_p$ is a general Lp-norm,

$$\|r\|_p = \left(\sum_{i=1}^{n} |r_i|^p\right)^{\frac{1}{p}}.$$

- Recall the three properties of norms:
  1. $\|r\|_p = 0$ iff $r = 0$,
  2. $\|\theta r\| = |\theta| \cdot \|r\|$ for a scalar θ, (absolute homogeneity)
  3. $\|r + u\| \leq \|r\| + \|u\|$, (triangle inequality)

  and that these imply norms are non-negative.

General Lp-norm Losses

- Consider minimizing the regression loss

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

where $\|\cdot\|_p$ is 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 using linear algebra.
  - By non-negativity, squaring it doesn't change the argmax.
- With $p = 1$, we can minimize the function using linear programming.
- With $p = \infty$, we can also use linear programming.
- For $1 < p < \infty$, we can use gradient descent (next lecture).
  - It's smooth once raise to the power p.

- If we use $p < 1$ (which is not a norm), minimizing f is NP-hard.

Convex Functions

- With $p \geq 1$ the problem is convex, while with $p < 1$ the problem is non-convex.
- Convexity is usually a good indicator of tractability:
  - Minimizing convex functions is usually easy.
  - Minimizing non-convex functions is usually hard.
- Existing software (like CVX) minimizes a wide variety of convex functions.
- To define convex functions, we first need the notion of a convex combination:
  - A convex combination of two variables w and v is given by

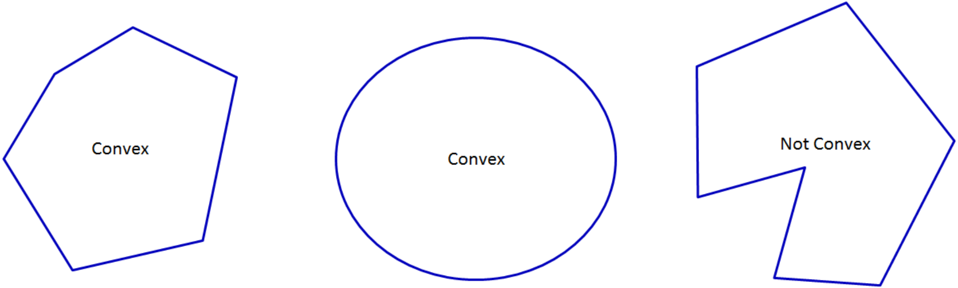
$$\theta w + (1-\theta)v \quad \text{for any} \quad 0 \leq \theta \leq 1.$$

  - A convex combination of k variables $\{w_1, w_2, \dots, w_k\}$ is given by

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

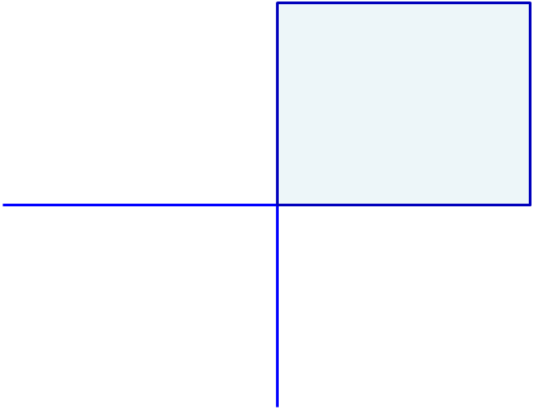
Convex Sets

- The domain of a convex function must be a convex set:
  - 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 $\theta w + (1-\theta)v \in \mathcal{C}$ for $0 \leq \theta \leq 1$.



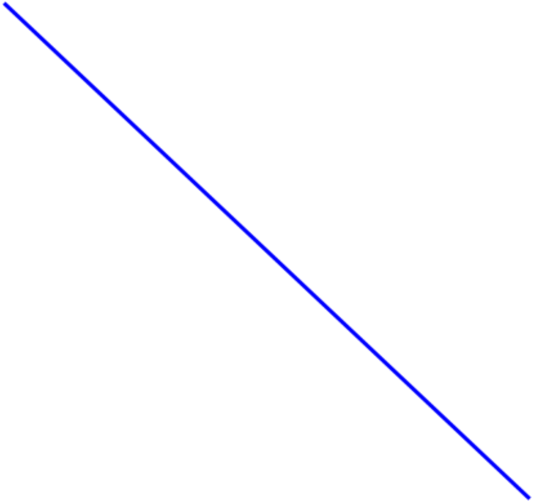
Examples of Simple Convex Sets

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



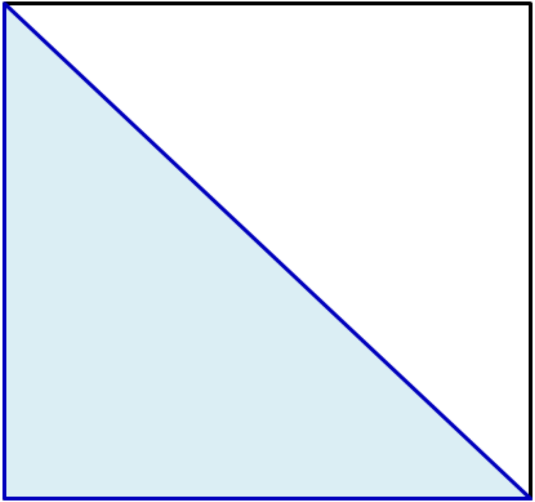
Examples of Simple Convex Sets

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



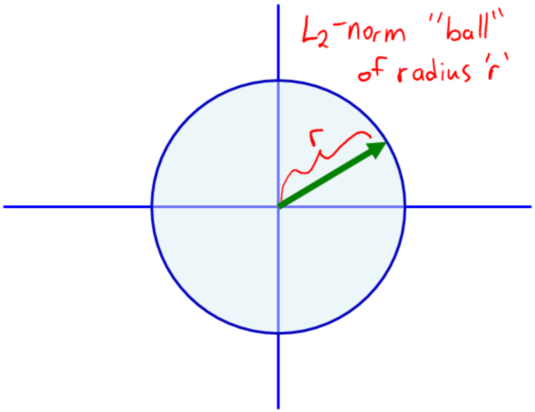
Examples of Simple Convex Sets

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



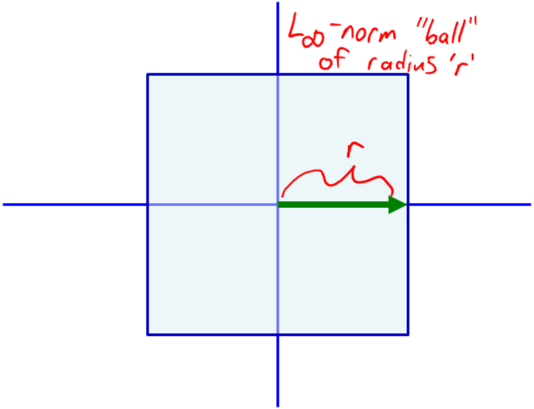
Examples of Simple Convex Sets

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



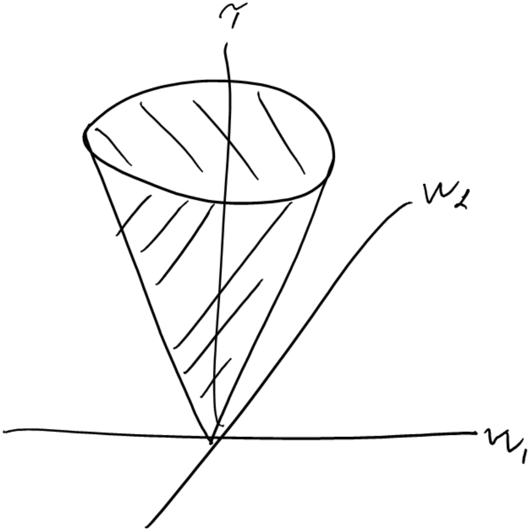
Examples of Simple Convex Sets

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



Showing a Set is Convex from Defintion

- We can prove convexity of a set from the definition:
  - Choose a generic w and v in $\mathcal{C}$, show that generic u between them is in the set.
- Hyper-plane example: $\mathcal{C} = \{w \mid a^T w = b\}$.
  - If $w \in \mathcal{C}$ and $v \in \mathcal{C}$, then we have $a^T w = b$ and $a^T v = b$.
  - To show $\mathcal{C}$ is convex, we can show that $a^T u = b$ for u between w and v.

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

Showing a Set is Convex from Defintion

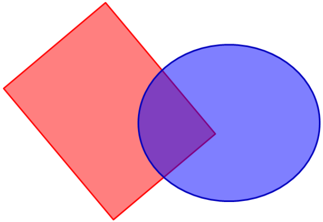
- We can prove convexity of a set from the definition:
  - Choose a generic w and v in $\mathcal{C}$, show that generic u between them is in the set.

- Norm-ball example: $\mathcal{C} = \{w \mid \|w\|_p \leq 10\}$.
  - If $w \in \mathcal{C}$ and $v \in \mathcal{C}$, then we have $\|w\|_p \leq 10$ and $\|v\|_p \leq 10$.
  - To show $\mathcal{C}$ is convex, we can show that $\|u\|_p \leq 10$ for u between w and v.

$$\begin{aligned}
\|u\|_p &= \|\theta w + (1-\theta)v\|_p \\
&\leq \|\theta w\|_p + \|(1-\theta)v\|_p && \text{(triangle inequality)} \\
&= |\theta| \cdot \|w\|_p + |1-\theta| \cdot \|v\|_p && \text{(absolute homogeneity)} \\
&= \theta\|w\|_p + (1-\theta)\|v\|_p && (0 \leq \theta \leq 1) \\
&\leq \theta 10 + (1-\theta)10 = 10.
\end{aligned}$$

Showing a Set is Convex from Intersections

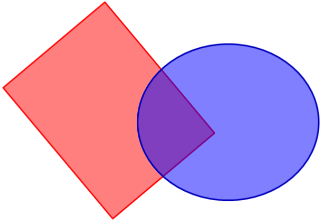
- The intersection of convex sets is convex.
  - Proof is trivial: convex combinations in the intersection are in the intersection.



- We can prove convexity of a set by showing it's an intersection of convex sets.
- Example: $\{w | a^T w = b, \|w\|_p \leq 10\}$ is convex.
  - It's the intersection of our two previous examples.

Showing a Set is Convex from Intersections

- The intersection of convex sets is convex.
  - Proof is trivial: convex combinations in the intersection are in the intersection.



- We can prove convexity of a set but showing it's an intersection of convex sets.

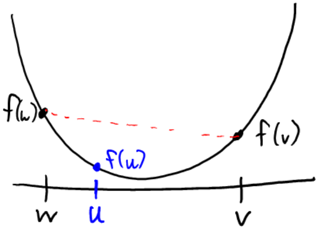
- Example: the w satisfying linear constraints form a convex set:

$$
\begin{aligned}
Aw &\leq b \\
A_{\text{eq}}w &= b_{\text{eq}} \\
LB \leq & w \leq UB.
\end{aligned}
$$

Convex Functions

- Two equivalent defintions of aconvex function:
  1. Area above the function is a convex set.
  2. The function is always below the "chord" between two points.

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



- Implications: all local minima are global minima.
- We can globally minimize a convex function by finding *any* stationary point.

Examples of Convex Functions

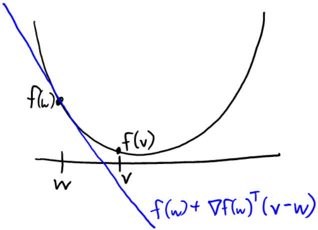
- 1D quadratic: $aw^2 + bw + c$, when $a > 0$.
- Linear: $a^T w + b$.
- Exponential: $\exp(aw)$.
- Negative logarithm: $-\log(w)$.
- Absolute value: $|w|$.
- Max: $\max_i\{w_i\}$.
- Negative entropy: $w \log w$, for $w > 0$.
- Logistic loss: $\log(1 + \exp(-w))$.
- Log-sum-exp: $\log(\sum_i \exp(w))$.

Differentiable Convex Functions

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

- For differentiable convex functions, there is third equivalent definiton:
  - A *differentiable* f is convex iff f is always above tangent.

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



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

Twice-Differentiable Convex Functions

- For twice-differentiable convex functions, there is a fourth equivalent definition:
  - A *twice-differentiable* f is convex iff f is curved upwards everywhere.
- For univariate functions, this means $f''(w) \geq 0$ for all w.
  - Usually the easiest way to show a twice-differentiable f is convex.
- For multivariate functions, means the Hessian is positive semi-definite for all w,

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

meaning that $v^T \nabla^2 f(w) v \geq 0$ for all w and v.

Convexity and Least Squares

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

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

- Using results from last time we have

$$\nabla^2 f(w) = X^T X = \sum_{i=1}^{n} x^i (x^i)^T$$

- So we want to show that $X^T X \succeq 0$ or equivalently that $v^T X^T X v \geq 0$ for all v.
- We did this last time in matrix notation, let's do it in summation notation:

$$v^T \left(\sum_{i=1}^{n} x^i (x^i)^T\right) v = \sum_{i=1}^{n} v^T x_i (x_i)^T v = \sum_{i=1}^{n} (v^T x_i) \left((x_i)^T v\right) = \sum_{i=1}^{n} (v^T x^i)^2 \geq 0,$$

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

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.

- But note that composition $f(g(w))$ 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,

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

- Second term has a Hessian of λI so is convex because $\lambda I \succeq 0$.
- 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.

Summary

- MLE and MAP estimation give probabilistic interpretation to losses/regularizers.
- Converting non-smooth problems involving max to constrained smooth problems.
- Convex functions are special functions where all stationary points are global minima.
- Showing functions are convex from definitions or convexity-preserving operations.

- How do we solve "large-scale" problems?

Bonus Slide: $\propto$ Probability Notation

- When we write
$$p(y) \propto f(y),$$
we mean that
$$p(y) = \kappa f(y),$$
where κ is the number need to make p a probability.
- If y is discrete taking values in $\mathcal{Y}$,
$$\kappa = \frac{1}{\sum_{y \in \mathcal{Y}} f(y)}.$$
- If y is continuous taking values in $\mathcal{Y}$,
$$\kappa = \frac{1}{\int_{y \in \mathcal{Y}} f(y)}.$$
- Return to Gaussian MLE.