Lecture 5

Least Squares and Optimization in Machine Learning

SWCON253, Machine Learning

Won Hee Lee, PhD

Learning Goals

- Understand the fundamental concepts of **least squares and optimization** in machine learning

Given

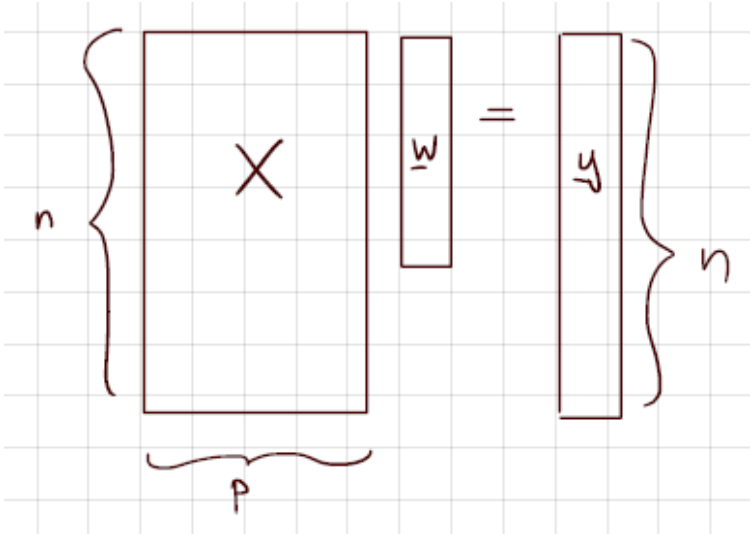
Labels $\underline{y} \in \mathbb{R}^n$ for n training samples

Features $X \in \mathbb{R}^{nxp}$ (p features)

We assume that n≥p, rank (X) = p (X has p linearly independent columns)

We want to find $\underline{w}$ so that $\underline{\hat{y}} = X\underline{w} \approx \underline{y}$

Pictorially,



In classification,

$y_i \in \{-1, +1\}$

$$\underline{\hat{w}} = \underset{\underline{w}}{\operatorname{argmin}} \left\| \underline{y} - X\underline{w} \right\|_2^2$$

We will compute the weight vector $\underline{\hat{w}}$ that minimizes the sum of squared errors.

So, we want to minimize the distance between $\underline{y}$ and $X\underline{w}$ that we predict using the linear model.

$$\underline{\hat{w}} = (X^T X)^{-1} X^T \underline{y}$$

The question is how do we use this within a classification setting.

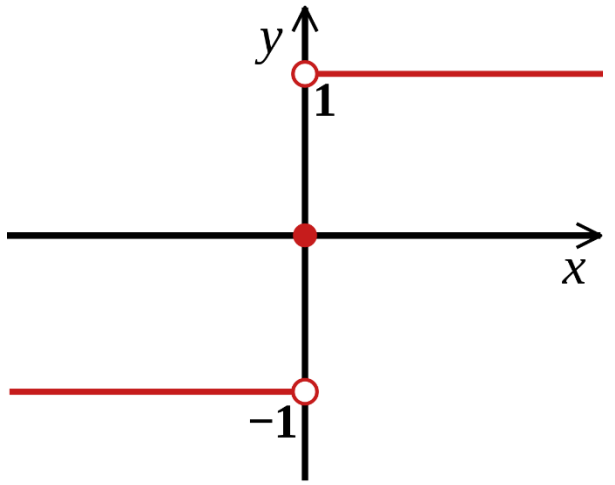
Let $\underline{\hat{y}} = X\underline{\hat{w}}$.

The elements of $\underline{\hat{y}}$ are not $\in \{-1, +1\}$

That is, we are not getting a bunch of plus and minus ones here. We are actually getting real numbers. For example, for one sample, $\underline{y}$ might be 0.5, for another sample it might be -0.7.

We have to come up with a **classification rule**.

Classification rule



We predict +1 if $\hat{y}_i > 0$ and -1 if $\hat{y}_i < 0$

In other words, $\tilde{y}_i = sign(\hat{y}_i) \in \{-1, +1\}$

We can actually see not only if $\hat{y}_i$ is close to y, but also whether the labels $\tilde{y}_i$ that we're predicting equal the labels that we actually observed.

For a new sample $x_{new} \in \mathbb{R}^p$, we want to predict its label (new unknown label)

$$\hat{y}_{new} = \langle \underline{\hat{w}}, \underline{x}_{new} \rangle$$

$\hat{y}_{new}$ is the inner product of the feature for this new sample and the weight vector that we estimated using our training data.

We use our training data to find a good weight vector and then we can now use that weight vector with our new feature to make new predictions.

$$\hat{y}_{new} = \langle \underline{\hat{w}}, \underline{x}_{new} \rangle$$

$$\tilde{y}_{new} = sign(\hat{y}_{new})$$

Overall, this is a pipeline of the classification system.

- It takes all the training data.
- We try to learn a good weight factor in the context of this linear model and then
- we take a new sample and
- we plugged the learned the weight vector $\underline{w}$ that we computed with our training data into the linear model and
- use that to come up with a predictive label.

$$\underline{\hat{w}} = \underset{\underline{w}}{\operatorname{argmin}} \left\| \underline{y} - X\underline{w} \right\|_2^2$$

$$\underline{\hat{w}} = (X^T X)^{-1} X^T \underline{y}$$

What we said is $\underline{\hat{w}}$ is the value w that minimizes the length $\underline{y}$ minus X$\underline{w}$ (2-norm squared)

We have arrived this equation for $\underline{\hat{w}}$ using this geometric argument.

What we would like to do today is to derive that same expression from different way by thinking about it as an **optimization** problem, instead of thinking about it geometrically.

Optimization approach

Recall

$$\|\underline{x}\|_2^2 = \sum_{i=1}^{n} x_i{}^2 = \underline{x}^T\underline{x} = \langle \underline{x}, \underline{x} \rangle$$

$$\underline{\hat{w}} = \underset{\underline{w}}{\operatorname{argmin}} \left\| \underline{y} - X\underline{w} \right\|_2^2$$

$$= \underset{\underline{w}}{\operatorname{argmin}} (\underline{y} - X\underline{w})^T (\underline{y} - X\underline{w})$$

$$= \underset{\underline{w}}{\operatorname{argmin}}\, \underline{y}^T\underline{y} - (X\underline{w})^T\underline{y} - \underline{y}^T X\underline{w} + (X\underline{w})^T (X\underline{w})$$

$$= \underset{\underline{w}}{\operatorname{argmin}}\, \underline{y}^T\underline{y} - \underline{w}^T X^T\underline{y} - \underline{y}^T X\underline{w} + \underline{w}^T X^T X\underline{w}$$

$$= \underset{\underline{w}}{\operatorname{argmin}}\, \underline{y}^T\underline{y} - 2\underline{w}^T X^T\underline{y} + \underline{w}^T X^T X\underline{w}$$

$$\underline{w}^T X^T\underline{y} = (\underline{y}^T X\underline{w})^T$$

Optimization approach

$$\underline{\hat{w}} = \underset{\underline{w}}{\operatorname{argmin}} \left\| \underline{y} - X\underline{w} \right\|_2^2$$

$$= \underset{\underline{w}}{\operatorname{argmin}}\, \underline{y}^T \underline{y} - 2\underline{w}^T X^T y + \underline{w}^T X^T X \underline{w}$$

- We started off by saying we wanted to minimize the sum of squared errors or sum of squared residuals and
- We have taken that objective function we want to minimize and we've rewritten it as this long matrix vector equation and
- Now what we want to do is we want figure out how to actually solve this optimization problem.

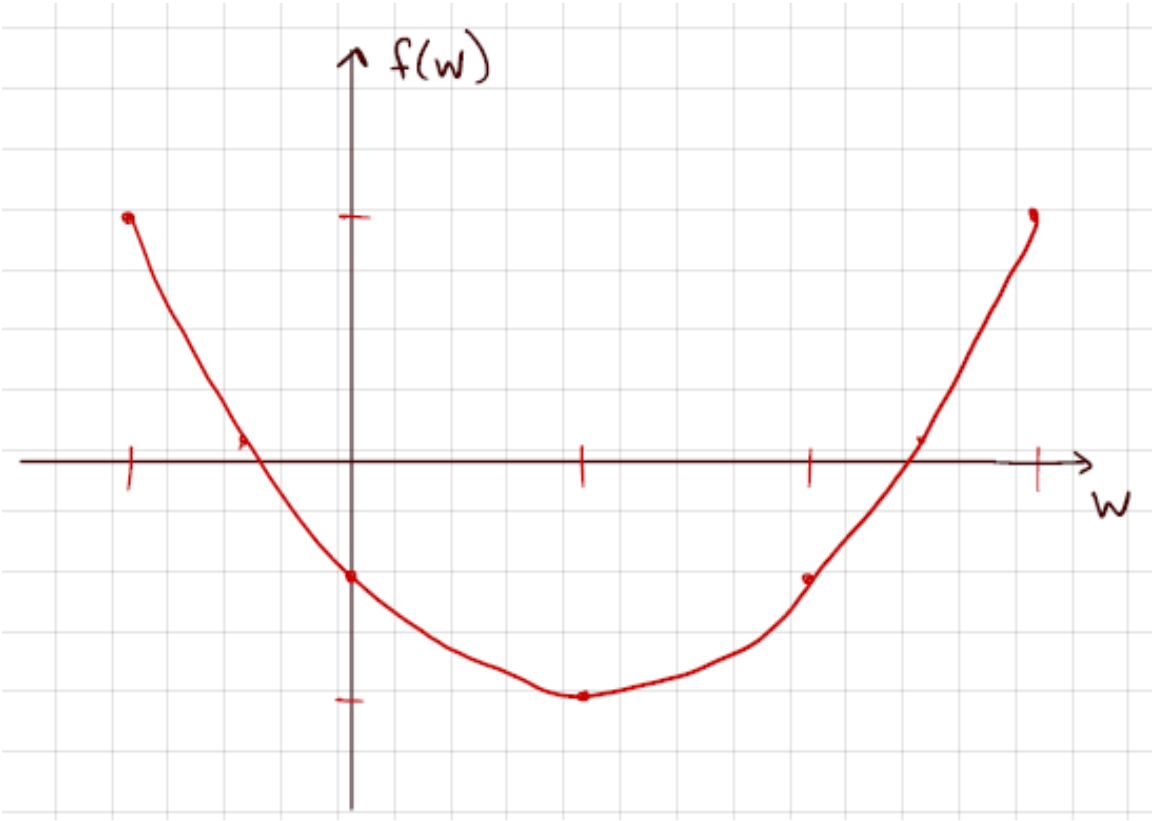
Warmup (simple optimization problem in 1D)

$$\mathrm{f(w)} = \frac{1}{2}w^2 - w - \frac{1}{2}$$

$$\hat{w} = \underset{w}{\operatorname{argmin}}\, f(w)$$

$$\frac{df}{dw} = w - 1 = 0$$

$$\hat{w} = 1$$



We take the derivative and set it equal to zero. That is going to give us the minimum.

So now we want to know is how can we actually go about solving optimization problems where instead of optimizing over scaler/scalar numbers like a scale w, we want to optimize over weight vectors

When we look at the geometric perspective, we said it was really important that x^Tx is invertible, and that's how we came up with our solution which is the function of x^Tx,

So something similar shows up to when we think about the optimization perspective as well.

Positive Definite Matrices

From the geometric perspective, we saw that it was important for finding a unique least squares solution $\underline{\hat{w}}$ that X^TX be invertible.

Is this important in the optimization setting as well? Yes!

The following two things are equivalent for $X \in \mathbb{R}^{nxp}$ with n ≥ p, rank(X) = p

$X^TX \in \mathbb{R}^{nxp}$ is invertible (in other words, $(X^TX)^{-1}$ exists)

X^TX is positive definite

A matrix Q is positive definite (p.d.) if

$\underline{x}^T Q \underline{x} > 0$ for all $\underline{x} \neq 0$ (shorthand: $Q \succ 0$) Q curly greater than zero

A matrix Q is positive semi-definite (p.s.d.) if

$\underline{x}^T Q \underline{x} \geq 0$ for all $\underline{x} \neq 0$ (shorthand: $Q \succcurlyeq 0$) Q curly greater than or equal to zero

Ex 1. 1D

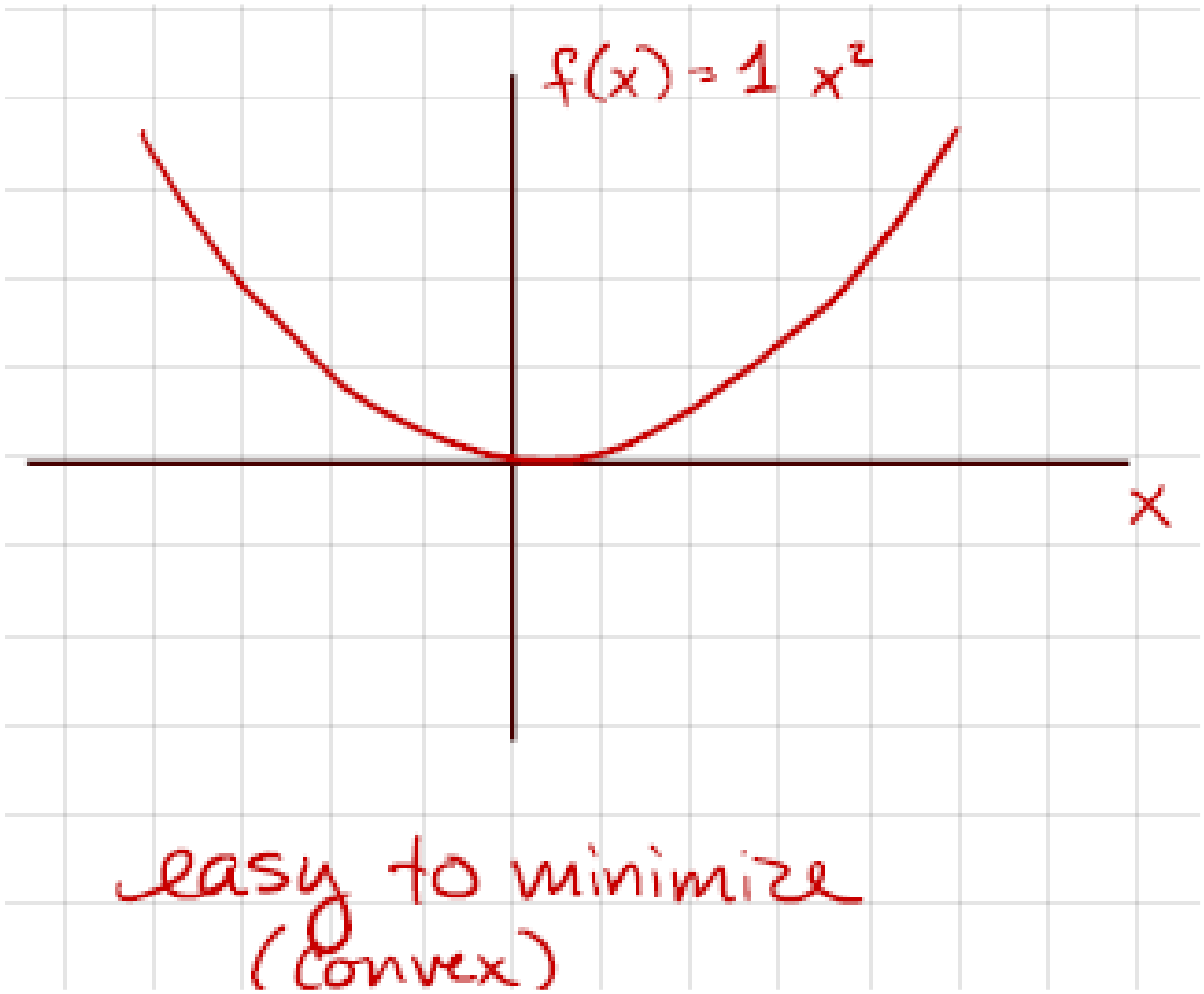
$x \in \mathbb{R}^n$, $Q \in \mathbb{R}^n$

$x^TQx = Qx^2$

When is this Qx^2 greater than 0?

$Qx^2 > 0$ if $Q > 0$

Imagine minimizing $f(x)=Qx^2$

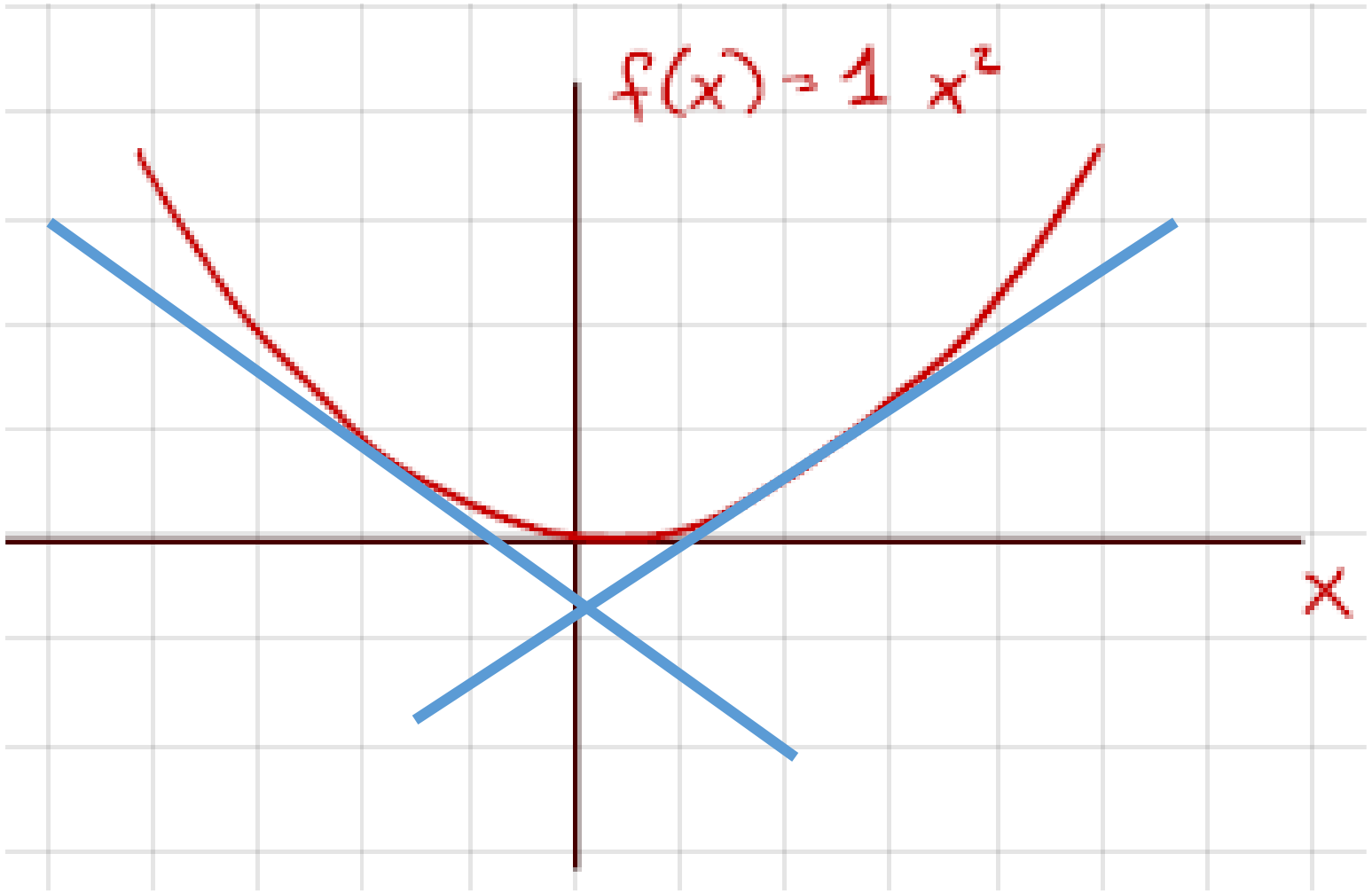


Easy to minimize
(Convex)
Bowl shaped function

Convexity
For any point on the curve,
Can compute its tangent

If we look at its tangent, the function is above the tangent

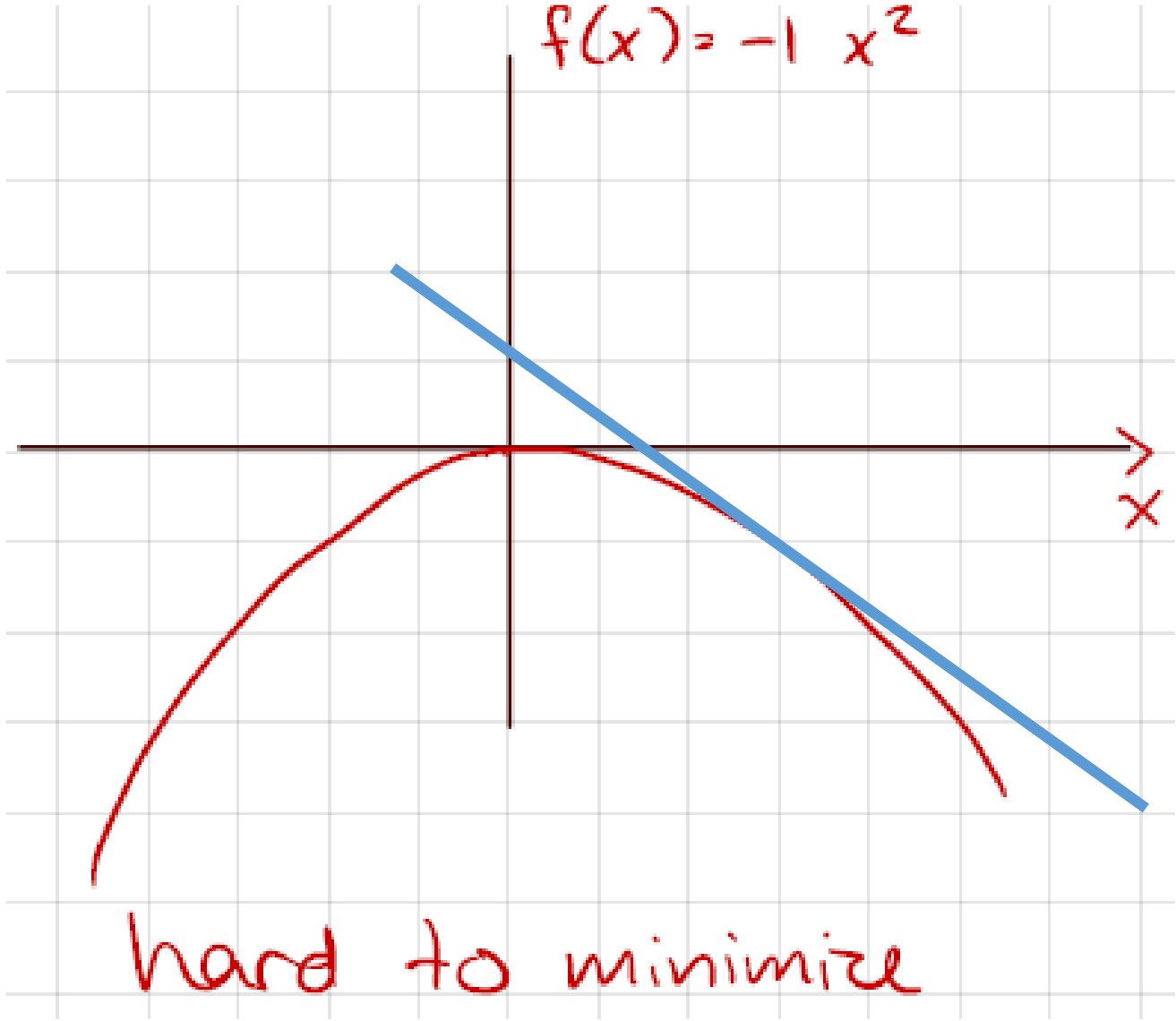
Optimization theory about how to actually find minimizers of convex functions



$Q < 0$

Hard to minimize

(non-convex)



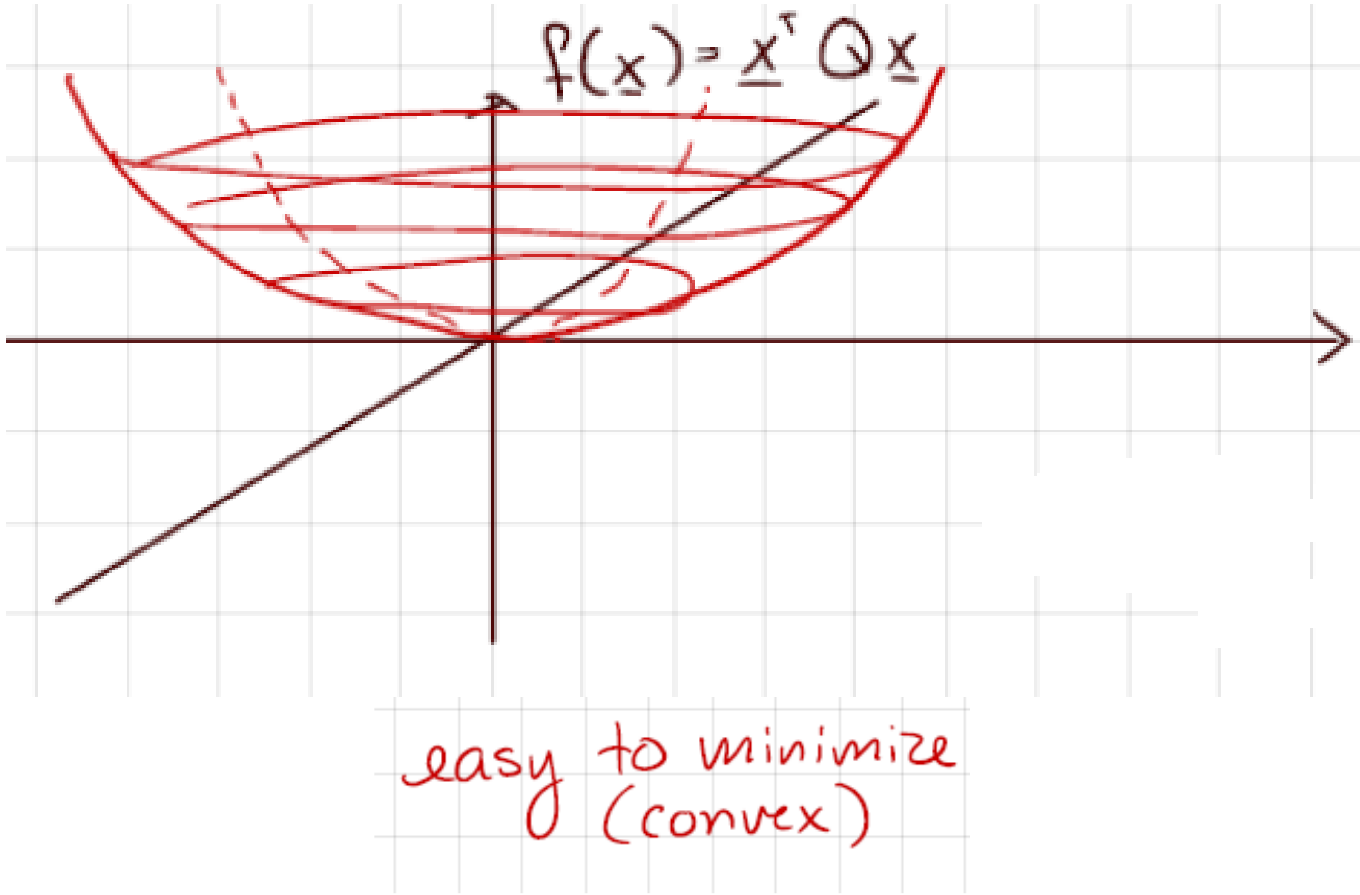
Ex 2.

$\underline{x} \in \mathbb{R}^2$, $Q \in \mathbb{R}^{2x2}$

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

We try to minimize $f(x) = \underline{x}^T Q \underline{x}$

$f(x) = x_1^2 + x_2^2 > 0$ for all $\underline{x} \neq 0$



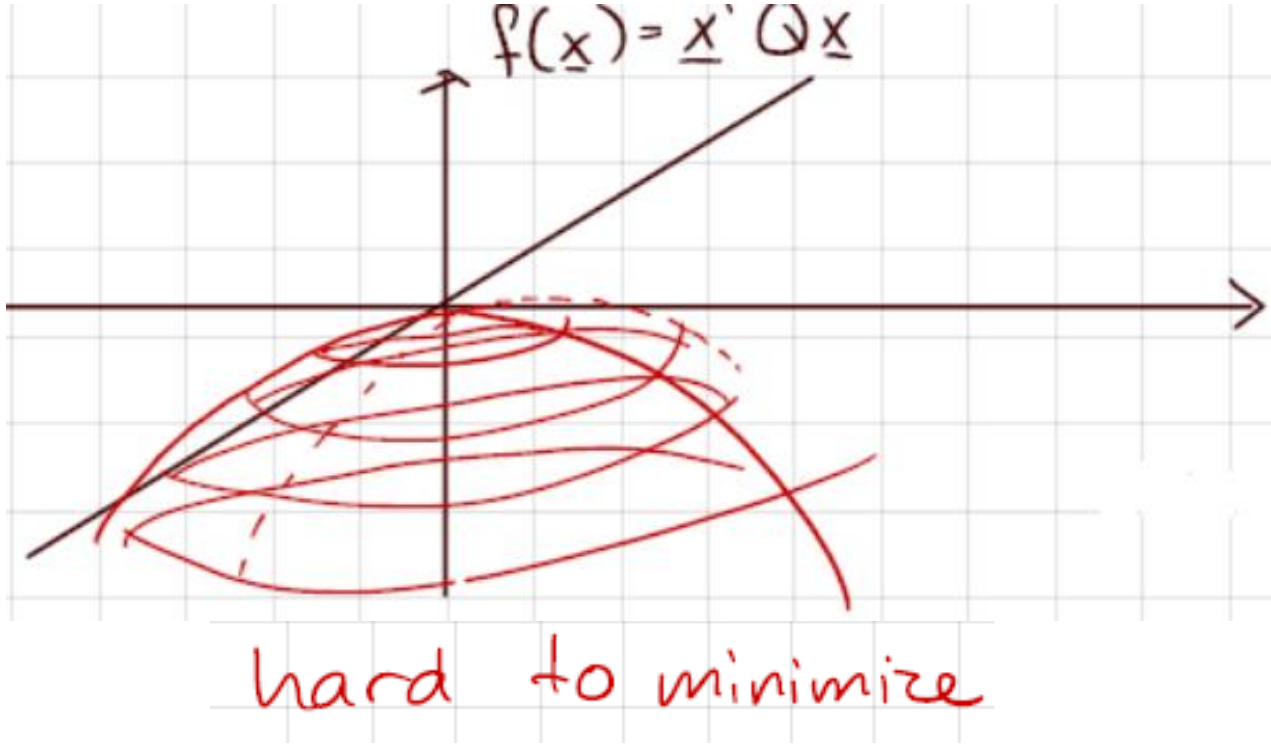
Ex 2.

$\underline{x} \in \mathbb{R}^2$, $Q \in \mathbb{R}^{2x2}$

$$Q = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

$f(x) = \underline{x}^T Q \underline{x}$

$f(x) = -x_1^2 - x_2^2 < 0$ for all $\underline{x} \neq 0$



So thinking about **positive definite matrices** gives us a lens into thinking about how we will solve this **optimization problem** associated with **least squares** and when we will be able to find a good solution versus when finding a good solution is hard.

Properties of Positive Definite Matrices

1. If $P \succ 0$ and $Q \succ 0$, then $P+Q \succ 0$

   $\underline{x}^T P \underline{x} > 0$ & $\underline{x}^T Q \underline{x} > 0 \rightarrow \underline{x}^T (P+Q) \underline{x} > 0 = \underline{x}^T P x + \underline{x}^T Q \underline{x} > 0$

2. If $Q \succ 0$ and $a>0$, then $aQ \succ 0$

   $\underline{x}^T Q \underline{x} > 0 \rightarrow \underline{x}^T (aQ) \underline{x} > 0 = a \underline{x}^T Q \underline{x} > 0$

3. For any A, $A^T A \succcurlyeq 0$ and $A A^T \succcurlyeq 0$

4. If $A \succcurlyeq 0$, then A^{-1} exists

5. $Q \succcurlyeq P$ means $Q - P \succcurlyeq 0$

Recall **Optimization Formulation**

$$\underline{\hat{w}} = \underset{\underline{w}}{\operatorname{argmin}}\, \underline{y}^T \underline{y} - 2\underline{w}^T X^T y + \underline{w}^T X^T X \underline{w}$$

We try to solve this optimization problem.

From property 3 of positive definite matrices, X^TX is positive semi-definite

We assume

$X^TX \succ 0 \rightarrow f(w) = \underline{y}^T \underline{y} - 2\underline{w}^T X^T y + \underline{w}^T X^T X w$

$\rightarrow$ this function is convex

$\rightarrow$ compute "derivative" & set to zero to find minimizer

Only we're not really dealing with scalars anymore.

We have vectors. So instead of derivative, use **gradients**

When w is a scalar, we set derivative $\frac{df}{dw}$ to zero and solve for w.

When $\underline{w}$ is a vector, we set gradient $\nabla_w f$ to zero and solve for $\underline{w}$.

$$\nabla_{\underline{w}} f = \begin{bmatrix} \frac{df}{dw_1} \\ \frac{df}{dw_2} \\ \vdots \\ \frac{df}{dw_p} \end{bmatrix}$$

with respect to $\underline{w}$

Ex 1.

$$f(\underline{w}) = \underline{c}^T \underline{w} = c_1 w_1 + c_2 w_2 + \ldots + c_p w_p$$

$$\nabla_w f =$$

$$\nabla_w f = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_p \end{bmatrix} = \underline{c}$$

Ex 2.

$$f(\underline{w}) = \underline{w}^T \underline{w} = \|\underline{w}\|_2^2 = w_1^2 + w_2^2 \ \ldots + w_p^2$$

$$\nabla_w f =$$

$$\nabla_w f = \begin{bmatrix} 2w_1 \\ 2w_2 \\ \vdots \\ 2w_p \end{bmatrix} = 2\underline{w}$$

Ex 3.

$$f(\underline{w}) = \underline{w}^T Q \underline{w}$$

$$= \sum_{i=1}^{p} \sum_{j=1}^{p} w_i Q_{ij} w_j$$

$$\frac{d(w_i Q_{ij} w_j)}{dw_k} = \begin{cases} if\ k = i = j \\ if\ i = k \neq j \\ if\ i \neq k = j \\ if\ k \neq i, k \neq j \end{cases}$$

Let's compute the derivative of just one term in the sum before the whole sum

To figure out the gradient, next we have to figure out the derivative of function f

$$\frac{df}{dw_k}=\sum_{i=1}^{p}\sum_{j=1}^{p}\frac{d(w_iQ_{ij}w_j)}{dw_k}$$

$$\nabla_w f = Qw + Q^T w$$

If Q is symmetric (i.e., $Q=Q^T$), then

$$\nabla_w f = 2Qw$$

Ex. **Least Squares**

$$f(\underline{w}) = \underline{y}^T\underline{y} - 2\underline{w}^T X^T y + \underline{w}^T X^T X\underline{w}$$

$$\nabla_w f = 0 - 2X^T y + 2X^T X\underline{w}$$

We can set $\nabla_w f = 0 - 2X^T y + 2X^T X\underline{w} = 0$

$\rightarrow X^T X\underline{w} = X^T\underline{y}$ because X^TX is positive definite, we take its inverse

$$\underline{\hat{w}} = (X^T X)^{-1} X^T \underline{y}$$

What we got is the same as the one from geometric perspective!

(based on Ex 2. $c = -2X^Ty$ and Ex 3. $Q=X^TX = Q^T$)

This time, we arrived at $\underline{\hat{w}} = (X^T X)^{-1} X^T \underline{y}$ in a completely different way using these **optimization arguments**

computing gradients of objective functions and

setting them equal to zero and

we saw how using notions of **positive definiteness**

allowed us to try to understand why having $X^T X$ being invertible is important.

Further Readings

- Mathematics for Machine Learning (MathML)
  - Chapters 2 & 7: Linear Algebra & Continuous Optimization

- Any linear algebra & optimization books should be fine!

Announcements

- Homework #1
  - 스스로 복습한 내용을 A4용지에 손글씨로 작성/스캔하여 제출
  - 1-page 이상 per a lecture (lectures 3-5)
  - **Due March 23th Tuesday at 11:59 pm**