EECS 127 Lecture Notes

Authored by Dun-Ming Huang, SID: 303******2

0.1 Preface

This section is directly copy-pasted from my GitHub repo for the note.

eecs127 moment.
Hi I'm Brandon. You might know me from work. If not, I think you should be glad you don't, or you would have had to deal with my EECS 16A Puns.
I update this note after every lecture and before exams (which I will notify about if I run a revision, or you can watch the repository).
Since I make these lecture notes by transcribing lecture contents on the fly, you would probably expect immature pedagogical formatting (which immediately appears) and typos in these efforts.

0.1.1 First of all, My Notes are Not Substitutes to the Course Readers

My notes are my own transcriptions for EECS 127 lecture notes. On a professional note, yes, you may use these for personal purposes and gaining extra insights on in-course contexts, but this is by no means a pedagogical substitute of the original course notes, even if it describes all the concepts of the course reader.

On a professional note, once again, the official course readers are structured in a pedagogical imperative: it connects concepts to optimize student understanding and the coherency of course content (even if it might have been confusing on the first reads). This course note does not. I was not obligated to connect concepts. I was only writing the work to organize information for my own sake. This work is not specifically written for the public just like my CSM resources were. My notes are by no means a substitute of EECS 127 readers.

But, thinking from the other angle, we may consider these notes I produced as a complementary resource for EECS 127.

0.1.2 How did Brandon Use This Note?

Great Question. I'm glad you asked. Cyrus.

I used this note extensively to lookup summaries of concepts when writing homeworks and mock exams, as well as to force myself to read through derivations on lecture notes via transcribing them onto LaTeX and running a rigorous revision later. I have also shared my notes to others (as you see, it is hosted as an open source project on Github), in the hope that they can use these notes as a lookup for summaries.

You can expect to see more formal course contents starting at the later half of Note 1.

0.1.3 What is the Last Four Digits of Your Social Security Number?

gXcQ

Contents

Chapter 1

Introduction and Least Squares

For this lecture, we will traverse through the Spring 2023 versions of course logistics, and then an introduction towards optimization.

1.1 An Introductory Prompt

In this course, we discuss a problem in Computer Science called, "optimization". Understanding optimization is not limited to just understanding mathematical details, and has more to do with "problem formulation": for whatever technology comes, optimization is an involved technique of critical thinking.

1.2 Administratives, Summary

Logistics. For Discussion, Homework policies, please check the lecture slides. Friday sections are equivalent to the subsequent Monday sections.
Advices from Instructor. Do homeworks, collaborate with people.
New: Projects. We will have an option to do a project at the end of the semester, where we complete a project of choice. This is an effort to bridge the gap between students and research experience; where, provided a research paper, the student will extend that literature at the end of project. This will count towards grade and remove weight from exams if the project grade helps. Schemes will be announced. Projects will be in groups, released after the midterm and due during RRR week.

1.3 Introduction to The Subject Topic: Optimization

Optimization is an approach to problems.
It is the technique where we attempt to optimize some statistic resembling of a result. For example, in machine learning from DATA C100 (or EECS 16A/B), we have chosen appropriate loss functions for our learning task to characterize the performance of a model:

- Minimizing the MSE of a linear regresssion model.
- Minimizing the cross-entropy loss of a logistic regression model.
- Minimizing the distance traveled by an agent in a maze.

Here, we model learning tasks and attempt to optimize a metric. Depending on how we model and represent a problem, the model will perform differently on its learning task. Therefore, optimization is a study about "picking the right loss function" for some same objective. We consider these design choices, study how they affect learning process.

For another example, Air Traffic control is another optimization problem (at the point of writing, US has just experienced a flight paralysis); for another example, queueing and revenue computation are also common optimization problems. Optimization itself is omnipresent in modern applications.

In this course, we will learn about:

- Low rank approximation
- Ridge regression
- Stochastic Gradient Descent
- Dual Program (which always provides a lower bound for some optimization problem)
- Applications: LQR Control, Classification, SVM

Let's get into the Math now!!!

1.3.1 An Example of Problem Formulation

Say, we work in an oil production firm (I know, very US). We are allowed to make 10,000 barrels of crude oil, which may be produced into either *Jet Fuel* or *Gasoline*.
For every barrel of Jet Fuel produced, a revenue of 30 cents; for Gasoline, 20 cents. Meanwhile, 1 barrel of crude oil produces 0.6 barrels of Jet Fuel or 0.7 barrels of Gasoline.
Furthermore, the firm demands that we produce more than some amount of Jet Fuel and Gasoline (respectively, say, 1000 and 2000 barrels).
Finally, there are transportation capacities, where 180000 barrel miles are available. Distributing Gasoline costs 30 miles, and distributing Jet Fuel costs 10 miles.

We have now been presented a prompt: provided the above conditions, how can I maximize my revenue?
Let us first formulate this prompt mathematically:

$$\begin{cases} x_j = \text{ Quantity of Jet Fuel in barrels} \\ x_g = \text{ Quantity of Gasoline in barrels} \end{cases}$$

The revenue we generate would be formulated as

$$R(x_j, x_g) = 0.3x_j + 0.2x_g$$

such that we are presented the **constraints**:

$$\begin{cases} \text{Production Quantity Minimum:} & x_j \geq 1000, x_g \geq 2000 \\ \text{Total Available Resources:} & \frac{x_j}{0.7} + \frac{x_g}{0.6} \leq 10000 \\ \text{Transportation Capacity:} & 10x_j + 30x_g \leq 180000 \end{cases}$$

which are mathematically translated from the above English text.

In an optimization framework, we formualte a problem with the following frame:

Explain 1.3.1. Formulating Optimization Problems

In a general optimization problem, we attempt to minimize some function $f_0(\vec{x})$, such that some constraint exists:

$$\forall i \in \{1, \ldots, m\}, f_i(\vec{x}) \leq b_i$$

and here, $\vec{x}$ is listed as a representation of some system's state, existing within the domain of possible inputs (states).
The general objective of optimization is to find a state of system that minimizes $f_0(\vec{x})$, which we mathemati-

cally express the solution as:

$$\vec{x}^* = argmin_{\forall i \in \{1,\ldots,m\}, f_i(\vec{x}) \leq b_i} f_0(\vec{x})$$

See the example from above, and try matching the parts of optimization problem with the above framework!

There are some more types of optimization problems! One of them is the famous Least Squares Regression:

1.4 Least Squares Regression

The problem statement:

For some matrix A and a vector $\vec{b}$, solve for:

$$\min_{\vec{x}} ||A\vec{x} - \vec{b}||_2^2$$

Such problem can be widely applied to many mathematical problems, such as regression and projection.

1.4.1 Formulating Least Squares Regression

Let us discuss least squares algorithm in the context of linear regression:

Provided a dataset of points:

$$\{(x_1, y_1), \ldots, (x_n, y_n)\}$$

let us attempt to model a linear equation

$$y = mx + c$$

First of all, what is the variable we attempt to search/optimize for?
It would be m, c that are the unknowns.
To formulate the Least Squares Problem, we would need to formulate our linear equation for points into some matrix-vector multiplication, and knowing that we are solving for m and c, the formulation follows:

$$A \begin{bmatrix} m \\ c \end{bmatrix} = \vec{b}$$

Let us now fill in the contents of A and $\vec{b}$ for formulation, where A and $\vec{b}$ are of known quantities:

$$\begin{bmatrix} \vec{x} & \vec{1} \end{bmatrix} \begin{bmatrix} m \\ c \end{bmatrix} = \vec{y}$$

1.4.2 Closed-Form Solution of Least Squares Problem

From prior coursework we also know that there is a closed-form solution:

$$\vec{x}^* = (A^T A)^{-1} A^T \vec{b}$$

for matrices A with nontrivial nullspaces (see EECS16A for a proof that $N(A) = N(A^T A)$).

To minimize the L2 norm of difference between $A\vec{x}$ and $\vec{b}$, we attempt to find a vector $\vec{x}$ that minimizes the differences between them.
Note that for any choice of $\vec{x}$, it is by definition that $A\vec{x} \in Col(A)$.

We now face two claims that we resolve to proceed on solving this problem:

1. The endpoint of $proj_{Col(A)}\vec{b}$ is the point closest to the endpoint of $\vec{b}$ (if they share a same starting point) in $Col(A)$.
2. $\vec{OP} = A\vec{x}^*$, where $\vec{x}^*$ has the closed-form solution as previously described.

Claim 1 can be proven by contradiction:

> Assume another point C closer than P to B (such that $\vec{OB}$ is closer to $\vec{b}$). However, Pythagorean Theorem argues against it.

Claim 2 is a series of algebraic manipulation as outlined in previous courseworks:
Based on that the error vector $\vec{e} = \vec{b} - A\vec{x}$ should be orthogonal to the columnspace $Col(A)$ (because of geometry):

$$\begin{aligned} A^T\vec{e} &= 0 \\ A^T(A\vec{x}^* - \vec{b}) &= 0 \\ A^TA\vec{x}^* &= A^T\vec{b} \\ \vec{x}^* &= (A^TA)^{-1}A^T\vec{b} \end{aligned}$$

Such result is also known as the **Normal Equation**.

Interestingly, Least Squares Algorithm has a quadratic property, causing it to be some parabolic object that performs a **convex function** to optimize along; that is, the local minimum of this function is a global minimum.

Chapter 2

Linear Algebra Bootcamp: Norms, Gram-Schmidt, QR, FTLA

2.1 Vectors and Norms

In the previous lecture, we have recognized the following symbol:

Symbol 2.1.1. Euclidean Norm

The Euclidiean Norm, otherwise known as a **2-Norm**, is a mathematical quantity for some vector $\vec{x}$:

$$\|\vec{x}\|_2 = \sqrt{\sum_i x_i^2}$$

which measures the Euclidean distance (Pythagorean Theorem's results) between the starting and terminal point of a vector.

While the Euclidean Norm is an extremely common norm, there exist more norms to vectors.
Particularly, **norms** are functions that satisfy the following properties:

Definition 2.1.1. Norm

A norm is a function $f : \mathcal{X} \to \mathbb{R}$ such that:

1. Non-negativeness: $\forall \vec{x} \in \mathcal{X}, \|\vec{x}\| \geq 0$, and $\|\vec{x}\| = 0 \iff \vec{x} = \vec{0}$
2. Triangle Inequality: $\forall \vec{x}, \vec{y} \in \mathcal{X}, \|\vec{x} + \vec{y}\| \leq \|\vec{x}\| + \|\vec{y}\|$
3. Scalar Multiplication: $\forall \alpha \in \mathbb{R}, \vec{x} \in \mathcal{X}, \|\alpha\vec{x}\| = \alpha\|\vec{x}\|$

2.1.1 LP-Norms

It is noteworthy to mention that, almost every norm demonstrates one property of the vector. For the diversity of characteristics that a vector may have, there certainly exists a diversity of norms for vectors.

For example, a general family of norms that satisfy the above properties would be the **LP-norms**:

Definition 2.1.2. LP-Norms

LP-Norms are norm functions defined as:

$$\|\vec{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}$$

for some natural number p.

Particularly, the Euclidean Norm, otherwise known as the *2-norm* is LP-norm with $p = 2$.
Meanwhile, the 1-norm and ∞-norm are defined as follows:

$$\|\vec{x}\|_1 = \sum_{i=1}^{n} |x_i|$$
$$\|\vec{x}\|_\infty = \max_{i=1,\ldots,n} |x_i|$$

where, the 1-norm of a vector can be stated as the Manhattan distance (Taxicab geometry) of between that vector's starting and terminal point.

2.2 Cauchy-Schwartz Inequality

This inequality was active in EECS 16A!

Definition 2.2.1. Cauchy-Schwartz Inequality

The inequality is phrased as:

$$|\vec{x}^T\vec{y}| \leq \|\vec{x}\|_2\|\vec{y}\|_2$$

And using the property,

$$\forall x \in \mathbb{R}, |\cos(x)| \leq 1$$

This inequality originates from the following algebraic work:

$$\begin{aligned}|\langle\vec{x},\vec{y}\rangle| = |\vec{x}^T\vec{y}| &= |\vec{y}^T\vec{x}| \\ &= |\|\vec{x}\|_2\|\vec{y}\|_2\cos(\theta_{\vec{x},\vec{y}})| \leq |\|\vec{x}\|_2\|\vec{y}\|_2|\end{aligned}$$

We observe that the above derivation holds statement on equivalence between inner product and product of magnitudes, cosine.
That is justified by the mechanics along which we find the projection of $\vec{y}$ onto $\vec{x}$:

$$\text{proj}_{\vec{x}}(\vec{y}) = \vec{x}\frac{\vec{x}^T\vec{y}}{\|\vec{x}\|_2^2}$$

where, since the projection is a multiple of $\vec{x}$ such that $\text{proj}_{\vec{x}}(\vec{y}) = t\vec{x}$, we also recognize that,

$$\cos(\theta_{\vec{x},\vec{y}}) = \frac{\|t\vec{x}\|_2}{\|\vec{y}\|_2}$$

Upon matching the two equations, I acquire:

$$\begin{aligned}t = \cos(\theta_{\vec{x},\vec{y}})\frac{\|\vec{y}\|_2}{\|\vec{x}\|_2} &= \frac{\vec{x}^T\vec{y}}{\|\vec{x}\|_2^2} \\ |\|\vec{x}\|_2\|\vec{y}\|_2| &= \langle\vec{x},\vec{y}\rangle\end{aligned}$$

2.2.1 Holder's Inequality and Norm Ball

In fact, we find that the Cauchy-Schwartz is a general case of this inequality:

Definition 2.2.2. Holder's Inequality

Provided the quantities

$$\vec{x}, \vec{y} \in \mathbb{R}^n, \text{and some } p, q \geq 1 \text{ s.t. } \frac{1}{p} + \frac{1}{q} = 1$$

Then,

$$|\vec{x}^T \vec{y}| \leq \|\vec{x}\|_p \|\vec{y}\|_p$$

From this concept, let us consider the concept of *"norm ball"*: the geographical object containing all vectors such that their lp-norm is 1.

- For a 2-norm, for example, the norm ball looks like a unit circle (containing all 2D vectors with length 1, hence a circle of radius 1).
- For a 1-norm, similar logic guides us to a diagonally placed circle (rotated $45°$) centered at the origin with side lengths 2.
- For the ∞-norm, has a norm ball of a square centered at origin with side length 2. This embeds the unit ball from 2-norm, which embeds the unit ball from 1-norm.

The last bullet point hints that the area of norm ball is larger for any increase in p such that it embeds the prior norm balls.

Now, let us attempt to solve for some optimization regarding a norm ball:

$$\max_{\|\vec{x}\|_p \leq 1} \vec{x}^T \vec{y}$$

p = 2:

The solution would be, by the Cauchy Schwartz's implication,

$$\vec{x}^* = \frac{\vec{y}}{\|\vec{y}\|_2}$$

p = 1:

The expression of dot product is equivalently

$$x_1 y_1 + \cdots + x_n y_n$$

Where the constraint is

$$|x_1| + \cdots + |x_n| \leq 1$$

For each value the components of $\vec{x}$, which is finite and upper bounded by 1, we should allocate maximum contribution to the maximum element of $\vec{y}$ to maximize this dot product (a weighted sum of $\vec{y}$'s component, essentially).
Therefore, let i be the index at which $\vec{y}$ has the component of largest absolute value,

> There is an achievable solution for $\vec{x}^*$, being the unit vector $\vec{e}_i$ multiplied by $sgn(y_i)$, so to counter for cases where $\vec{y}$ is negative.

In turn, we see that

$$\vec{x}^T \vec{y} = \max_i |y_i| = \|\vec{y}\|_\infty$$

Meanwhile, let us use the Holder's Inequality to achieve a more rigorous proof. Holder's Inequality states that,

$$|\vec{x}^T \vec{y}| \leq \|\vec{x}\|_1 \|\vec{y}\|_\infty = \max_i |y_i|$$

The Holder's Inequality expresses **an upper bound**, and the formulation in above section shows **achievability of upper bound**.

Alternative proof of p = 1 via Triangle Inequality:
Via the triangle inequality, we acquire:

$$\begin{aligned}|\vec{x}^T\vec{y}| &= |\sum_i x_i y_i| \\ &\underset{Triangle\ Inequality}{\leq} \sum_i |x_i y_i| \\ &= \sum_i |x_i||y_i| \\ &\leq \sum_i |x_i| \max_i |y_i| = \max_i |y_i| = \|\vec{x}\|_\infty\end{aligned}$$

We see the expression $|\vec{x}^T\vec{y}|$ has an **achievable upper bound**, assembling all necessary aspects of deriving the solution for the optimization problem.

p = ∞:

We can find upper bound of the expressionvia Holder's Inequality,

$$|\vec{x}^T\vec{y}| \leq \|\vec{x}\|_1 \|\vec{y}\|_\infty = \sum_i |y_i|$$

The infinity norm of $\vec{x}$ shows the constraint that:

$$\max_i |x_i| = 1$$

and we may simply compute:

$$x_i = sgn(y_i)$$

to find and certify the achievability of this upper bound.

2.3 Gram-Schmidt Orthonormalization and QR Decomposition

Gram-Schmidt Orthonormalization is an algorithmic technique to find an orthonormal basis for a set of vectors. The procedure of such algorithm is portrayed as defined in the following cell:

Definition 2.3.1. The Procedure of Gram-Schmidt Orthonormalization

Let us have a set of vectors $\{\vec{a_1}, \ldots, \vec{a_n}\}$.

1 $\vec{q_1} = \frac{\vec{a_1}}{\|\vec{a_1}\|_2}$

2 for i in $\{2, \ldots, n\}$:

3 $\quad \vec{z_i} = proj_{\{\vec{q_1},\ldots,\vec{q_{i-1}}\}}\vec{a_i} = \sum_{j=1}^{i-1} \vec{q_j}\langle\vec{a_i}, \vec{q_j}\rangle$

4 $\quad \vec{s_i} = \vec{a_i} - \vec{z_i}$

5 $\quad \vec{q_i} = \frac{\vec{s_i}}{\|\vec{s_i}\|_2}$

6 return $\{\vec{q_1}, \ldots, \vec{q_n}\}$

Then, QR Decomposition is a technique to factorize a matrix A based on its orthonormal basis:

Definition 2.3.2. QR Decomposition

The QR decomposition of a matrix A is to factorize A as the product QR, where the two matrices are decided as:

$$\begin{cases} Q, & \text{An orthonormal matrix whose span is equal to that of } A \\ R, & R_{ij} = \vec{q_i} \cdot \vec{a_j} \end{cases}$$

where we may define vectors $\vec{a_i}$ as the i^{th} column of A, and set of vectors $\vec{q_i}$ may be found as the orthonormal base of $\mathcal{R}(A)$ via Gram-Schmidt.

Explain 2.3.1. Example of QR Decomposition

Problem: Solve for its QR Decomposition of

$$A = \begin{bmatrix} 3 & -3 & 1 \\ 4 & -4 & 7 \\ 0 & 3 & 3 \end{bmatrix}$$

Finding the first column of Q:

$$\vec{q_1} = \frac{1}{5}\begin{bmatrix} 3 \\ 4 \\ 0 \end{bmatrix}$$

Finding the second column of Q:

$$\begin{aligned} \text{proj}_{\vec{q_1}}(\vec{a_2}) &= (\vec{a_2} \cdot \vec{q_1})\vec{q_1} \\ \vec{z_2} &= \vec{a_2} - \text{proj}_{\vec{q_1}}(\vec{a_2}) \\ &= \begin{bmatrix} -3 \\ -4 \\ 3 \end{bmatrix} - \begin{bmatrix} -3 \\ -4 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 3 \end{bmatrix} \\ \vec{q_2} &= \frac{\vec{z_2}}{\|\vec{z_2}\|_2} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{aligned}$$

Finding the third column of Q:

$$\begin{aligned} \text{proj}_{\{\vec{q_1},\vec{q_2}\}}(\vec{a_3}) &= (\vec{a_3} \cdot \vec{q_1})\vec{q_1} + (\vec{a_3} \cdot \vec{q_2})\vec{q_2} \\ &= \begin{bmatrix} -3 \\ -4 \\ 3 \end{bmatrix} \\ \vec{z_3} &= \vec{a_3} - \text{proj}_{\{\vec{q_1},\vec{q_2}\}}(\vec{a_3}) \\ &= \begin{bmatrix} 1 \\ -7 \\ 3 \end{bmatrix} - \begin{bmatrix} -3 \\ -4 \\ 3 \end{bmatrix} = \begin{bmatrix} 4 \\ -3 \\ 0 \end{bmatrix} \\ \vec{q_3} &= \frac{\vec{z_3}}{\|\vec{z_3}\|_2} = \frac{1}{5}\begin{bmatrix} 4 \\ -3 \\ 0 \end{bmatrix} \end{aligned}$$

And now, finding out that we still have to find the matrix R (sob emoji):

$$R = \begin{bmatrix} \vec{q_1} \cdot \vec{a_1} & \vec{q_1} \cdot \vec{a_2} & \vec{q_1} \cdot \vec{a_3} \\ 0 & \vec{q_2} \cdot \vec{a_2} & \vec{q_2} \cdot \vec{a_3} \\ 0 & 0 & \vec{q_3} \cdot \vec{a_3} \end{bmatrix} = \begin{bmatrix} 5 & -5 & -5 \\ 0 & 3 & 3 \\ 0 & 0 & 5 \end{bmatrix}$$

Chapter 3

Linear Algebra: Symmetric Matrices

3.1 Fundamental Theorem of Linear Algebra

Spaces are sets of vectors that follow the ten commandments (or rules) of being a vector space.
Among such perspective, we find ourselves curious about the orthogonality of sets of vectors (vector space):

Theorem 3.1.1. Orthogonal Decomposition of Space

Let us consider some vector space $\mathcal{X}$ and some subspace $\mathcal{S} \subseteq \mathcal{X}$.
Then, we may state that,

$$\forall \vec{x} \in \mathcal{X}, \vec{x} = \vec{s} + \vec{r} \text{ uniquely, s.t. } \vec{s} \in \mathcal{S} \wedge \vec{r} \in \mathcal{S}^{\perp}$$

such that,

$$\mathcal{S}^{\perp} = \{\vec{y} : \langle \vec{x}, \vec{y} \rangle = 0, \vec{x} \in \mathcal{S}\}$$

Conseuqeuntially, we are stating that for the space $\mathcal{X}$,

$$\mathcal{X} = \mathcal{S} \oplus \mathcal{S}^{\perp}$$

In other words, it is a **direct sum** of the vector spaces $\mathcal{S}$ and $\mathcal{S}^{\perp}$. We may write any vector in $\mathcal{X}$ as the sum of two components where each component belongs to some specific subspace of the direct sum.

This leads to a theorem about decomposing a vector space from some given matrix:

Theorem 3.1.2. Fundamental Theorem of Linear Algebra

Theorem. Consider a matrix $A \in \mathbb{R}^{m \times n}$, then,

$$\mathbb{R}^n = N(A) \oplus R(A^T)$$

where, similarly,

$$\mathbb{R}^m = \mathcal{R}(A) \oplus N(A^T)$$

Proof. We may use Orthogonal Decomposition of Space (Theorem 3.1.1) to aid our proof. If so, we would only need to show the two following facts:

1. $N(A) \perp \mathcal{R}(A^T)$, or equivalently, $N(A) = \mathcal{R}(A^T)^{\perp}$
2. $N(A^T) \perp \mathcal{R}(A)$

Let us perform a proof on the first fact, which would be to prove the equivalence of two sets.
First, we may prove that $N(A) \subseteq \mathcal{R}(A^T)^{\perp}$:

Suppose that there is an arbitrary vector $\vec{u} \in N(A)$, where $A\vec{u} = \vec{0}$.
Then, we realize that,

$$(A\vec{u}) = \vec{0}, \vec{0}^T = \vec{u}^T A^T$$

then, for any arbitrary vector $\vec{u}'$, we may find that

$$\vec{u}^T A^T \vec{u}' = \vec{0}^T \vec{u}' = 0$$

Therefore, the vector $\vec{u}$ is orthogonal to any vector belonging to $\mathcal{R}(A^T)$, which would mathematically state,

$$\vec{u} \in \mathcal{R}(A^T)^\perp$$

Then, we may prove the opposite direction: $\mathcal{R}(A^T)^\perp \subseteq N(A)$

Suppose that there is an arbitrary vector $\vec{w} \in \mathcal{R}(A^T)$, and $\vec{x} \in \mathcal{R}(A^T)^\perp$; by which, we would state for the arbitrary pair $(\vec{w}, \vec{x})$ that

$$\forall \vec{w} \in \mathcal{R}(A^T), \vec{x} \in \mathcal{R}(A^T)^\perp,\ (\vec{x} \cdot \vec{w} = 0)$$

Thus, $\forall \vec{w} \in \mathcal{R}(A^T)$:

$$\begin{aligned} \langle \vec{x}, \vec{w} \rangle &= 0 \\ \langle \vec{x}, A^T \vec{w} \rangle &= 0 \\ \vec{x}^T A^T \vec{w} &= 0 \end{aligned}$$

Which qualifies us to state that $A\vec{x} = \vec{0}$.
Therefore, $\vec{x} \in N(A)$.

Applying a symmetrical proof to the second fact allows us to provide a complete proof for this theorem.

3.2 Minimum Norm Problem

This is a sister problem to the Least Squares Problem. Let the following table show their differences:

Least Squares	Minimum Norm
Solves an overdetermined system	Solves an underdetermined system
Formulation: $\min_{\vec{x}} \|A\vec{x} - \vec{b}\|_2^2$	Formulation: $\min_{A\vec{x}=\vec{b}} \|\vec{x}\|_2^2$

3.2.1 Solution to the Minimum Norm Problem

Let us provide some $\vec{x}$ that is a solution to $A\vec{x} = \vec{b} \neq \vec{0}$.
Some component of $\vec{x}$ might belong to the nullspace of A. Meanwhile, we also acknowledge that $\vec{x}$ is some arbitrary vector in the real space. Therefore, the vector $\vec{x}$ may be decomposed into $\vec{x} = \vec{n} + \vec{r}$, where

$$\vec{n} \in N(A), \vec{r} \in \mathcal{R}(A^T)$$

In that case, we now minimize for $\vec{r} \in \mathcal{R}(A^T)$ under the setup that,

$$A\vec{x} = A(\vec{n} + \vec{r}) = A\vec{r} = \vec{b}$$

We may recognize that $\vec{n} \cdot \vec{r} = 0$ due to the orthogonal decomposition's nature. Therefore,

$$\|\vec{x}\|_2^2 = \|\vec{n}\|_2^2 + \|\vec{r}\|_2^2$$

Therefore, let $\vec{x}^* = A^T\vec{w}$ for some specific $\vec{w}$, so to state that $\vec{x}^* \in \mathcal{R}(\mathcal{A}^\mathcal{T})$:

$$\begin{aligned} A\vec{x} &= \vec{b} \\ AA^T\vec{w} &= \vec{b} \\ \vec{w} &= (AA^T)^{-1}\vec{b} \\ \vec{x}^* &= A^T(AA^T)^{-1}\vec{b} \end{aligned}$$

3.3 Principal Component Analysis

Principal Component Analysis is a technique to reduce dimensionality in high-dimension datapoints, so to find underlying feature patterns and enable plotting. This can be achieved via projecting data onto a lower dimension structure to recover lower dimensional structure from the original high dimensional data.

Mathematically, such technique would be to project $\vec{x_1}, \ldots, \vec{x_n}$ onto $\vec{w} \in \mathbb{R}^p$ such that the projected vectors are all as close to the original vectors as possible.

Explain 3.3.1. Projections

For projecting some vector $\vec{x}$ onto $\vec{w}$, we would obtain

$$\text{proj}_{\vec{w}}(\vec{x}) = (\vec{w}^T\vec{w})^{-1}(\vec{w}^T\vec{x})\vec{w}$$

Projecting vector $\vec{x}$ onto a columnspace of matrix A then follows the least squares formula,

$$(A^TA)^{-1}A^T\vec{x}$$

Now, let's discuss how would PCA work:

Explain 3.3.2. PCA as an Optimization Problem

Formulation.
Let the individual projection errors be phrased as

$$\|\vec{x_i} - \langle\vec{w}, \vec{x_i}\rangle\vec{w}\|_2^2$$

Where, the average projection error can thus be quantified as,

$$R(\vec{w}) = \frac{1}{n}\sum_{i=1}^{n}\|\vec{x_i} - \langle\vec{w}, \vec{x_i}\rangle\vec{w}\|_2^2 = MSE(\vec{w})$$

Therefore, the formulation of optimization problem is:

$$\min_{\vec{w}} R(\vec{w}) \text{ subject to } \|\vec{w}\|_2^2 = 1$$

And the assumption of PCA dataset is that it is zero-meaned, because de-meaned data will allow less bias when deciding the weight vector for PCA.
Now, let us gently poke at the problem, even if we might barely know the problem.
Let us consider the projection error,

$$\|\vec{x_i} - \langle\vec{w}, \vec{x_i}\rangle\vec{w}\|_2^2 = \|\vec{e_i}\|_2^2$$

The projection error can be algebraically simplified:

$$\begin{aligned}\|\vec{e_i}\|_2^2 &= (\vec{x_i} - \langle \vec{w}, \vec{x_i} \rangle \vec{w})^T \cdot (\vec{x_i} - \langle \vec{w}, \vec{x_i} \rangle \vec{w}) \\ &= \|\vec{x_i}\|_2^2 - 2\langle \vec{x_i}, \vec{w} \rangle^2 + \langle \vec{x_i}, \vec{w} \rangle^2 \|\vec{w_i}\|_2^2 \\ &= \|\vec{x_i}\|_2^2 - \langle \vec{x_i}, \vec{w} \rangle^2\end{aligned}$$

Therefore, removing the fixed costs in the error, we are essentially facing the following optimization problem:

$$\max_{\vec{w}} \frac{1}{n} \sum_{i=1}^{n} \langle \vec{x_i}, \vec{w} \rangle^2$$

Chapter 4

Principal Component Analysis

4.1 Symmetric Matrix

Let us begin with a formal definition of Symmetric Matrix:

Definition 4.1.1. Symmetric Matrix

A matrix $A \in \mathbb{R}^{n \times n}$ is a **symmetric matrix** if

$$A = A^T$$

or equivalently,

$$\forall i, j \in \{1, \cdots, n\}, a_{i,j} = a_{j,i}$$

and in such case we express,

$$A \in \mathbb{S}^n$$

Symmetric Matrices have such properties summarized by the Spectral Theorem:

Theorem 4.1.1. Spectral Theorem

If $A \in \mathbb{S}^n$, then:

1. All of its eigenvalues are real.
2. Eigenspaces corresponding to distinct eigenvalues are orthogonal.
3. The algebraic multiplicity of an eigenvalue is equal to its geometric multiplicity

   (alternatively, the matrix is diagonalizable, such that there exists an orthonormal U and diagonal Σ to compose $A = U\Sigma U^T$).

An interlude here:

Definition 4.1.2. Multiplicity of Eigenvalues

A review from EECS 16B:

- The algebraic multiplicity of an eigenvalue (m_A) is the number of times such eigenvalue appears in its matrix's characteristic polynomial.
- The geometric multiplicity of an eigenvalue (m_G) is the number of lienarly independent eigenvectors that such eigenvalue corresponds to.

Meanwhile, symmetric matrices also enjoy special properties in their eigenvalues:

Theorem 4.1.2. Variational Characteristics of Rayleigh Coefficient

The Rayleigh Coefficient of some symmetric matrix $A \in \mathcal{S}^n$ and vector $\vec{x} \in \mathbb{R}^n$ is defined as:

$$R_{A,\vec{x}} = \frac{\vec{x}^T A \vec{x}}{\vec{x}^T \vec{x}}$$

Theorem. Then, we may acknowledge via proof that:

$$\forall \vec{x} \neq \vec{0}, \lambda_{min}(A) \leq R_{A,\vec{x}} \leq \lambda_{max}(A)$$

and alternatively we may formulate the above inequality such that,

$$\begin{cases} \lambda_{max}(A) = \max_{\|\vec{x}\|_2^2=1} \vec{x}^T A \vec{x} \\ \lambda_{min}(A) = \min_{\|\vec{x}\|_2^2=1} \vec{x}^T A \vec{x} \end{cases}$$

Proof. Let us define that $\vec{y} = U^T \vec{x}$, and we will proceed onto the algebraic work:

$$\begin{aligned} \vec{x}^T A \vec{x} &= \vec{x} U \Lambda U^T \vec{x} \\ &= \vec{y}^T \Lambda \vec{y} \\ \|\vec{y}\|_2^2 &= 1 \end{aligned}$$

And, furthermore,

$$\begin{aligned} \vec{y}^T \Lambda \vec{y} &= \sum \lambda_i y_i^2 \\ &\leq \sum \lambda_{max} y_i^2 = \lambda_{max} \\ \vec{y}^T \Lambda \vec{y} &= \sum \lambda_i y_i^2 \\ &\geq \sum \lambda_{min} y_i^2 = \lambda_{min} \end{aligned}$$

So we have now found the upper bound and lower bound of the optimization problem, justifying the boundaries of inequality.
And, we may also acknowledge that providing $\vec{x}$ as the normalized eigenvector of maximum and minimum eigenvalues provide the solution for achieving the lower and upper bound:

$$\begin{aligned} \vec{v}_{\lambda_{max}}^T A \vec{v}_{\lambda_{max}} &= \lambda_{max} \|\vec{v}_{\lambda_{max}}\|_2^2 \\ &= \lambda_{max} \\ \vec{v}_{\lambda_{min}}^T A \vec{v}_{\lambda_{min}} &= \lambda_{min} \|\vec{v}_{\lambda_{min}}\|_2^2 \\ &= \lambda_{min} \end{aligned}$$

Below, let us discuss how we may use the properties of symmetric matrices to solve Principal Component Analysis, which is a problem we were studying in the previous lecture.

4.2 PCA

Principal Component Analysis is a technique to reduce dimensionality in high-dimension datapoints, so to find underlying feature patterns and enable plotting. This can be achieved via projecting data onto a lower dimension structure to recover lower dimensional structure from the original high dimensional data.

We will pick up from last lecture with a new formulation for optimization problem:

Explain 4.2.1. Introduction of Principal Component Analysis as Problem

Suppose we have a set of vectors:

$$\{\vec{x_1}, \ldots, \vec{x_n}\}, \forall i(\vec{v_i} \in \mathbb{R}^p)$$

And we would like to project our data into a lower dimensional subspace, such that we may reduce computational cost, and projected vectors are close to the original data.

Let us find the first principal component, $\vec{w}$, then, such that the dataset projected on this direction is as close to the original data as possible. Then, we will repeat the process of:

- Find principal component $\vec{w}_i$

↓ Remove from dataset (usually a matrix) all projected components onto principal component i

- Repeat process to find principal component $\vec{w}_{i+1}$

4.2.1 Formulating PCA as an Optimization Problem

Let us now formulate the mathematical aspects of PCA.

Suppose we have a set of vectors:

$$\{\vec{x_1}, \ldots, \vec{x_n}\}, \text{and } \forall i(\vec{v_i} \in \mathbb{R}^p)$$

Let us find a vector $\vec{w}$ along the constraint and objective function that:

$$\begin{cases} \|\vec{w}\|_2 & = 1 \\ MSE(\vec{w}) & = \frac{1}{n}\sum_{i=1}^{n} e_i^2 \end{cases}$$

where, the individual projection error terms were defined and derived into:

$$\begin{aligned} \|\vec{e_i}\|_2^2 &= (\vec{x_i} - \langle \vec{w}, \vec{x_i} \rangle \vec{w})^T \cdot (\vec{x_i} - \langle \vec{w}, \vec{x_i} \rangle \vec{w}) \\ &= \|\vec{x_i}\|_2^2 - 2\langle \vec{x_i}, \vec{w} \rangle^2 + \langle \vec{x_i}, \vec{w} \rangle^2 \|\vec{w_i}\|_2^2 \\ &= \|\vec{x_i}\|_2^2 - \langle \vec{x_i}, \vec{w} \rangle^2 \end{aligned}$$

Therefore,

$$MSE(\vec{w}) = \frac{1}{n}\sum_{i=1}^{n}(\|\vec{x_i}\|_2^2 - \langle \vec{w}, \vec{x_i} \rangle^2)$$

To consider the optimization problem holistically, we should formulate that,

$$\underset{\vec{w}}{argmin} MSE(\vec{w}) = \underset{\vec{w}}{argmin} - \frac{1}{n}\sum_{i=1}^{n} \langle \vec{w}, \vec{x_i} \rangle^2$$

which is equivalently solving for

$$\underset{\vec{w}}{argmax} \sum_{i=1}^{n} \langle \vec{w}, \vec{x_i} \rangle^2$$

and we would recognize so becuase the ignored aspect of MSE, $\frac{1}{n}\sum_{i=1}^{n} \|\vec{x}\|_2^2$, is a nonnegative fixed cost.

4.2.2 Solution to PCA as an Optimization Problem

Now, let us attempt to solve for the above formulation:

Keep in mind that our data matrix would appear as:

$$X = \begin{bmatrix} \vec{x_1}^T \\ \vdots \\ \vec{x_n}^T \end{bmatrix}$$

We may begin with the algebraic work then.

$$X\vec{w} = \begin{bmatrix} \vec{x_1}^T \vec{w} \\ \vdots \\ \vec{x_n}^T \vec{w} \end{bmatrix}$$

Let us capture the information and make some algebraic derivation:

$$\begin{aligned} \frac{1}{n}\sum \langle \vec{w}, \vec{x_i} \rangle^2 &= \frac{1}{n}\sum (\vec{x_i}^T \vec{w})^2 \\ &= \frac{1}{n}\|X\vec{w}\|_2^2 \\ &= \vec{w}^T \frac{X^T X}{n} \vec{w} \end{aligned}$$

We call the matrix in above intermediate form the covariance matrix:

$$C = \frac{X^T X}{n} = C^T$$

And, using the knowledge from Section 4.1 about Rayleigh coefficient, we obtain that the solution of:

$$argmax_{\vec{w}}\ \vec{w}^T C \vec{w} \text{ subject to } \|\vec{w}\|_2^2 = 1$$

would be the eigenvector of $\lambda_{max}(C)$.

Chapter 5

SVD and Low-Rank Approximation

5.1 Singular Value Decomposition (SVD)

Let us suppose that there is a matrix $A \in \mathbb{R}^{m \times n}$ whose rank is $rk(A) = r$, then there exists a decomposition of A in the shape of:

$$A = \sum_{i=1}^{n} \sigma_i \vec{u_i} \vec{v_i}^T$$

where,

- The set $\{\vec{u_1}, \ldots, \vec{u_n}\}$ is orthonormal
- The set $\{\vec{v_1}, \ldots, \vec{v_n}\}$ is orthonormal
- $\forall i \in [1, r], \sigma_i > 0$, and $\forall i \in [1, n], \sigma_i \geq 0$

Symbol 5.1.1. Compact SVD

The above dyadic decomposition of SVD written in terms of matrix transformation would be:

$$\begin{aligned} A &= \begin{bmatrix} \vec{u_1} & \cdots & \vec{u_r} \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix} \begin{bmatrix} \vec{v_1}^T \\ \vdots \\ \vec{v_r}^T \end{bmatrix} \\ &= U_r \Sigma_r V_r^T \end{aligned}$$

Meanwhile, the full SVD that still considers zero-valued σ_i is written as follows:

Symbol 5.1.2. Full SVD

The formulation follows:

$$\begin{aligned} A &= \begin{bmatrix} \vec{u_1} & \cdots & \vec{u_m} \end{bmatrix} \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \vec{v_1}^T \\ \vdots \\ \vec{v_n}^T \end{bmatrix} \\ &= U \Sigma V^T \end{aligned}$$

from which, we may observe that the shape of Σ is highly dependent on the shape of A:

- If A is tall, so is Σ
- If A is wide, so is Σ

The squareness of U and V in full SVD allows us to also decompose A^TA as:

$$A^TA = (U\Sigma V^T)^T(U\Sigma V^T) = V\Sigma^TU^TU\Sigma V^T = V\Sigma^2V^T$$

which is more convenient than the compact form SVD that instead uses rectangular U and V.

5.1.1 Formation of SVD

As singular vectors are eigenvectors of symmetric matrices A^TA, by applying Spectral Theorem, we find that singular vectors of A would all be orthonormal.
Then, let's use this point as an opportunity to praise A^TA's symmetricness:

Theorem 5.1.1. The Formation of SVD

Let the nonzero eigenvalues of A^TA be

$$\lambda_1 \geq \cdots \geq \lambda_r > 0$$

where, upon dimming that $rk(A^TA) = r$, we may obtain $rk(A) = rk(A^TA) = r$ (using rank-nullity theorem, and the fact that $N(A^TA) = N(A)$).
Let us also suppose that orthonormal vectors in V form eigenpairs with the above eigenvalues:

$$\forall i \in [1, r], A^TA\vec{v_i} = \lambda_i\vec{v_i}$$

Now, define that,

$$\sigma_i := \sqrt{\lambda_i}$$

such that:

$$\vec{u_i} := \frac{A\vec{v_i}}{\sigma_i}$$

Let us now check, whether the above formation grants an orthonormal U:

Theorem 5.1.2. Orthonormality of Set of $\vec{u_i}$

Let us follow the definition of $\vec{u_i}$ from the last block.
Then,

$$A\vec{v_i} = \sigma_i\vec{u_i}$$

Suppose now we have some vectors $\vec{u_i}$ and $\vec{u_j}$, then

$$\begin{aligned}\vec{u_i} \cdot \vec{u_j} &= \frac{\vec{v_i}^TA^T}{\sigma_i}\frac{A\vec{v_j}}{\sigma_j} \\ &= \frac{\vec{v_i}^T\lambda_j\vec{v_j}}{\sigma_i\sigma_j} = 0\end{aligned}$$

Now that we have proven the orthonormality and note the definitions of $\vec{u_i}$ and $\vec{v_i}$, we arrive at the conclusion:

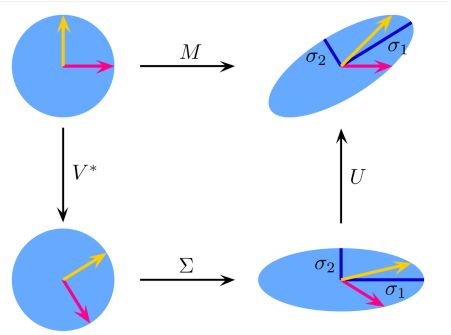
$$AV = U\Sigma, A = U\Sigma V^T$$

such arithmetic operation is only possible under the full SVD form (such that V can be noted as orthonormal at all). This is once again a demonstration of significance of full SVD.

5.1.2 Geometry of SVD

Figure 5.1.1. Geometric Property of SVD as Transformation

We may see from the below image's operations,



$$M = U \cdot \Sigma \cdot V^*$$

that SVD is, as priorly mentioned, a combination of three transformations. Where,

- The orthonormal matrices U, V are reflections or rotations.
- The matrix Σ is a dilation.

Furthermore, let us observe the vectors drawn on the unit circle.
Since V is orthonormal, we may conclude that $\vec{v_1} \perp \vec{v_2}$. However, bizzarely, we may also find that

$$A\vec{v_1} \perp A\vec{v_2}$$

5.2 Low Rank Approximation

Low Rank Approximation is the task of approximating a matrix A in a lower rank than $rk(A)$, to conserve computational cost and perform efficient computations.

5.2.1 Matrix Norms

Let us first discuss the optimization function of this task: matrix norm.

The norm of matrix also comes in diverse form, because matrices can be interpreted in many forms.
For example:

- Chunk of data (like an array)
- Operator of transformation

Therefore, along these interpretations, we may present respective definitions of matrix norms.

Definition 5.2.1. Frobenius Norm

The Frobenius Norm is defined as:

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{i,j}^2} = \sqrt{Trace(A^T A)}$$

This is because the Frobenius inner product may be defined as:

$$\langle A, B \rangle_F = Trace(A^T B) = \sum_{i=1}^{n} \vec{A_i} \cdot \vec{B_i}$$

and, as we recognize the relation between norm and inner product,

$$\|A\|_F = \sqrt{\langle A, A \rangle_F} = \sqrt{\sum_{i=1}^{n} \vec{A_i} \cdot \vec{A_i}}$$

The Frobenius Norm has some interesting properties:

Theorem 5.2.1. Invariance of Frobenius Norm upon Orthonormal Matrices

Let $U_1 \in \mathbb{R}^{m \times m}$, $A \in \mathbb{R}^{m \times n}$, and $U_2 \in \mathbb{R}^{n \times n}$, where U_1, U_2 are orthonormal.
Then,

$$\begin{aligned} \|U_1 A\|_F &= \sqrt{Trace(A^T U_1^T U_1 A)} \\ &= \sqrt{Trace(A^T A)} = \|A\|_F \\ \|AU_2\|_F &= \sqrt{Trace(U_2^T A^T A U_2)} \\ &\underset{\text{Cyclic Property}}{=} \sqrt{Trace(A^T A)} = \|A\|_F \end{aligned}$$

Now, let's discuss another definition of matrix norm:

Definition 5.2.2. Operator Norm (aka. Spectral Norm, L2-Norm)

Such norm is defined as:

$$\|A\|_2 = \max_{\|\vec{x}\|_2 = 1} \|A\vec{x}\|_2 = \sigma_{max}(A)$$

The derivation follows:

$$\begin{aligned} \max_{\|\vec{x}\|_2 = 1} \|A\vec{x}\|_2 &= \max_{\|\vec{x}\|_2 = 1} \sqrt{\vec{x}^T A^T A \vec{x}} \\ &\underset{\text{Rayleigh Coefficient}}{=} \sqrt{\lambda_{max}(A^T A)} = \sigma_{max}(A) \end{aligned}$$

Alternatively, it may be stated as the maximum possible transformation some matrix A can offer a normal vector. Upon that above statement,

$$\forall \vec{x}, \|A\vec{x}\|_2 \leq \|A\|_2 \|\vec{x}\|_2$$

Furthermore, L2-Norms are also invariant upon orthonormal matrices.
Suppose we attach another orthonormal matrix U to the original matrix A, then the optimization problem develops a solution fairly similaly:

$$\begin{aligned} \max_{\|\vec{x}\|_2 = 1} \|AU\vec{x}\|_2 &= \max_{\|\vec{x}\|_2 = 1} \sqrt{\vec{x}^T U^T A^T A U \vec{x}} \\ &= \max_{\|\vec{x}\|_2 = 1} \sqrt{\vec{y}^T A^T A \vec{y}} \\ &= \sqrt{\lambda_{max}(A^T A)} = \sigma_{max}(A) = \|A\|_2 \end{aligned}$$

where since we still find that $\vec{y}$ is normal, the attachment of U providing no change to the optimization problem. Meanwhile, we find this phenomenon to persist even if U is multiplied from an opposite direction:

$$\begin{aligned}\max_{\|\vec{x}\|_2=1} \|UA\vec{x}\|_2 &= \max_{\|\vec{x}\|_2=1} \sqrt{\vec{x}^T A^T U^T U A\vec{x}} \\ &= \max_{\|\vec{x}\|_2=1} \sqrt{\vec{x}^T A^T A\vec{x}} \\ &= \sqrt{\lambda_{max}(A^T A)} = \sigma_{max}(A) = \|A\|_2\end{aligned}$$

Therefore, we may also observe similar properties in Frobenius Norm, where:

$$\|A\|_2 = \|U_1 A\|_2 = \|AU_2\|_2 \text{ for orthonormal matrices } U_1, U_2$$

Chapter 6

Low-Rank Approximation

6.1 Discussion of L-RA

Suppose we have some matrix $A \in \mathbb{R}^{m \times n}$, where storing this matrix would take significant computational resources. Is there a method to store an approximation of this matrix, such that the approximation is more efficient to store for, and we would thus reduce computational cost of storage?

We have discussed the possibility of such technique within the previous lecture, where we have solved the optimization problem:

$$\max_{\|\vec{x}\|_2=1} \|A\vec{x}\|_2$$

Today, we will discuss the extension of such solution: Low-Rank Approximation.

LRA would be characterized from the optimization problem:

$$\operatorname*{argmin}_{B \in \mathbb{R}^{m \times n}, rk(B)=k} \|A - B\|_F$$

where we may see a contextually similar optimization problem:

$$\operatorname*{argmin}_{B \in \mathbb{R}^{m \times n}, rk(B)=k} \|A - B\|_2$$

Let us solve the low-rank approximation of matrices based on their Frobenius norm and L2 norm.

6.1.1 The LRA Optimization Problem on Spectral Norm

Problem.

$$\operatorname*{argmin}_{B \in \mathbb{R}^{m \times n}, rk(B)=k} \|A - B\|_2$$

Solution.
Keep in mind that our proof proceeds in the direction of:

- Finding a possible upper bound
- Finding the achievability of such upper bound
- Finding the legitimacy of such upper bound

Now, let us move on with the subparts of a proof.

Finding a Possible Upper Bound.
Let us define from the SVD of A:

$$A_k = \sum_{i=1}^{k} \sigma_i \vec{u_i} \vec{v_i}^T$$

Then, the difference between such approximation and A would be:

$$\|A - A_k\|_2 = \|A_n - A_k\|_2 = \left\| \sum_{i=k+1}^{n} \sigma_i \vec{u_i} \vec{v_i}^T \right\|_2$$

which, following along the optimization problem phrased last lecture, we would obtain the solution to be:

$$\left\| \sum_{i=k+1}^{n} \sigma_i \vec{u_i} \vec{v_i}^T \right\|_2 = \sigma_{k+1}(A)$$

Finding the Legitimacy (and Achievability) of Upper Bound.
Then, let us show that for every B where $rk(B) \leq k$, the difference $\|A - B\|_2$ is larger than our current possible minimum, σ_{k+1}. By proving so we secure the result of maximization is as illustrated above.
Let us first define $\vec{w}$ that,

$$\|A - B\|_2 = \max_{\|\vec{w}\|=1} \|(A - B)\vec{w}\|_2$$

Where, if $\vec{w} \in N(B)$, then we would be able to simplify the above optimization problem into something easier to solve: along such above constraint, let us note that,

$$\|A - B\|_2^2 \geq \|(A - B)\vec{w}\|_2^2 = \|A\vec{w}\|_2^2$$

And decomposing A into SVD form, which contains orthonormal matrices,

$$\|A - B\|_2^2 \geq \|U\Sigma V^T \vec{w}\|_2^2 = \|\Sigma V^T \vec{w}\|_2^2$$

Provided that $\vec{w}$ is a unit vector, and V is invertible, we can guarantee that

$$\exists \vec{a}, V\vec{a} = \vec{w}$$

Therefore, let us now substitute such value in, and we will obtain that

$$\begin{aligned} \|A - B\|_2^2 &\geq \|\Sigma V^T \vec{w}\|_2^2 \\ &= \|\Sigma \vec{a}\|_2^2 \\ &= \vec{a}^T \Sigma^T \Sigma \vec{a} = \vec{a}^T \Lambda \vec{a} \end{aligned}$$

Back to the Achievability of Upper Bound, Final Step.
Let us come back to our assumption that $\vec{w} \in N(B)$, and extract more insights from it.
Now, we may define along the V of $A = U\Sigma V^T$ that:

$$V_{k+1} = \begin{bmatrix} \vec{v}_1 & \cdots & \vec{v}_{k+1} \end{bmatrix}$$

where along the property of SVD we understand that $rk(V_{k+1}) = k + 1$.
To find some $\vec{w}$ such that $\vec{w} \in N(B) \wedge \vec{w} \in \mathcal{R}(V_{k+1})$, we should notice that:

1. Along rank-nullity theorem, $\dim(N(B)) \geq n - k$.
2. Along above description, $\dim(\mathcal{R}(V_{k+1})) = k + 1$

At this point, we must note that the achievability in our proposed upper bound exists in the possibility of having such $\vec{w}$. Therefore, **the achievability of upper bound is protected by the existence of $\vec{w}$, which we must guarantee**.
These two subspaces of $\mathbb{R}^n$ must have overlap. Therefore, there must exist some vector that belongs to both subspaces. This guarantees the existence of our $\vec{w}$.
Let us now follow along, and revisit the derivation that $\vec{a} = V^T\vec{w}$ on the change of basis.
Then, here we perform the final computations:

$$\begin{aligned}
\vec{a} &= V^T\vec{w} \\
&= V^T(\sum_{i=1}^{k+1} a_i\vec{V_i}) \\
&= \begin{bmatrix} a_1 & \cdots & a_{k+1} & 0 & \cdots & 0 \end{bmatrix}^T \\
\|A-B\|_2^2 \geq \vec{a}^T\Lambda\vec{a} &= \sum_{i=1}^{k+1} a_i^2\Lambda_i \\
&\geq \sum_{i=1}^{k+1} a_i^2\Lambda_{k+1} = \Lambda_{k+1}\sum_{i=1}^{k+1} a_i^2 \\
&= \Lambda_{k+1} = \sigma_{k+1}^2
\end{aligned}$$

Note, $\|\vec{a}\|_2 = \|\vec{w}\|_2 = 1$, which allows us to use the analysis of Rayleigh Coefficient on the last aligned equation.
We have thus established that the minimal possible expression of our minimization problem is σ_{k+1}^2

6.1.2 The LRA Optimization Problem on Frobenius Norm

Problem.

$$\underset{B\in\mathbb{R}^{m\times n}, rk(B)=k}{\operatorname{argmin}} \|A-B\|_F$$

Solution.
Let us consider A_k as priorly defined to be the solution of optimization again.

$$\begin{aligned}
\|A\|_F^2 &= \|U\Sigma V^T\|_F^2 \\
&= \|\Sigma\|_F^2 = \sum_{i=1}^{n} (\sigma_i(A))^2 \\
\|A-A_k\|_F^2 &= \sum_{i=k+1}^{n} (\sigma_i(A))^2 \\
\|A-B\|_F^2 &= \sum_{i=1}^{n} (\sigma_i(A-B))^2
\end{aligned}$$

Note that in above, we are expressing the singular values of difference matrices, not multiplying a singular value by a matrix.

Now, following the logic before, we want to prove a lowerbound:

> Prove that $\forall B \in \mathbb{R}^{m\times n}\ s.t.\ rk(B) = k, \|A-B\|_F \geq \|A-A_k\|_F$

Using the aligned equations above, let us find an equivalent proof prompt,

> Prove that $\forall i, \sigma_i(A-B) \geq \sigma_{k+i}(A)$

Note once again that these are singular values of different matrices. Where, along spectral norm's definition, we can also obtain that,

$$\sigma_{k+i}(A) = \|A - A_{k+i-1}\|_2$$

translated in text stating that A_{k+i-1} is the best rank $(k+i-1)$ approximation to A in spectral norm sense.
Now, let us return to the above equivalent prompt and define such that $C := A - B$.

$$\begin{aligned}\sigma_i(A-B) &= \sigma_i(C) \\ &= \|C - C_{i-1}\|_2 = \|A - B - C_{i-1}\|_2\end{aligned}$$

Pay attention to the rank of these matrices.

- B must be some rank-k (or less) matrix, as we are looking for B to be a k-rank approximation of A.
- C_{i-1} would have some rank less than $i-1$.
- Define the combination $D = B + C_{i-1}$, $rank(D) \leq k+i-1$

We would thus reuse the equation addressed before and state,

$$\sigma_i(A-B) = \sigma_i(A-D)$$

Using the solution from spectral norm side proof of LRA, for any matrix $D = B + C_{i-1}$ with at most rank $k+i-1$,

$$\begin{aligned}\sigma_i(A-B) &= \|A - (B + C_{i-1})\|_2 \\ &\geq \|A - A_{k+i-1}\|_2 = \sigma_{k+i}(A)\end{aligned}$$

Chapter 7

Vector Calculus

The motivation of vector calculus is to work with linear algebra problems more analytically, in more interpretable methods.
The tools that help to optimize linear algebra expression is the calculus of vector: vector calculus.

7.1 Function Expansion: Taylor Series

Let's start with functions, the basic of calculus.

Definition 7.1.1. Scalar Valued Function of Vector

Let a function f be such that:

$$f(\vec{x}) : \mathbb{R}^n \to \mathbb{R}$$

Such a function is by definition a scalar valued function of vector.

Then, we may begin our effort to generalize the derivative of such functions, as follows.

First of all, let us recap Calculus II knowledge of polynomial expansion of some fucntion:

Theorem 7.1.1. Taylor's Theorem (Expansion)

Let f be a scalar valued function for scalar:

$$f(x) : \mathbb{R} \to \mathbb{R}, x_0 \in \mathbb{R}$$

Then, Taylor expansion allows us to write such that:

$$f(x_0 + \Delta x) = f(x_0) + \left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{x=x_0} \Delta x + \cdots$$

Essentially,

$$f(x_0 + \Delta x) = \sum_{i=0}^{n} \frac{f^{(i)}(x_0)(\Delta x)^i}{i!}$$

where we call the Taylor expansion towards the n^{th} degree term be an n^{th} order Taylor expansion.

What is the purpose of introducing Taylor's expansion here? The fact is, we may approximate a value $f(x_0 + \Delta x)$ based on $f(x)$ along the product of Δx and a derivative. **The derivative affects how large the "perturbation" is that occurs in the approximation of** $f(x_0 + \Delta x)$. Similar logic is applied for upcoming n-order derivatives.

Then, along the above reintroduction of Taylor's Series, let us look at a vector edition of Taylor's Theorem:

Theorem 7.1.2. Taylor's Theorem for Vectors

Let f be a scalar valued function for vector.
Then, we may express that:

$$f(\vec{x_0}+\Delta\vec{x}) = f(\vec{x_0}) + \frac{\partial f}{\partial x}\bigg|_{\vec{x}=\vec{x_0}} \Delta\vec{x} + \frac{1}{2!}(\Delta\vec{x})^T \nabla^2 f(\vec{x})\bigg|_{\vec{x}=\vec{x_0}} \Delta\vec{x}$$

We usually stop at the second order approximation when using Taylor's Theorem for vectors, because this is usually a good equilibrium point for computational precision and simplification.

7.2 Function Expansion: Derivative of Vector Functions

In addition, we define the derivative of vectors as follows:

Definition 7.2.1. Derivatives of Scalar Valued Vector Function

Here, we observe that the first order derivative of such functions are row vectors, and the second order derivative is a matrix called **Hessian**:

$$\frac{\partial f}{\partial x} = (\nabla_{\vec{x}} f)^T = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_n} \end{bmatrix}$$

$$(\nabla^2 f(\vec{x}))_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

$$\nabla^2 f(\vec{x}) \in \mathbb{S}^n$$

Note that it is only for convex functions where the symmetric argument of Hessian applies.
This is beacuse:

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{}{\partial x_i}\frac{\partial f}{\partial x_j}$$

where the order of variables put matters for functions.
The reason why convex functions can host the symmetric Hessian is because of Clairaut's Theorem (not explicitly mentioned in lecture):

Theorem 7.2.1. Clairaut's Theorem

If the second partial derivatives of a function are continuous, then the order of differentiation is immaterial.
Alternatively (quoted from Purdue University),

> Let $f : \mathbb{R}^2 \to \mathbb{R}$ have all partial derivatives up to second derivative be continuous near (a, b), then:
>
> $$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x}$$

7.2.1 Examples of Polynomial Expansion

Now, let us provide an example in approximating a scalar valued vector function via Taylor's Theorem:

Explain 7.2.1. Polynomial Expansion of 2-Norm

Let us attempt to find a Taylor Expansion for the function

$$f(\vec{x}) = \|\vec{x}\|_2^2$$

We will define the level sets of this function as:

$$\{\vec{x} \big| f(\vec{x}) = C\}$$

which would be circles centered at the origin with radius $\sqrt{C}$.
The gradient of such function would be:

$$\nabla_{\vec{x}} f = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix}$$

And the Hessian:

$$H_{\vec{x}} f = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Therefore, upon Taylor's Theorem,

$$\begin{aligned} f(\vec{x_0} + \Delta\vec{x}) &= f(\vec{x_0}) + (\nabla_{\vec{x}} f)^T \Delta\vec{x} + \frac{1}{2!}(\Delta\vec{x})^T H_{\vec{x}} f \Delta\vec{x} \\ &= x_{01}^2 + x_{02}^2 + 2x_{01}\Delta x_1 + 2x_{02}\Delta x_2 + (\Delta x_1)^2 + (\Delta x_2)^2 \\ &= (x_{01} + \Delta x_1)^2 + (x_{02} + \Delta x_2)^2 = \|\vec{x_0} + \Delta\vec{x}\|_2^2 \end{aligned}$$

Since the Hessian is symmetric, we can perform a lot of interesting mathematics with it.
A third order derivative is known as a tensor, but this is out of scope for EECS 127.

Let's look at some more examples of Taylor Expansion:

Explain 7.2.2. Polynomial Expansion of Euclidean Inner Product

Let us find a Taylor Expansion to the function:

$$f(\vec{x}) = \vec{x}^T\vec{a}, \vec{a} \in \mathbb{R}^n$$

The gradient of such function is then,

$$\nabla_{\vec{x}} f = \vec{a}$$

Then, the Hessian of such function would be a zero matrix.
Therefore,

$$\begin{aligned} f(\vec{x_0} + \Delta\vec{x}) &= f(\vec{x_0}) + (\nabla_{\vec{x}} f)^T \Delta\vec{x} \\ &= \sum_{i=1}^{n} a_i(x_{0i} + \Delta x_i) = (\vec{x_0} + \Delta\vec{x})^T\vec{a} \end{aligned}$$

7.2.2 Matrix in Vector Calculus

Let us discuss how the roles of matrices are when we take the derivative of a scalar-valued function by a vector.

Explain 7.2.3. Derivatives of Matrix-Involving Inner Product

Let us find the derivative expressions to the function:

$$f(\vec{x}) = \vec{x}^T A\vec{x}, A \in \mathbb{R}^{n\times n}$$

We can derive the gradient as follows:

$$\begin{aligned}
\frac{\partial}{\partial x_i}\vec{x}^T A\vec{x} &= \frac{\partial}{\partial x_i}\vec{x}^T \begin{bmatrix}\vec{A_1} & \cdots & \vec{A_n}\end{bmatrix}\vec{x} \\
&= \frac{\partial}{\partial x_i}\begin{bmatrix}\vec{x}\cdot\vec{A_1} & \cdots & \vec{x}\cdot\vec{A_n}\end{bmatrix}\vec{x} \\
&= \frac{\partial}{\partial x_i}\sum_{k=1}^{n}\sum_{j=1}^{n} A_{j,k}x_k x_j \\
&= \frac{\partial}{\partial x_i}\left(\sum_{j=1}^{n} A_{j,i}x_i x_j + \sum_{k=1}^{n} A_{i,k}x_i x_k\right) \\
&= \sum_{j=1}^{n} A_{j,i}x_j + \sum_{k=1}^{n} A_{i,k}x_k = (\vec{A^T})_i \cdot \vec{x} + \vec{A_i}\cdot\vec{x} \\
\frac{\partial}{\partial x}\vec{x}^T A\vec{x} &= (A + A^T)\vec{x}
\end{aligned}$$

And, uh, Hessians.

$$\nabla^2 f(\vec{x}) = \frac{\partial}{\partial x}(A + A^T)\vec{x} = A + A^T$$

Then, let us define the derivative of a vector-valued function with respect to some vector variable:

Definition 7.2.2. Jacobian (Derivative) of Vector-Valued Vector Function

For some function $f(\vec{x}) = \vec{y}$ such that:

$$f : \mathbb{R}^n \to \mathbb{R}^m$$

Then,

$$J = \begin{bmatrix}\frac{\partial \vec{f}}{\partial x_1} & \cdots & \frac{\partial \vec{f}}{\partial x_n}\end{bmatrix}$$

Explain 7.2.4. The Jacobian of Polar-Cartesian Coordinate Translator

Let f be the function:

$$f(\vec{v}) = \begin{bmatrix} r\cos(\theta) \\ r\sin(\theta)\end{bmatrix}$$

Then the Jacobian of it would be:

$$J = \begin{bmatrix}\cos(\theta) & -r\sin(\theta) \\ \sin(\theta) & r\cos(\theta)\end{bmatrix}$$

whose determinant would then be $\det(J) = r$.

Last but not least, let us explore chain rule in matrix calculus:

Definition 7.2.3. Chain Rule

Let a function f be

$$f(\vec{x}) = g(h(\vec{x}))$$

Then, we may express that

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\mathrm{d}g}{\mathrm{d}y}\frac{\mathrm{d}h}{\mathrm{d}x}$$

Example.
Now, let's look at the least squares problem.
Originally, we perform this computation:

$$\begin{aligned} f(x) &= \|A\vec{x} - \vec{b}\|_2^2 \\ &= (A\vec{x} - \vec{b})^T ((A\vec{x} - \vec{b})) \\ &= \vec{x}^T A^T A\vec{x} - 2\vec{x}^T A\vec{b} + \vec{b}^T\vec{b} \\ \nabla_{\vec{x}} f(\vec{x}) &= (A^T A + A^T A)\vec{x} - 2A^T\vec{b} \\ &= 2A^T A\vec{x} - 2A\vec{b} \\ \vec{x}^* &= (A^T A)^{-1} A^T\vec{b} \end{aligned}$$

Instead, we may use chain rule:

$$\begin{aligned} \frac{\mathrm{d}f}{\mathrm{d}x} &= \frac{\mathrm{d}g}{\mathrm{d}y}\frac{\mathrm{d}h}{\mathrm{d}x} \\ &= (\nabla_{\vec{y}} g)^T \frac{\mathrm{d}}{\mathrm{d}x}(A\vec{x}) \\ &= 2(A\vec{x} - \vec{b})^T \cdot A \end{aligned}$$

We may thus obtain the gradient to be:

$$\nabla_{\vec{x}} f = 2A^T(A\vec{x} = \vec{b})$$

Chapter 8

The Extension of Vector Calculus

Note: This lecture's content will be shorter than other notes because half of the lecture was put on reviewing Lecture 7

8.1 The Main Theorem

The. Main.

Theorem 8.1.1. The Main Theorem

Theorem. Let Ω be an open subset of $\mathbb{R}^n$, and let function f be differentiable and such that

$$f : \mathbb{R}^n \to \mathbb{R}$$

then, for an optimal solution $\vec{x}^*$ that solves the optimization problem:

$$\min_{\vec{x} \in \Omega} f(\vec{x})$$

Then, the Main Theorem states that,

$$\frac{\mathrm{d}f}{\mathrm{d}\vec{x}}\Big|_{\vec{x}=\vec{x}^*} = 0$$

Proof. We realize that Ω is an open set, meaning $\vec{x}^* + \Delta\vec{x}$ can be guaranteed to be involved in Ω, for the definition of an open set is such that,

$$\forall x \in \Omega, \exists \epsilon > 0 \; (|x - y| < \epsilon \implies y \in \Omega)$$

Then, by the Taylor expansion and optimality assumption, we may state that:

$$f(\vec{x}^* + \Delta\vec{x}) = f(\vec{x}^*) + \frac{\mathrm{d}f}{\mathrm{d}x}\bigg|_{\vec{x}=\vec{x}^*} (\Delta\vec{x}) + \frac{1}{2}(\Delta\vec{x})^T \frac{\mathrm{d}^2 f}{\mathrm{d}x^2}\bigg|_{\vec{x}=\vec{x}^*} (\Delta\vec{x}) \geq f(\vec{x}^*)$$

Manipulating the above inequality,

$$\begin{aligned}
\frac{\mathrm{d}f}{\mathrm{d}x}\bigg|_{\vec{x}=\vec{x}^*} (\Delta\vec{x}) + \frac{1}{2}(\Delta\vec{x})^T \frac{\mathrm{d}^2 f}{\mathrm{d}x^2}\bigg|_{\vec{x}=\vec{x}^*} (\Delta\vec{x}) &\geq 0 \\
\frac{\mathrm{d}f}{\mathrm{d}x}\bigg|_{\vec{x}=\vec{x}^*} + \frac{sum(H.O.T.)}{\Delta\vec{x}} &\geq 0 \\
\lim_{\Delta\vec{x}\to 0} \left(\frac{\mathrm{d}f}{\mathrm{d}x}\bigg|_{\vec{x}=\vec{x}^*} + \frac{sum(H.O.T.)}{\Delta\vec{x}} \right) = \frac{\mathrm{d}f}{\mathrm{d}x}\bigg|_{\vec{x}=\vec{x}^*} &\geq 0
\end{aligned}$$

We may then provide a symmetric argument on the value $\vec{x}^* - \Delta\vec{x}$:

$$\begin{aligned} \left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{\vec{x}=\vec{x}^*}(-\Delta\vec{x}) + \frac{1}{2}(-\Delta\vec{x})^T\left.\frac{\mathrm{d}^2 f}{\mathrm{d}x^2}\right|_{\vec{x}=\vec{x}^*}(-\Delta\vec{x}) &\geq 0 \\ -\left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{\vec{x}=\vec{x}^*} + \frac{sum(H.O.T.)}{\Delta\vec{x}} &\geq 0 \\ \lim_{\Delta\vec{x}\to 0} -\left(\left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{\vec{x}=\vec{x}^*} + \frac{sum(H.O.T.)}{\Delta\vec{x}}\right) = -\left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{\vec{x}=\vec{x}^*} &\geq 0 \end{aligned}$$

$$\left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{\vec{x}=\vec{x}^*} \leq 0$$

Consequently, we reach the following conclusion:

$$\left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{\vec{x}=\vec{x}^*} \geq 0 \wedge \left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{\vec{x}=\vec{x}^*} \leq 0 \implies \left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{\vec{x}=\vec{x}^*} = 0$$

Note: H.O.T. means "Higher Order Terms" of Taylor Expansion, starting from the second-order term

8.2 Perturbation Analysis, Effect of Noise

In this problem, we consider the following concern:

> Let $A\vec{x} = \vec{y}$, where A is a square invertible matrix.
> Then, for some change in $\vec{y}$ (characterized as $\Delta\vec{y}$), how would this affect $\vec{x}$?
> In other words, we compare $\frac{\|\Delta\vec{x}\|_2}{\|\vec{x}\|_2}$ with $\frac{\|\Delta\vec{y}\|_2}{\|\vec{y}\|_2}$

Let us now set up the problem, by priorly noting what we know about the relationship between matrices, measurements, and perturbations:

$$\begin{aligned} A\vec{x} &= \vec{y} \\ A(\vec{x} + \Delta\vec{x}) &= \vec{y} + \Delta\vec{y} \end{aligned}$$

Then, we may manipulate the expressions to find that,

$$\begin{aligned} A\Delta\vec{x} = \Delta\vec{y} \implies \Delta\vec{x} &= A^{-1}\Delta\vec{y} \\ \|\Delta\vec{x}\|_2 = \|A^{-1}\Delta\vec{y}\|_2 &\leq \|A^{-1}\|_2\|\Delta\vec{y}\|_2 \end{aligned}$$

The last line is certified because the spectral norm of A^{-1} measures how much can it transform the norm of a vector.

Then, we may provide a minimization problem to attempt find some expression about the ratio of perturbation and actual measurement:

$$\min \frac{\|\Delta\vec{x}\|_2}{\|\vec{x}\|_2}$$

Now, let us observe the following work of upperbounding such denominator:

$$\begin{aligned} A\vec{x} = \vec{y} \implies \|A\vec{x}\|_2 &\leq \|A\|_2\|\vec{x}\|_2 \\ \|\vec{y}\|_2 &\leq \|A\|_2\|\vec{x}\|_2 \\ \frac{1}{\|\vec{x}\|_2} &\leq \frac{\|A\|_2}{\|\vec{y}\|_2} \end{aligned}$$

And therefore,

$$\begin{aligned}\frac{\|\Delta\vec{x}\|_2}{\|\vec{x}\|_2} &\leq \|A^{-1}\|_2\|\vec{\Delta y}\|_2\frac{\|A\|_2}{\|\vec{y}\|_2}\\ &= \|A^{-1}\|_2\|A\|_2\frac{\|\Delta\vec{y}\|_2}{\|\vec{y}\|_2}\\ &= \sigma_{\max}(A)\sigma_{\max}(A^{-1})\frac{\|\Delta\vec{y}\|_2}{\|\vec{y}\|_2} = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}\frac{\|\Delta\vec{y}\|_2}{\|\vec{y}\|_2}\end{aligned}$$

Consequently, we arrive at the conclusion (summary) of:

$$\frac{\|\Delta\vec{x}\|_2}{\|\vec{x}\|_2} \leq \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}\frac{\|\Delta\vec{y}\|_2}{\|\vec{y}\|_2}$$

where we call the fraction $\kappa(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$ the **condition number**.

Chapter 9

Ridge Regression

9.1 Perturbation Analysis Guides into Ridge Regression

In the concept of perturbation analysis, we ask that, for some system

$$A\vec{x} = \vec{y}$$

with a square, invertible A, how much would $\vec{x}$ change provided some small change $\vec{y} \to \vec{y} + \vec{\partial y}$?

Then, our solution (cited in Chapter 8, or Lecture 8) is as follows:

$$\frac{\|\vec{\partial x}\|_2}{\|\vec{x}\|_2} \leq \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)} \frac{\|\vec{\partial y}\|_2}{\|\vec{y}\|_2}$$

From whcich, we discover the characteristic of a matrix called **condition number**:

$$\frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$$

Let us connect the above discussion to the Least Squares Problem.
In the context of least squares problem, we are provided a more robust normal equation via the use of pseudoinverse:

$$(A^\dagger A)\vec{x} = A^T\vec{b}$$

As our least square system is sensitive to noise, we may investigate the condition number of A^TA to observe the system's change in its solution provided the perturbation in system measurements.
This is significant in computations. A high condition number makes a system's matrix highly unstable for numerical precisions to persist provided noise in measurements and systems. Such property is, in fact, demonstrated by the convergence warnings in Jupyter iPython notebooks!

But, provided the significance of condition number, we may also discuss ways to alleviate its problems. To reduce the condition number of some matrix, we may attempt to reduce the ratio of singular values via adding some multiple of identity matrix (λI) to it. This allows us to greatly reduce the condition number, shifting it away from instability and being less prone to variance in training data:

$$A^TA + \lambda I$$

As a side note, in CS189, we discover similar approaches that can help tackle training data that leads to singular covariance matrices.
Let us investigate below why adding a multiple of diagonal matrix does not make our regression problem have a very deviated solution (despite the fact we somehow alter the design matrix of our problem).

9.2 Ridge Regression

In the least squares problem, we have a formulated optimization problem of:

$$\min_{\vec{x}} \|A\vec{x} - \vec{b}\|_2$$

to offer insight to the optimization problem to prevent its divergence from true parameter value, is to present a new formulation of the problem:

$$\min_{\vec{x}} \|A\vec{x} - \vec{b}\|_2^2 + \lambda^2 \|\vec{x}\|_2^2$$

where, we regularize the least squares solution by stating that large solutions of $\vec{x}$ must lead to unidealistic increase in the cost (loss) function. We penalize a large least squares solution $\vec{x}$.

Here, using different norms in the regularizer will provide different properties. The L1 regularizer has the LASSO property (as outlined in DATA C100), while the L2 regularizer we use now provides a convex function as well as a closed-form solution unlike the L1 effort (once again, as outlined in DATA C100).

Let us first compute the gradient of such loss function:

$$\nabla_{\vec{x}} \|A\vec{x} - \vec{b}\|_2^2 + \lambda^2 \|\vec{x}\|_2^2 = 2A^T(A\vec{x} - \vec{b}) + 2\lambda^2 I\vec{x}$$
$$\vec{x}^* = \left(A^T A + \lambda^2 I\right)^{-1} A^T \vec{b}$$

The eigenvalues of $A^T A + \lambda^2 I$, meanwhile, would be the eigenvalues of $A^T A$ added the value λ^2 (by manipulating the definition of eigenvalues and eigenvectors).
This is also known as the shift property of eigenvalues:

Theorem 9.2.1. Shift Property of Eigenvalues

Let $\vec{v}$ be an eigenvector of $A^T A$.
Then, provided that:

$$A^T A\vec{v} = \mu\vec{v}$$

we may see that,

$$(A^T A + \lambda^2 I)\vec{v} = (\mu + \lambda^2)\vec{v}$$

Thus determine that:

> for any eigenpair of $A^T A$ being $(\mu, \vec{v})$, a corresponding eigenpair exists in $A^T A + \lambda^2 I$ being $(\mu + \lambda^2, \vec{v})$.

Therefore, small λ corresponds to less regularizaiton effort, and vice versa.

9.2.1 Development of Ridge Regression

Now, suppose we have a least square system where $\vec{x}$ has a smaller norm, where $\lambda I\vec{x} \sim \vec{0}$.
Then, by that close-to-zero property we observe in $\lambda I\vec{x}$, adding the information of λI into the least squares problem formulation would barely alter the original formulation too much, due to the close-to-zero-ness of ridge regression's current least squares solution.
This allows us to use larger λ (have greater regularization).
We have two ways of incorporating such information into the least squares problem now:

- Vertically concatenate λI below A.
- Use Ridge Regression's format, which uses a similar idea to preserve the original system as much as possible. We will prove later that this is exactly the first idea.

In the concatenation idea, our system of least squares problem is reformulated as:

$$\begin{bmatrix} A \\ \lambda I \end{bmatrix} \vec{x} = \begin{bmatrix} \vec{b} \\ \lambda I \vec{x} \end{bmatrix}$$

Let's use both block matrix and normal equation to further explore the idea of concatenating λI:

$$\begin{aligned} &\left(\begin{bmatrix} A^T & \lambda I \end{bmatrix} \begin{bmatrix} A \\ \lambda I \end{bmatrix} \right)^{-1} \begin{bmatrix} A^T & \lambda I \end{bmatrix} \begin{bmatrix} \vec{b} \\ \lambda I \vec{x} \end{bmatrix} \\ &= (A^T A + \lambda^2 I)(A^T \vec{b} + \lambda^2 I \vec{x}) \\ &\sim (A^T A + \lambda^2 I) A^T \vec{b} \end{aligned}$$

and we have therefore demonstrated the similarity between the two ideas of ridge regression.
It is noteworthy that the common notation of ridge regression does not use the notation λ^2 when addressing regularization parameter. We are doing so in the context of demonstrating how ridge regression is developed, through a simpler set of mathematical notations.

Once we expand the idea to adding weights for matrices, creating the system:

$$\begin{bmatrix} W_1 A \\ W_2 I \end{bmatrix} \vec{x} = \begin{bmatrix} W_1 \vec{b} \\ W_2 \vec{x_0} \end{bmatrix}$$

we end up with a new optimization problem:

$$\min_{\vec{x}} \|W_1(A\vec{x} - \vec{b})\|_2^2 + \|W_2(\vec{x} - \vec{x_0})\|_2^2$$

which we call the **Tikhonov Regularization** technique.
Note: $\vec{x_0}$ is an arbitrary piece of information.

9.3 Probabilistic Information from Ridge Regression

Suppose that we instead now have probabilistic information:

$$(\vec{x_i}, y_i), \text{ where } y_i = g(x_i) + z_i$$

and, $z_i \sim \mathcal{N}(0, \sigma_i^2)$ is a noise defined on a Gaussian distribution.
And, suppose we have a linear model,

$$y_i = \vec{w}^T \vec{x} + z_i$$

Then, we may state that,

$$f_{z_i}(z_i) = \frac{\exp\left(-\frac{z_i^2}{2\sigma_i^2}\right)}{\sqrt{2\pi}\sigma_i}$$

and we attempt to learn the weights $\vec{w}$ to complete the model for some related problem.
Therefore, with multivariate Gaussian noise $\vec{z}$ and datapoint matrices X, we formulate this as a least square system,

$$X\vec{w} + \vec{z} = \vec{y}$$

9.3.1 Maximum Likelihood Estimation and Maximum A-Posteriori

Now, we may attempt to estimate the parameters that makes the observed data most likely.
This is related to the technique of Maximum Likelihood Estimation. Let me shamelessly copy a segment of my CS189 notes here to provide a brief explanation:

Explain 9.3.1. MLE from CS189

Suppose that we go back to the coin flip example (just like 126 does), where heads appear with a probability p (and otherwise for tails).
Then, statisticians world ask, provided the real data of coin, what value of p (the parameter of coin flip probability distribution) is closest to its true inherent value.
Let us suppose that the number of heads we obtain is a discrete random variable, $X \sim Binomial(n, p)$:

$$\mathbb{P}(X = x) = \binom{n}{x} p^x (1-p)^{n-x}$$

Then, let us propose that the real data presents k heads, and we would defined the Likelihood function $\mathcal{L}$ as:

$$\mathcal{L}(p) = \mathbb{P}(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

where it is a function of distribution parameters.

Then, Maximum Likelihood Estimation (MLE) is the method of estimating the parameters of a statistical distribution by picking the parameters that maximize $\mathcal{L}$. Furthermore, it would be a method of density estimation, where we estimate some probability density function from the provided dataset.

In this case, we are performing MLE to obtain weight $\vec{w}$ provided the Gaussian noise $\vec{z}$:

Explain 9.3.2. Solving Maximum Likelihood Estimation Phrased Optimization

Let us begin from the prompt:

$$\begin{aligned}
argmax_{\vec{w_0}} f(\vec{y}|\vec{w} = \vec{w_0}) &= argmax_{\vec{w_0}} \prod_{i=1}^{n} f(Y_i = y_i|\vec{w} = \vec{w_0}) \\
&= argmax_{\vec{w_0}} \prod_{i=1}^{n} f(z_i = y_i - \vec{x_i}^T\vec{w}|\vec{w} = \vec{w_0}) \\
&= argmax_{\vec{w_0}} \prod_{i=1}^{n} \frac{\exp\left(-\frac{(y_i - \vec{x_i}^T)^2}{2\sigma_i^2}\right)}{\sqrt{2\pi}\sigma_i} \\
&= argmax_{\vec{w_0}} \frac{1}{\sqrt{2\pi}^n} \exp\left(\sum_{i=1}^{n} \frac{-(y_i - \vec{x_i}^T\vec{w})}{2\sigma_i^2}\right) \prod_{i=1}^{n} \frac{1}{\sigma_i} \\
&= argmin_{\vec{w_0}} \sum_{i=1}^{n} \frac{(y_i - \vec{x_i}^T\vec{w})^2}{2\sigma_i^2} \\
&= argmin_{\vec{w_0}} \|S(X\vec{w_0} - \vec{y})\|_2^2
\end{aligned}$$

where,

$$S = \begin{bmatrix} \frac{1}{\sqrt{2}\sigma_1} & & \\ & \ddots & \\ & & \frac{1}{\sqrt{2}\sigma_n} \end{bmatrix}$$

and we end up on a weighted least squares setup.
Furthermore, upon IID uniform Gaussians, we will end up with $S = \frac{1}{\sqrt{2}\sigma_i}$.

Meanwhile, the Maximum A-Posteriori approach attempts to provide a priori distribution on $\vec{w}$.
We solve the optimization problem of:

$$argmax_{\vec{w_0}} f(\vec{w} = \vec{w_0}|\vec{y})$$

As you may observe, both techniques offer the most likely weights for some condition, but the conditions are framed quite differently:

- MLE find the most likely $\vec{w}$ to **lead to current observation**.
- MAP find the most likely $\vec{w}$ **given the current observation** that occurs.

9.3.2 A Personal Learning on MLE vs MAP

While this is not entirely the focus of exam scope, I'd like to address the difference between MLE and MAP with a few lines.
First of all, We may notice the relationship between objective function of MLE and MAP to have the following relationship:

$$f(\vec{y}|\vec{w} = \vec{w_0}) = \pi_y f(\vec{w} = \vec{w_0}|\vec{y})$$

If we consider the priori π_y to be uniform (for example, all possible components of $\vec{y}$ address an event of uniform distribution like coin flips or dice rolls), then maximizing the MLE objective function indeed maximizes the MAP objective function (as they are scalar multiples of each other). This means MLE and MAP, under a uniform priori for any possible component of $\vec{y}$, are equal optimization problems.

Chapter 10

Convexity

10.1 Convex Set

Let us first define a convex set geometrically,

Definition 10.1.1. Convex Set

A set $C \subseteq \mathbb{R}^n$ is convex if the line segment between any two points in C is contained in C.
For example, convex polygons resemble a convex set of points.

Then, to express such concept algebraically, we would arrive at the algebraic addenda to definition:

Definition 10.1.2. Convex Set in Algebraic Perspective

The set of points within a line segment terminated by points $\vec{x}, \vec{y} \in C$ would be expressable as

$$L_{\vec{x},\vec{y}} := \{\theta\vec{x} + (1-\theta)\vec{y} | \theta \in [0,1]\}$$

And, a set C is convex iff

$$\forall \vec{x}, \vec{y} \in C, L_{\vec{x},\vec{y}} \subseteq C$$

Upon the algebraic interpretation of convexity in sets, we may also define convexity of combinations:

Definition 10.1.3. Convex Combination

The combination $\vec{x} = \sum_i \theta_i \vec{x_i}$ is a convex combination iff it satisfies the following qualities:

1. $\forall i, \theta_i \geq 0$
2. $\sum_i \theta_i = 1$

Along which, we may then define convexity on numerous mathematical objects. Another example is shown here,

Definition 10.1.4. Convex Hull

The convex hull of some set $S \subseteq \mathbb{R}^n$ is the set of all convex combinations of members of S.
This formulates a concept very similar to "span".

10.1.1 Proof of Convexity

Sets of mathematical objects (like matrices) can be convex as well. Let us demonstrate with the following proofs, where we in unison practice to prove convexity of sets.

Explain 10.1.1. The Convexity of Set of Rank-1 Matrices

Problem. Let us find a set of all rank-1 Matrices:

$$\{M_1 = \{A \in \mathbb{R}^{m\times n} | rk(A) = 1\}\}$$

decide whether it is convex or not.

Proof. Let us observe whether for any arbitrary matrices $X, Y \in M_1$, the "line segment" along these matrices are included in M_1.
In other words, we ask, whether the matrix $\theta X + (1-\theta)Y$ belongs to M_1.
The answer would be no. Suppose X_0 and Y_0 each have a basis $\{\vec{x}\}$, $\{\vec{y}\}$, and let $\vec{x}$ and $\vec{y}$ be linearly independent vectors.

$$X_0 = \begin{bmatrix} 2\vec{x} & \vec{x} \end{bmatrix}, Y_0 = \begin{bmatrix} \vec{y} & \vec{y} \end{bmatrix}$$

Consequently,

$$\forall \theta \in [0,1], rk(\theta X_0 + (1-\theta)Y_0) > 1$$

And thus,

$$\neg(\forall X, Y \in C, L_{\vec{x},\vec{y}} \subseteq C)$$

Similalry, let's explore another example of proof/disproof for convexity:

Explain 10.1.2. The Convexity of Set of PSD Matrices

Problem. Let us find a set of all rank-1 Matrices:

$$\{P = \{A | A \succcurlyeq 0 \wedge A \in \mathbb{S}^n\}\}$$

decide whether it is convex or not.

Proof. To prove that a matrix is positive-semidefinite by definition, we need only prove it suits one of the three definitions of positive-semidefiniteness.
For most convenience, I'd like to use the definition that:

$$A \succcurlyeq 0 \iff \forall \vec{x} \in \mathbb{R}^n, \vec{x}^T A \vec{x} \geq 0$$

Now, let me work with two arbitrary matrices $X, Y \in P$, and attempt to prove the following equivalent problem:

Equivalent Prompt. Prove that $\forall \theta \in [0,1], \theta X + (1-\theta)Y \in P$.

Fortunately, we may observe that,

$$\forall \vec{x}, \vec{x}^T(\theta X + (1-\theta)Y)\vec{x} = \vec{x}^T\theta X\vec{x} + \vec{x}^T(1-\theta)Y\vec{x}$$
$$\forall \vec{x}, \vec{x}^T X\vec{x} \geq 0 \wedge \vec{x}^T Y\vec{x} \geq 0 \implies \forall \vec{x}, \vec{x}^T\theta X\vec{x} + \vec{x}^T(1-\theta)Y\vec{x} \geq 0$$

Furthermore,

$$\begin{aligned}(\theta X + (1-\theta)Y)^T &= \theta X^T + (1-\theta)Y^T \\ &= \theta X + (1-\theta)Y\end{aligned}$$

Therefore,

$$\forall \theta \in [0,1], \theta X + (1-\theta)Y \succcurlyeq 0 \wedge \forall \theta \in [0,1], \theta X + (1-\theta)Y \in \mathbb{S}^n$$

then by definition it must belong to the set P. Via this subproof, we conclude that P is convex.

10.1.2 Hyperplanes

Hyperplanes are prevalent mathematical objects in Computer Science applications (such as the infamous Machine Learning), and its formal definition follows:

Definition 10.1.5. Hyperplane

Hyperplanes are sets of points that have the two following alternative forms:

$$H = \{\vec{x} \in \mathbb{R}^n | \vec{a}^T \vec{x} = \vec{b}\}$$
$$H = \{\vec{x} \in \mathbb{R}^n | \vec{a}^T (\vec{x} - \vec{x_0}) = 0\}, \vec{b} = \vec{a}^T \vec{x_0}$$

in the above notes, you may see the equivalence of two forms via adjusting the names of variables in the set-builder notation.

Upon defining it, let us then discuss the properties of such mathematical object. Is the hyperplane per se a convex set?

Explain 10.1.3. Convexity of Hyperplane

Proof. For a hyperplane defined as,

$$H = \{\vec{x} \in \mathbb{R}^n | \vec{a}^T (\vec{x} - \vec{x_0}) = 0\}$$

Let us take two arbitrary points $\vec{y}, \vec{z} \in H$.
Then, for some $\theta \in [0, 1]$,

$$\begin{aligned} \vec{a}^T(\theta\vec{y} + (1-\theta)\vec{z} - \vec{x_0}) &= \vec{a}^T\theta\vec{y} + \vec{a}^T(1-\theta)\vec{z} - \vec{a}^T\vec{x_0} \\ &= \theta\vec{a}^T\vec{x_0} + (1-\theta)\vec{a}^T\vec{x_0} - \vec{a}^T\vec{x_0} = 0 \end{aligned}$$

Therefore, by definition,

$$\forall \vec{y}, \vec{z} \in H, L_{\vec{y},\vec{z}} \subseteq H$$

and we have proven the convexity of hyperplanes by its definition.

Furthermore, we can define specific subsets of a space defined by some hyperplane:

Definition 10.1.6. Halfspace

For a hyperplane

$$H = \{\vec{x} \in \mathbb{R}^n | \vec{a}^T (\vec{x} - \vec{x_0}) = 0\} \in \mathbb{R}^n$$

it partitions the real space $\mathbb{R}^n$ into two halves, which we formally call the halfspace.
The halfspaces can then be defined as follows:

$$H_+ = \{\vec{x} \in \mathbb{R}^n | \vec{a}^T (\vec{x} - \vec{x_0}) \geq 0\}$$
$$H_- = \{\vec{x} \in \mathbb{R}^n | \vec{a}^T (\vec{x} - \vec{x_0}) \leq 0\}$$

We call H_+ the positive halfspace, and H_- the negative halfspace.

And, along this concept of halfspace, we have developed the following theorem:

Theorem 10.1.1. Separating Hyperplane Theorem

Let $C, D \in \mathbb{R}^n$ be nonempty disjoint convex sets, then

$$\exists \vec{a} \neq \vec{0}, \vec{x_0} \in \mathbb{R}^n \big(\forall \vec{x} \in C, \vec{a}^T (\vec{x} - \vec{x_0} \geq 0) \wedge \forall \vec{x} \in D, \vec{a}^T (\vec{x} - \vec{x_0} \leq 0)\big)$$

In other words, there must exist a hyperplane that separates the members of C and D.

Such theorem is incredibly relevant to linear classifiers (specifically, this is an insight extracted from SVMs in CS189).
On a side note, the proof of this theorem is essentially the mathematical work to construct such a separating hyperplane, whose normal vector would ideally be some vector between a point of C and a point of D and the hyperplane crosses the midpoint of $\overline{pq}$.

10.2 Convex Functions

Once again, we will begin with a definition:

Definition 10.2.1. Convex Functions

A scalar-valued function $f : \mathbb{R}^n \to \mathbb{R}$ is convex iff the following qualities are satisfied:

- The domain of f is a convex set.
- The function f obeys the Jensen's Inequality:

$$\forall \vec{x}, \vec{y} \in Domain(f), \theta \in [0, 1], f(\theta\vec{x} + (1-\theta)\vec{y}) \leq \theta f(\vec{x}) + (1-\theta)f(\vec{y})$$

Essentially, the Jensen's Inequality states that:

> Any line segment between two points $(x, f(x)), (y, f(y))$ of a convex function would not be below the function curve between those two points $(x, f(x)), (y, f(y))$.

Chapter 11

Convex Optimization Problems

11.1 Convex Functions, Continued

We have defined last time that

Definition 11.1.1. Convex Functions

A scalar-valued function $f : \mathbb{R}^n \to \mathbb{R}$ is convex iff the following qualities are satisfied:

- The domain of f is a convex set.
- The function f obeys the Jensen's Inequality:

$$\forall \vec{x}, \vec{y} \in Domain(f), \theta \in [0,1], f(\theta\vec{x} + (1-\theta)\vec{y}) \leq \theta f(\vec{x}) + (1-\theta)f(\vec{y})$$

Essentially, the Jensen's Inequality states that:

> Any chord between two points $(x, f(x)), (y, f(y))$ of a convex function would not be below the function curve between those two points $(x, f(x)), (y, f(y))$.

Concave functions are, on the other hand, the version of convex functions where many theories apply on maximization. For example, while convex functions' critical points are directly their global minimum, concave functions' critical points are directly their global maximum.
By studying convex optimization, we equivalently study many aspects of concave optimization. (Moreover, inverting the sign of some concave function f already make a convex objective function).
Interestingly, linear functions are both concave and convex: they obey Jensen's Inequality by having the chord overlap with the objective function curve itself.

Now, let us discuss the properties of functions further:

Definition 11.1.2. Epigraph

An epigraph of a function f is the set:

$$epi(f) = \{(\vec{x}, t) \big| \vec{x} \in dom(f), f(\vec{x}) \leq t\}$$

Reviewing the above definition, we discover that an epigraph is the space above the objective function value, the "volume above curve".

Such figure is related to the convexity of a function:

Theorem 11.1.1. Epigraph and Convexity

Theorem. Function f is convex iff $epi(f)$ is a convex set.

Proof. Let us perform proof(s) for such theorem.
Direction 1: f is convex $\implies$ $epi(f)$ is convex.
A set C is convex if:

$$\forall \vec{x}, \vec{y} \in C, L_{\vec{x},\vec{y}} = \{\theta\vec{x} + (1-\theta)\vec{y} | \theta \in [0,1]\} \subseteq C$$

A function f is convex if:

$$\forall \vec{x}, \vec{y} \in Domain(f), \theta \in [0,1], f(\theta\vec{x} + (1-\theta)\vec{y}) \leq \theta f(\vec{x}) + (1-\theta)f(\vec{y})$$

Let us take two arbitrary points from $epi(f)$ be $(\vec{x}, t_x), (\vec{y}, t_y)$ such that $t_x \geq f(\vec{x})$ and $t_y \geq f(\vec{y})$.
Let us consider some point in $L_{\vec{x},\vec{y}}$ and see if it belongs to C:

$$\theta(\vec{x}, t_x) + (1-\theta)(\vec{y}, t_y) = (\theta\vec{x} + (1-\theta)\vec{y}, \theta t_x + (1-\theta)t_y)$$

Where, via the property of a convex function, we know that $\theta\vec{x} + (1-\theta)\vec{y}$ is included in its convex domain.
Furthermore, via Jensen's Inequality

$$\begin{aligned} \theta t_x + (1-\theta)t_y &\geq \theta f(\vec{x}) + (1-\theta)f(\vec{y}) \\ &\geq f(\theta\vec{x} + (1-\theta)\vec{y}) \end{aligned}$$

Therefore, by definition, a convex function f has a convex epigraph $epi(f)$.
Direction 2: $epi(f)$ is convex $\implies$ f is convex.
Take two arbitrary points in $epi(f)$ to be $(\vec{x}, t_x = f(\vec{x})), (\vec{y}, t_y = f(\vec{y}))$, where by the definition of convexity, we realize that:

$$\forall \theta \in [0,1], (\theta\vec{x} + (1-\theta)\vec{y}, \theta t_x + (1-\theta)t_y) \in epi(f)$$

Since such point exists in an epigraph, it must infer that,

$$\theta t_x + (1-\theta)t_y = \theta f(\vec{x}) + (1-\theta)f(\vec{y}) \geq f(\theta\vec{x} + (1-\theta)\vec{y})$$

which is the Jensen's Inequality: what characterizes the definition of a convex function f.

This thus connects the definitions of convex sets and convex functions. But, ready or not, there are more definitions of convexity:

Definition 11.1.3. First Order Condition of Convexity

Let f be a differentiable function, then f is convex iff the following condition is satisfied:

$$\forall \vec{x}, \vec{y} \in Domain(f), f(\vec{y}) \geq f(\vec{x}) + \frac{\mathrm{d}f}{\mathrm{d}x}\Big|_{\vec{x}} \cdot (\vec{y} - \vec{x})$$

The implication of such definition is:

$$\nabla f(\vec{x}) = 0 \implies \forall \vec{y}, f(\vec{y}) \geq f(\vec{x}) + 0$$

Proof.
Direction 1: f is convex implies the first order condition of convexity
The definition of convex function allows us to state for some $\theta \in [0,1]$ that

$$\forall \vec{x}, \vec{y} \in Domain(f), \theta \in [0,1], f((1-\theta)\vec{x} + \theta\vec{y}) \leq (1-\theta)f(\vec{x}) + \theta f(\vec{y})$$

Rearranging the above inequality grants

$$\theta f(\vec{y}) \geq f((1-\theta)\vec{x} + \theta\vec{y}) - f(\vec{x}) + \theta f(\vec{x})$$
$$f(\vec{y}) \geq \frac{1}{\theta} f(\vec{x} + \theta(\vec{y} - \vec{x})) - \frac{1}{\theta} f(\vec{x}) + f(\vec{x})$$

Upon aligning the above equation with the definition of derivative, we may discover that:

$$\lim_{t \to 0} \frac{f(\vec{x} + \theta(\vec{y} - \vec{x})) - f(\vec{x})}{\theta(\vec{y} - \vec{x})} = \frac{\mathrm{d}f}{\mathrm{d}x}$$

Therefore, let us substitute the above derivative term into the function:

$$f(\vec{y}) \geq \frac{\mathrm{d}f}{\mathrm{d}x} \cdot (\vec{y} - \vec{x}) + f(\vec{x})$$

which is exactly the first order condition of convexity.
Direction 2: the first order condition of convexity implies f is convex.
Suppose we have two arbitrary points $\vec{x}, \vec{y}$, and that:

$$\vec{z} = \theta\vec{x} + (1-\theta)\vec{y}$$

Let us use the first order condition of convexity to state that,

$$f(\vec{x}) \geq f(\vec{z}) + f'(\vec{z})(\vec{x} - \vec{z})$$
$$f(\vec{y}) \geq f(\vec{z}) + f'(\vec{z})(\vec{y} - \vec{z})$$

Then,

$$\begin{aligned} \theta f(\vec{x}) + (1-\theta) f(\vec{y}) &\geq f(\vec{z}) + \theta f'(\vec{z})(\vec{x} - \vec{z}) + (1-\theta) f'(\vec{z})(\vec{y} - \vec{z}) \\ &= f(\vec{z}) + \theta f'(\vec{z})(\theta\vec{x} - \theta\vec{z} + (1-\theta)\vec{y} - (1-\theta)\vec{z}) \\ &= f(\vec{z}) \end{aligned}$$

Upon the first order condition, we also have a second order condition of convexity:

Definition 11.1.4. Second Order Condition of Convexity

Let f be a twice-differentiable function, then f is convex iff the following condition is satisfied:

$$\nabla^2 f(x) \succcurlyeq 0$$

11.2 Convex Optimization Problem

Let us have some problem:

$$p^* = \min f_0(\vec{x}) \text{ s.t. } f_i(\vec{x}) \leq 0$$

A problem is a convex optimization problem if for all i, $f_i(x)$ is a convex function.

Chapter 12

Descent Methods

12.1 Strict Strong Convexity

Convexity also comes in different strictness. These more refined definition provide us flexibility and rigor in proofs. For example, we may first consider that, the zeroth order definition of convexity would be:

> For a function $f : \mathbb{R}^n \to \mathbb{R}$, the domain of f needs to be convex, and Jensen's Inequality applies to any two arbitrary points on the domain.

However, we fail to classify linear functions as either convex or concave. Linear functions were concluded to be both convex and concave, which is very confusing.

Definition 12.1.1. Strict Convexity

The zeroth order condition of strict convexity is altered to:

> For a function $f : \mathbb{R}^n \to \mathbb{R}$, the domain of f needs to be convex, and:
>
> $$f(\theta\vec{x} + (1-\theta)\vec{y}) < \theta f(\vec{x}) + (1-\theta)f(\vec{y})$$

The first order condition is altered to:

$$\forall \vec{x}, \vec{y} \in Domain(f), f(\vec{y}) > f(\vec{x}) + \frac{\mathrm{d}f}{\mathrm{d}x} \cdot (\vec{y} - \vec{x})$$

The second order condition is altered along a very similar logic:

$$\nabla^2 f(\vec{x}) > 0$$

And in summary, a strictly convex function must have a unique minimum (unlike, say, a ReLU function).

But just like humans, not only can convexity have strict variants, they can also have strong variants. Strong convexity implies strict convexity, and strict convexity implies convexity. These definitions are increasingly difficult to satisfy:

Definition 12.1.2. Strong Convexity

If f is differentiable, then for some $\mu > 0$, it is μ-strongly convex if domain of f is convex and

$$\forall \vec{x}, \vec{y} \in domain(f), f(\vec{y}) \geq f(\vec{x}) + \frac{\mathrm{d}f}{\mathrm{d}x} \cdot (\vec{y} - \vec{x}) + \frac{\mu}{2}\|\vec{y} - \vec{x}\|_2^2$$

The insight such variant provides is, for some μ-strongly convex function:

$$\nabla^2 f(x) \succcurlyeq \mu I \implies \nabla^2 f(x) - \mu I \succcurlyeq 0$$

The smallest possible eigenvalue of the Hessian of f is then μ. The larger such value μ is, the stronger its convex; in other words, the further the function is from being nonconvex.

12.2 Gradient Descent

As you might have learned in prior courseworks (during college, or, high school given Berkeley), there are many variants of Gradient Descent.
Let's think about the vanilla version of gradient descent first, as an unconstrained optimization problem:

12.2.1 Inventing Gradient Descent

$$p^* = \min_{\vec{x} \in \mathbb{R}^n} f_0(\vec{x})$$

The foundation of gradient descent algorithm is an "oracle" that guides us at the direction of some greatest descent along the provided function f.
Now, we should understand when does the "oracle" guide us to convergence, such that we do not get lost in the search space.

Let's find the "oracle" via some perspectives on linear perturbation:

$$f(\vec{x} + \Delta\vec{x}) = f(\vec{x}) + \frac{\mathrm{d}f}{\mathrm{d}x} \cdot (\Delta\vec{x}) + \text{H.O.T.}$$

If $\Delta\vec{x}$ is a good direction to perturb $\vec{x}$ to, then we wish as well that $f(\vec{x} + \Delta\vec{x}) < f(\vec{x})$.
Suppose that $\Delta\vec{x} = s\vec{v}$, where s is a real scalar and $\vec{v}$ is some direction-resembling vector.
Then, let us rewrite $f(\vec{x} + s\vec{v})$ as:

$$\begin{aligned} f(\vec{x} + s\vec{v}) &= f(\vec{x}) + s\frac{\mathrm{d}f}{\mathrm{d}x}\vec{v} \\ &= f(\vec{x}) + s\langle \nabla f(\vec{x}), \vec{v} \rangle < f(\vec{x}) \\ \langle \nabla f(\vec{x}), \vec{v} \rangle &< 0 \end{aligned}$$

This means $\nabla f(\vec{x})$ and $\vec{v}$ should be in opposite direction.
Furthermore, to maximize such above expression for finding a greatest descent, we may let

$$\vec{v} = -\nabla f(\vec{x})$$

and we conclude that the opposite direction of gradient is the great direction of descent– the "oracle".

Now, let us write up a draft for the gradient descent algorithm (which I will follow the minimal-energy principle of Physics and shamelessly copy from my notes in prior courseworks), that:

Definition 12.2.1. Gradient Descent Algorithm

Let the initial guess of minimum be $\vec{x_0}$.
Then, provided a stepsize η, and defining $\vec{x_k}$ to be the k^{th} guess of minimum, the iterative approach of algorithm follows as:

$$\vec{x}_{k+1} = \vec{x}_k - \eta \nabla f(\vec{x_k})$$

To get better constant stepsizes, we can perform hyperparamter tuning; or, solve the next subsection.

12.2.2 Finding η

Let us first define the function we perform gradient descent on, to demonstrate the process of finding η.
Let

$$f_0(\vec{x}) = \|A\vec{x} - \vec{b}\|_2^2$$
$$\nabla f_0(\vec{x}) = 2A^T(A\vec{x} - \vec{b})$$

The iterative rule of gradient descent on function f_0 is thus:

$$\begin{aligned}\vec{x}_{k+1} &= I\vec{x}_k - 2\eta A^T(A\vec{x} - \vec{b}) \\ &= (I - 2\eta A^T A)\vec{x}_k + 2\eta A^T\vec{b}\end{aligned}$$

Let us discuss the recursion in the above form.
First of all, we require that the matrix coefficient of guess term: $(I - 2\eta A^T A)$, to provide some scaling that does not consistently scale the vector larger, so to prevent divergence. In matrices, we may consider eigenvalue as the indicator of such divergence or convergence. We require that the eigenvalues of $(I - 2\eta A^T A)$ satisfy some condition such that it does not lead to a consistently larger guess.

Now, let us look at the guess rule:

$$\begin{aligned}\vec{x}_{k+1} - \vec{x}^* &= (I - 2\eta A^T A)\vec{x}_k + 2\eta A^T\vec{b} - (A^T A)^{-1}A^T\vec{b} \\ &= (I - 2\eta A^T A)\vec{x}_k + (2\eta A^T A - I)(A^T A)^{-1}A^T\vec{b} \\ &= (I - 2\eta A^T A)[\vec{x}_k - (A^T A)^{-1}A^T\vec{b}] \\ &= (I - 2\eta A^T A)[\vec{x}_k - \vec{x}^*]\end{aligned}$$

Solve the recursion above,

$$\begin{aligned}\vec{x}_{m+n} - \vec{x}^* &= (I - 2\eta A^T A)[\vec{x}_{m+(n-1)} - \vec{x}^*] \\ &\vdots \\ &= (I - 2\eta A^T A)^n[\vec{x}_{m+(n-n)} - \vec{x}^*] = (I - 2\eta A^T A)^n[\vec{x}_m - \vec{x}^*]\end{aligned}$$

Upon the fact that $I - 2\eta A^T A$ is a symmetric matrix (i.e., it is diagonalizable), we are granted the following statements:

$$\begin{aligned}I - 2\eta A^T A &= U\Lambda U^T \\ \vec{x}_k - (A^T A)^{-1}A^T\vec{b} &= (I - 2\eta A^T A)^k[\vec{x}_0 - (A^T A)^{-1}A^T\vec{b}] \\ &= U\Lambda^k U^T[\vec{x}_0 - (A^T A)^{-1}A^T\vec{b}]\end{aligned}$$

Hinting at what a good choice of η is such that we may guarantee the subsequent guesses decrease as timestep increase. Particularly, we choose some η such that the eigenvalues of $(I - 2\eta A^T A)$ are less than 1 in magnitude. This allows us to upperbound the scaling effect of a matrix on some vector to not increase the vector's size (in the sense that, our guess does not travel away from the optimum we attempt to find via gradient descent).

12.2.3 Generalizations

For functions that are μ-strongly convex, their gradients do not change too slowly.
For functions that are L-smooth, such that

$$f(\vec{y}) \leq f(\vec{x}) + \nabla f(\vec{x}) \cdot (\vec{y} - \vec{x}) + \frac{L}{2}\|\vec{y} - \vec{x}\|_2^2$$

their gradients do not change too fast.

Theorem 12.2.1. Lemma on Gradients of L-smooth Functions

Lemma: For an L-smooth function f

$$\|\nabla f(\vec{x})\|_2^2 \leq 2L(f(\vec{x}) - f(\vec{x}^*))$$

Using the L-smooth property of a function,

$$f(\vec{y}) \leq f(\vec{x}) + \nabla f(\vec{x}) \cdot (\vec{y} - \vec{x}) + \frac{L}{2}\|\vec{y} - \vec{x}\|_2^2$$

$$f(\vec{x} - \eta\nabla f(\vec{x})) \leq f(\vec{x}) + \frac{\mathrm{d}f}{\mathrm{d}x}(-\eta\nabla f(\vec{x})) + \frac{L}{2}\|\eta\nabla f(\vec{x})\|_2^2$$

Chapter 13

Descent Methods and Convex Optimizations

13.1 Continued, Gradient Descent Convergence Proof

While there was a splendid convergence rate for the least squares problem, we want to discuss how gradient descent operates for other functions.
To have some similar quadratic form for the gradient descent convergence proof, we may work with smooth and strongly convex function, and it would offer the interval at which a gradient descent convergence lies at. In lecture, this was phrased as a "quadratic sandwich".

> *Puts two bread across the two side of a strongly convex, smooth function
> **WHAT ARE YOU**
> *A quadratic sandwich, chef*

sorry for the digression.

13.1.1 Breads of Quadratic Sandwich

Here, let us review two definitions introduced in the prior lecture:

Definition 13.1.1. Review: Strong Convexity

If f is differentiable, then for some $\mu > 0$, it is μ-strongly convex if domain of f is convex and

$$\forall \vec{x}, \vec{y} \in domain(f), f(\vec{y}) \geq f(\vec{x}) + \frac{\mathrm{d}f}{\mathrm{d}x} \cdot (\vec{y} - \vec{x}) + \frac{\mu}{2}\|\vec{y} - \vec{x}\|_2^2$$

Definition 13.1.2. L-smooth Functions

For an L-smooth function,

$$f(\vec{y}) \leq f(\vec{x}) + \nabla f(\vec{x}) \cdot (\vec{y} - \vec{x}) + \frac{L}{2}\|\vec{y} - \vec{x}\|_2^2$$
$$f(\vec{x} - \eta \nabla f(\vec{x})) \leq f(\vec{x}) + \frac{\mathrm{d}f}{\mathrm{d}x}(-\eta \nabla f(\vec{x})) + \frac{L}{2}\|\eta \nabla f(\vec{x})\|_2^2$$

The L-smoothness of a function guarantees the gradient doesn't change too fast, while the μ-strong convexity guarantees the gradient doesn't change too slow.

Let us prove the lemma addressed in prior lecture regarding the L-smooth functions:

Theorem 13.1.1. Lemma on Gradients of L-smooth Functions

Lemma: For an L-smooth function f

$$\|\nabla f(\vec{x})\|_2^2 \leq 2L(f(\vec{x}) - f(\vec{x}^*))$$

Proof: Suppose the current guess is some vector $\vec{x}$, such that the next guess is $\vec{y} = \vec{x} - \eta\nabla f(\vec{x})$.
Provided that $f(\vec{x}^*)$ is a global minimum, we may determine that,

$$\forall \vec{x}, f(\vec{x}^*) \leq f(\vec{y}) = f(\vec{x} - \eta\nabla f(\vec{x}))$$

Meanwhile, we may find via the L-smoothness condition that,

$$\begin{aligned} f(\vec{x} - \eta\nabla f(\vec{x})) &\leq f(\vec{x}) + \frac{\mathrm{d}f}{\mathrm{d}x}(-\eta\nabla f(\vec{x})) + \frac{L}{2}\|\eta\nabla f(\vec{x})\|_2^2 \\ &= f(\vec{x}) - \eta\|\nabla f(\vec{x})\|_2^2 + \frac{\eta^2 L}{2}\|\nabla f(\vec{x})\|_2^2 \\ &= f(\vec{x}) + \eta\left(\frac{\eta L}{2} - 1\right)\|\nabla f(\vec{x})\|_2^2 \end{aligned}$$

To produce the most efficient upperbound, we should minimize the coefficient of gradient's squared L2 norm. This brings us to a convex optimization problem:

$$\min_{\eta} \frac{\eta^2 L}{2} - \eta$$

which we may solve via calculus to see,

$$\eta^* = \frac{1}{L}$$

Let us substitute this back into the above derivation,

$$\begin{aligned} f(\vec{x} - \eta\nabla f(\vec{x})) &= f(\vec{x}) + \eta\left(\frac{\eta L}{2} - 1\right)\|\nabla f(\vec{x})\|_2^2 \\ &= f(\vec{x}) - \frac{L}{2}\|\nabla f(\vec{x})\|_2^2 \end{aligned}$$

Once again, via the property of $f(\vec{x}^*)$ to be the global minimum,

$$\begin{aligned} f(\vec{x}^*) &\leq f(\vec{x} - \eta\nabla f(\vec{x})) \\ &= f(\vec{x}) - \frac{L}{2}\|\nabla f(\vec{x})\|_2^2 \end{aligned}$$

This can then be derived into the aforementioned lemma.

13.1.2 Quadratic Sandwich

Let us now demonstrate the "quadratic sandwich" we discussed in prior:

Theorem 13.1.2. Quadratic Sandwich

Theorem. For function f that is L-smooth and μ-strongly convex, then for some appropriate α,

$$\|\vec{x}_{t+1} - \vec{x}^*\|_2^2 \leq \alpha\|\vec{x}_t - \vec{x}^*\|_2^2$$

Proof. For a μ-strongly convex function, we may see that,

$$f(\vec{x}') \leq f(\vec{x}) + \frac{\mathrm{d}f}{\mathrm{d}x} \cdot (\vec{x}' - \vec{x}) + \frac{\mu}{2}\|\nabla f(\vec{x})\|_2^2$$

Be a little bit manipulative, and we will obtain:

$$\frac{\mathrm{d}f}{\mathrm{d}x} \cdot (\vec{x}' - \vec{x}) \leq f(\vec{x}') - f(\vec{x}) - \frac{\mu}{2}\|\nabla f(\vec{x})\|_2^2$$

Now, to find a relationship (such that we may continue a proof), we will have to dedicate the following algebraic effort.
Particularly, we attempt to find a relationship between different guesses that are one timestamp across:

$$\begin{aligned}
\|\vec{x}_{t+1} - \vec{x}^*\|_2^2 &= \|\vec{x}_t - \eta\nabla f(\vec{x}_t) - \vec{x}^*\|_2^2 \\
&= \|(\vec{x}_t - \vec{x}^*) - \eta\nabla f(\vec{x}_t)\|_2^2 \\
&= \big((\vec{x}_t - \vec{x}^*) - \eta\nabla f(\vec{x}_t)\big)^T\big((\vec{x}_t - \vec{x}^*) - \eta\nabla f(\vec{x}_t)\big) \\
&= \|\vec{x}_t - \vec{x}^*\|_2^2 + \|\eta\nabla f(\vec{x}_t)\|_2^2 + 2(\eta\nabla f(\vec{x}_t))^T(\vec{x}^* - \vec{x}_t) \\
&\leq \|\vec{x}_t - \vec{x}^*\|_2^2 + 2L\eta^2(f(\vec{x}_t) - f(\vec{x}^*)) + 2\eta(f(\vec{x}^*) - f(\vec{x}_t) - \frac{\mu}{2}\|\nabla f(\vec{x})\|_2^2) \\
&= (1 - \eta\mu)\|\vec{x}_t - \vec{x}^*\|_2^2 + (2L\eta^2 - 2\eta)(f(\vec{x}_t) - f(\vec{x}^*))
\end{aligned}$$

Here, by letting $\eta = \frac{1}{L}$, we can cancel out the latter term.
We thus obtain that the appropriate α is:

$$\alpha = 1 - \eta\mu = 1 - \frac{\mu}{L}$$

Therefore, our interval of convergence for L is:

$$\begin{aligned}
-1 < 1 - \frac{\mu}{L} < 1 \\
0 < \mu < 2L
\end{aligned}$$

The implication of quadratic sandwich is that it bounds any function's convergence in gradient descent via some working quadratic forms.
If boundable by quadratics, convergence can nicely occur.

13.2 Stochastic Gradient Descent

There is a variety of other names, but SGD is perhaps the most popular choice.
Essentially, SGD is "lazy gradient descent". This name comes from its algorithmic property of reducing computational cost to avoid computing an entire gradient for some large vector or dataset.

The loss function of a dataset is very frequently written in the form:

$$L(\vec{x}) = \frac{1}{m}\sum_{i=1}^{m} L_i(\vec{x})$$

For example, in least squares algorithm,

$$L(\vec{x}) = \frac{1}{m}\|A\vec{x} - \vec{b}\|_2^2 = \frac{1}{m}\sum_{i=1}^{m}(\vec{a_i}^T\vec{x} - \vec{b_i})^2$$

Fortunately, by the additive property of derivatives, we may also state that:

$$\nabla f(\vec{x}) = \frac{1}{m} \sum_{i=1}^{m} \nabla f_i(\vec{x})$$

In SGD, instead of stepping in the direction of $\nabla f(\vec{x})$, we take a step in $\nabla f_i(\vec{x})$.
This is a legitimate approach, because

$$\mathbb{E}[\nabla f_i(\vec{x})] = \frac{1}{m} \sum_{i=1}^{m} \nabla f_i(\vec{x}) = \nabla f(\vec{x})$$

The features of SGD involve:

- Computationally efficient!
- Can be comfortably applied on online data
- Is a more noisy algorithm, and will thus help to escape local minimum

Chapter 14

Applications and Extensions of Gradient Descent

14.1 Stochastic Gradient Descent, Continued

From last lecture, we have obtained the guess rule of gradient descent, and produced another guess rule named "Stochastic Gradient Descent" that is a comparatively effortless guess version of gradient descent.
For objective functions that can be decomposed into some components:

$$f(\vec{x}) = \frac{1}{m}\sum_{i=1}^{m} f_i(\vec{x})$$

the SGD guess rule is that

$$\vec{x}_{k+1} = \vec{x}_k - \eta \nabla f_i(\vec{x}_k)$$

where we may, via prior derivation, see that:

$$\mathbb{E}[\nabla f_i(\vec{x}_k) = \nabla f(\vec{x}_k)]$$

SGD is a rather noisy algorithm that doesn't use the true gradient.
Consequently, SGD may not converge with a constant stepsize, and it also loses the property of gradient descent where: upon convergence, the increment between guesses becomes zero due to the gradient at optimum points being $\nabla f(\vec{x}^*) = \vec{0}$. For SGD, then, there needs to have a decaying stepsize over time to portray some similar property of convergence.

14.1.1 A Demonstration of SGD

Let us consider an example: Let us discuss the objective function:

$$f(\vec{x}) = \frac{1}{m}\sum_{i=1}^{m} \frac{1}{2}\|\vec{x} - \vec{p}_i\|_2^2$$

which provides the optimum:

$$\vec{x}^* = \frac{1}{m}\sum_{i=1}^{n} \vec{p}_i$$

Suppose we initiate our guess at $\vec{0}$, and provide $\eta = \frac{1}{k}$.
SGD may either randomly choose one component of the objective function to take guess rule along, or do so circularly

(which is easier to implement). The circular approach comes with occasional benefits and elegance.
Let us discuss its iterations below.

$$\vec{x_1} = \vec{x_0} - \eta_1 \nabla f_1(\vec{x_0}) = -(\vec{x_0} - \vec{p_1}) = \vec{p_1}$$
$$\vec{x_2} = \vec{x_1} - \eta_2 \nabla f_2(\vec{x_1}) = \vec{p_1} - \frac{1}{2}(\vec{x_1} - \vec{p_2}) = \frac{1}{2}(\vec{p_1} + \vec{p_2})$$
$$\vec{x_3} = \vec{x_2} - \eta_3 \nabla f_3(\vec{x_2}) = \frac{1}{2}(\vec{p_1} + \vec{p_2}) - \frac{1}{3}(\vec{x_2} - \vec{p_3}) = \frac{1}{3}(\vec{p_1} + \vec{p_2} + \vec{p_3})$$

where we may see the guess is the mean of all points $\vec{p_i}$ involved until the current guess.

Choosing the stepsize decay pattern is a dark art of itself. Its not a story the Jedi would tell you. Its a Sith legend of Hyperparameter Tuning. The dark side of the Force is a pathway to many abilities some consider to be unnatural, but convex-optimal.

14.1.2 Mathematical Case Study on Convergence of SGD

Let us perform a case study for the least squares problem applied with SGD. We will use a variant of MSE for the loss function to optimize on:

$$f(x) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2}(a_i x - b_i)^2$$

where,

$$\begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} x = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}, \vec{x^*} = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|_2^2}$$

We may observe:

$$\nabla f_i(x) = a_i(a_i x - b_i), \vec{x_i^*} = \frac{b_i}{a_i}$$

Then, we would want to show that,

$$\min_i \vec{x_i^*} \leq \vec{x^*} \leq \max_i \vec{x_i^*}$$

Let us perform some algebraic manipulation as follows:

$$\begin{aligned} \vec{x^*} &= \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|_2^2} = \frac{\sum_{i=1}^{m} a_i^2 \frac{b_i}{a_i}}{\|\vec{a}\|_2^2} \\ &\leq \frac{\max_i \vec{x_i^*} \sum_{i=1}^{m} a_i^2}{\|\vec{a}\|_2^2} = \max_i \vec{x_i^*} \end{aligned}$$

observing that such logic can be applied onto the case of $\min_i \vec{x_i^*}$ as well.

The implication of such phenomenon aids us towards convergence. Provided that our gradient of least squares loss function is:

$$\nabla f_i(x) = a_i^2(x - \frac{b_i}{a_i})$$

If we have $x > \frac{b_i}{a_i}$, which means it would be larger than the maximum possible optimum $\max_i \vec{x_i^*}$, and we would be pushed towards the interval marked by $[\min_i \vec{x_i^*}, \max_i \vec{x_i^*}]$.
Furthermore, a similar logic applies to the case of $x < \frac{b_i}{a_i}$.
Our oracle (gradient) is generally correct (albeit occasionally inelegant). Now, we only require the step size to be some appropriate amount such that, provided the correct direction towards which the next guess should exist, we do not overshoot outside the aforementioned proper interval of optimization.

Keep in mind that such interval is still highly theoretical and will not be realistically attained, so the tuning of step size highly depends on whether the phenomenon of convergence is truly observed (via some plotting measure).

14.2 Gradient Descent with Prior Optimization Constraint

Suppose that we attempt to solve the following problem:

$$\min_{\vec{x} \in \mathcal{X}} f(\vec{x})$$

where, $\mathcal{X}$ is a convex, compact set upon which we perform gradient descent on.
However, using conventional methods of the general, unconstrained gradient descent would possibly put the convergence outside $\mathcal{X}$. The concept of solution is **Projected Gradient Descent**, where we define the projection of one point $\vec{y}$ onto another point $\vec{x}$ as:

$$\Pi_{\mathcal{X}}(\vec{y}) = argmax_{\vec{x} \in \mathcal{X}} \|\vec{y} - \vec{x}\|_2$$

Therefore, as we obtain a guess that is outside $\mathcal{X}$, we project that guess back into $\mathcal{X}$ to satisfy the constraint.
In other words, upon the usual guess rule computation, we would also add the computational cost of projections onto each iterative step of the original gradient descent algorithm. The guess rule of PGD is therefore:

$$\vec{x}_{k+1} = \Pi_{\mathcal{X}}(\vec{x}_k - \eta f(\nabla f(\vec{x}_k)))$$

Such problem is rarely used in practice, however, because we would be solving another optimization problem per step. The effect of such operation is inefficiency. Hence, although PGD works conceptually, it is not employed practically.

14.2.1 Conditional Gradient Descent

We may also propose an alternative variation from Frank Wolfe, where suppose we are solving for:

$$\vec{y_k} \in argmin_{\vec{y} \in \mathcal{X}} \nabla f(\vec{x}_k) \cdot \vec{y}$$

and γ is a fixed decating stepsize sequence.
Then the guess rule of CGD would be phrased as:

$$\begin{aligned} \vec{x}_{k+1} &= (1 - \gamma_k)\vec{x}_k + \gamma_k \vec{y}_k \\ &= \vec{x}_k + \gamma_k(\vec{y}_k - \vec{x}_k) \end{aligned}$$

This setup allows the iterated guess to be a convex combination within $\mathcal{X}$.

Chapter 15

Interlude: Logistic Regression

insert something about taking 127 and 189 together here

15.1 Monotone Transformations

In prior work, we have minimized a positive function by minimizing its squared value.
For example:

$$argmin_{\vec{x}}\|A\vec{x}-\vec{b}\|_2 = argmin_{\vec{x}}\|A\vec{x}-\vec{b}\|_2^2$$

This is an example of monotone transformation:

Definition 15.1.1. Monotone Transformation

For a function $\Phi(x)$ that is continuous and strictly increasing, then

$$p^* = \min_{\vec{x}} f_0(\vec{x}) \text{ s.t. } \forall i \in [1,n] f_i(\vec{x}) \leq 0$$

Then, such problem is equivalent to the following:

$$g^* = \min_{\vec{x}} \Phi(f_0(\vec{x})) \text{ s.t. } \forall i \in [1,n] f_i(\vec{x}) \leq 0$$

There are many continuous and strictly increasing functions, including but not excluded to:

- $\Phi(x) = x^2$, with domain restructed to non-negative x
- $\Phi(x) = \log(x)$
- $\Phi(x) = e^x$

15.1.1 Building Logistic Regression

Suppose we have datapoints $\{(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\}$, such that:

$$\forall i \in [1,n], y_i \in \{-1, 1\}$$

The task of logistic regression is to build a classifier between class labels 0 and 1 via predicting the probability that some datapoint belongs to a certain class:

$$\mathbb{P}(y_i = 1 | x = \vec{x}) = q(\vec{x})$$

Now, suppose that we construct such function as an affine transformation:

$$q(\vec{x}) = \vec{w}^T\vec{x} + \beta$$

However, the probability of an event must be between 0 and 1 (inclusively), so we must apply some other transformation such that $\vec{w}^T\vec{x}$ is no longer unbounded as it is now. Here, q is some other function we will involve in our estimation for probability, where maximizing q should grant a maximization of $\vec{w}^T\vec{x}$ as well.

Concretely, performing the monotone transformation $\Phi(x) = \log(x)$ onto $\vec{w}^T\vec{b}$ would make its range just unbounded in one single direction: $(\log(0), \log(1)) = (-\infty, 0)$.
And similarly, performing the monotone transformation $\Phi(x) = \frac{x}{1-x}$ makes $\vec{w}^T\vec{b}$ unbounded in anotehr direction: $(0, \infty)$.
Combining the above result, let us perform the monotone transformation:

$$\log\left(\frac{q(\vec{x})}{1 - q(\vec{x})}\right) = \vec{w}^T\vec{x} + \beta$$

such that $\vec{w}^T\vec{x}$ is indeed a suitable function for predicting probability.

From above, we may perform some algebraic manipulation, and would eventually find that:

$$q(\vec{x}) = \frac{\exp\big(\vec{w}^T\vec{x} + \beta\big)}{1 + \exp\big(\vec{w}^T\vec{x} + \beta\big)}$$

To maximize the probability at which we obtain our current dataset, we also have the option of performing MLE.
Suppose we apply MLE (Maximum Likelihood Estimation) on our above hypothesis for probability function, then we would be solving the optimization problem of maximizing the following expression:

$$\mathbb{P}(y_1, \ldots, y_n | \vec{w}^T, \beta)$$

Furthermore, suppose the probability at which $\mathbb{P}(y_i = 1)$ is as noted above, then we may dictate that:

$$\mathbb{P}(y_i = -1 | x = \vec{x}_i) = \frac{1}{1 + \exp\big(\vec{w}^T\vec{x}_i + \beta\big)} = \frac{\exp\big(-(\vec{w}^T\vec{x}_i + \beta)\big)}{1 + \exp\big(-(\vec{w}^T\vec{x}_i + \beta)\big)}$$

Then, substituting such expression:

$$\mathbb{P}(Y = y_i) = \frac{\exp\big(y_i(\vec{w}^T\vec{x}_i + \beta)\big)}{1 + \exp\big(y_i(\vec{w}^T\vec{x}_i + \beta)\big)}$$

such that the expressions of $\mathbb{P}(Y = y_i)$ is coherent with our algebraic works above:

$$\mathbb{P}(Y = y_i) = \begin{cases} \frac{\exp\big((\vec{w}^T\vec{x}_i + \beta)\big)}{1 + \exp\big((\vec{w}^T\vec{x}_i + \beta)\big)}, & y_i = 1 \\ \frac{\exp\big(-(\vec{w}^T\vec{x}_i + \beta)\big)}{1 + \exp\big(-(\vec{w}^T\vec{x}_i + \beta)\big)}, & y_i = -1 \end{cases}$$

Finally, applying maximum likelihood, we solve:

$$(\vec{w}^*, \beta^*) = argmax_{\vec{w},\beta} \prod_{i=1}^{n} \frac{\exp\big(y_i(\vec{w}^T\vec{x}_i + \beta)\big)}{1 + \exp\big(y_i(\vec{w}^T\vec{x}_i + \beta)\big)}$$

to build the parameters of logistic regression classifier.
Now, performing a monotone transformation with $\Phi(x) = \log(x)$ will grant us a convex objective function to optimize for:

$$(\vec{w}^*, \beta^*) = argmax_{\vec{w},\beta} \log\left(\prod_{i=1}^{n} \frac{\exp\big(y_i(\vec{w}^T\vec{x}_i + \beta)\big)}{1 + \exp\big(y_i(\vec{w}^T\vec{x}_i + \beta)\big)}\right)$$

Namely, while monotone transformations may change the convexity of a function, it will still provide an equivalent optimization problem.

Thanks for staying through all the maths.