EECS 127
Lecture Notes

Soklynin Nou
Fall 2025

Table of Contents

Introduction

Lecture 1

8/28/25

1. Background

For multiplication of matrix $A = m \times n$ and $B = n \times m$, we can rewrite it as:

$$AB = \sum_{i=0}^{m} \sum_{j=0}^{n} a_{ij} b_{ji} \tag{1}$$

2. Structure of Optimization

In general, optimization problems is a problem where we are maximizing or minimizing a function f_0 bounded by given constraints f_i:

$$\min_{x \in \mathbb{R}^n} f(x) \text{ such that } f_i(x) \leq 0 \tag{2}$$

The problem above is in **Canonical Form**, which is the standard form of solving these problems. Some ways to convert to canonical form is:

$$f_i(x) = 0 \implies f_i(x) \leq 0, -f_i(x) \leq 0 \tag{3}$$

There are two types of optimization problems:

- **Tractable Problems:** Can be solved in polynomial time
- **Non-Tractable Problems:** Is solved in exponential time

2.1. Feasibility

The **Feasible Solution** is a set of solutions that satisfies all constraints. We can think about it as the intersection of all the domains of each constraint $f_i(x)$:

Example:

$$\begin{gathered} \min x^2 \\ \text{s.t.} \\ -1 \leq x \leq 1 \end{gathered} \tag{4}$$

In this example, there is one constraint highlighting the feasible solution $-1 \leq x \leq 1$. Solving this, we can see that the minimum value of the function is at $x = 0$, which outputs a value of 0.

Types of Optimization Problems

Lecture 2

9/2/25

3. Cases of Solutions

When does the optimization solution have no function?

- When the solution is not a number such as $\pm\infty$.

This leads to the idea of feasible and infeasible solutions. A solution is infeasible there is no x that fits into the constrain(s). Sometimes the objective value could be $+\infty$ (infeasible), $-\infty$, or finite.

4. Types of Problems

We say that the optimization problem is tractable or not tractable.

- **Tractable:** There is an algorithm to solve the problem efficiently.
- **Not-Tractable:** There is no algorithm to solve the problem efficiently.

Example

Maximize the company's profit where we need to make n decisions.
What we do is we map each decision n to $+1$ if it is a yes decision and -1 if it is a no decision.
Optimization: max $f_0(x)$ or min $-f_0(x)$
Such that: $x_i = \{-1, +1\}$
Since there is 2 choices for each decision, this makes the total sample space 2^n.
This means that the problem is intractable since it must be solved in exponential time.
On the other hand, if the problem is solvable in polynomial time, the problem is tractable.

5. Linear Algebra Review

One notion we see repeatedly is the notion of *Space*. A *Space* is a collection os a certain type
Example: We say R^n is a collection of vectors with n elements.
In this class we will be focusing on the R^n space

5.1. Subspace

We also have the notion of *Subspace*. A non-empty set V is called a subspace of R^n if for every x,y in V, $ax + by$ is also in V. Additionally, the origin vector, a vector containing all zeroes, must also be in the subspace.

5.2. Span

The *Span* of vector m = $v_1, v_2, v_3, ...$ is the set of vectors forms from the linear combination of all the components of m.

5.3. Basis

Consider a subspace V, a set of vectors $v_1, v_2, v_3, ...$ in v is the basis of V is if all the vectors are linearly independent from each other. V_i and V_j are linear independent if the only coefficients between v_i and v_j are zeroes. In other words, those vectors are orthogonal to each other.

Given a vector, the set of basis is not unique but they have the number of basis vectors is equal to the dimension of the vector

Example
$v_1 = [1, 1, 1], v_2 = [1, 2, 3], v_3 = [-1, 0, 1]$
We define V = Span(v_1, v_2, v_3)
We can see that: $-2x_1 + 1x_2 - 1x_3 = 0$
Because $\exists$ non-zero coefficients that results the zero vector, the vectors are **linearly dependent**.
If $b_1x_1 + b_2x_2 = 0$ and $b_1 = b_2 = 0$, the vectors are **linearly independent**.

5.4. Affine Set

S is an an *Affine set* iff for every pair of points $x, y \in S$, the infinite line containing x and y is a subset of S. More formally, S is an *Affine set* if every affine combination of its points $x_1, ..., x_n \in S$. An affine combination takes the form of:

$$\alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_n x_n \in S \tag{5}$$

such that:

$$\alpha_1 + \alpha_2 + ... + \alpha_n = 1 \tag{6}$$

Example
For a vector $v \in V$ subspace, if adding x_0 to v results in a shifted version of the vector then S is an *Affine Set*. In other words

5.5. Norm of Vectors

We call $\| \cdot \|$ in R^n a *Norm* if:

- $\|x\| \geq 0, \forall x \in R^n \wedge \|x\| = 0 \leftrightarrow x = 0$
- $\|x + y\| \leq \|x\| + \|y\|, \forall x, y \in R^n$
- $\|ax\| = a \cdot \|x\|, \forall a \in R \wedge x \in R^n$

Example

- L_1 Norm: $\|x\|_1 = |x_1| + ... + |x_n|$
- L_2 Norm: $\|x\|_2 = \sqrt{x_1^2 + ... + x_n^2}$
- L_∞ Norm: $\|x\|_\infty \max\{|x_1|, ..., |x_n|\}$
- L_p Norm: $1 \leq p < \infty, \|x\|_p = (|x_1| + |x_2|)$, is a norm if p >= 1
- L_0: $\lim_{p \to 0} \left(|x_1|^p + ... + |x_n|^p\right)^{\frac{1}{p}}$ = $|x|$ = number of non-zero elements of x, not a norm

5.6. Inner Product

The inner product on a space X can be used to find the relationship between two vectors/data. For example, when analyzing two articles, we can turn them into n-dimensional vectors. We could then find the inner product of the two articles to see similarities with larger inner products indicating more similarities. They is because we are taking the cosine of the angle between the vectors.

5.6.1. Properties of Inner Product

$$\begin{aligned}
< x, y > &= x^T y = x_1 y_1 + x_2 y_2 + ... + x_n y_n \\
< x, x > &\geq 0, \forall x, y \in X \\
< x, x > &= 0 \longleftrightarrow x = 0 \\
< x, y, z > &= < x, y > = < y, z > \forall x, y, z \in X \\
< ax, y > &= a < x, y >, \forall x, y \in X \wedge a \in R \\
< x, y > &= < y, x >, \forall x, y \in X \\
< x, y > &= 0 \longleftrightarrow x \perp y
\end{aligned} \tag{7}$$

This means that the vectors are mutually orthogonal/linearly independent. We call a set of vectors **orthonormal** if:
$< x, y >= 0$ if $x \neq y$ and $< x, y >= 1$ if $x = y$
if $x \perp y$, then $\|x + y\|^2 = \|x\|^2 + \|y\|^2$

Subspaces and Dimensions

Lecture 3

9/4/25

6. Projection on Subspaces

Projection can be used to reduce the dimensions onto a subspace. Lets assume S is a subspace of some space X and given a point $x \in X$, the projection of x onto S is:

$$\Pi_s(x) = \text{argmin}_{y \in S} \|y - x\| = \min_{y \in S} \|y - x\| \tag{8}$$

The solution to this optimization is denoted as x^*.
Theorem: $x^* = \Pi_s^x$ exists and is unquie. This means that $x^* = \Pi_s^x \iff (x - x^*) \perp S$.

6.1. Proof of Projection Minimization

Let x^* be the project of x onto a subspace S and y be another point on $S, x \neq y$.

$$\|x - y\|^2 = \|x - x^* + x^* - y\|^2 \tag{9}$$

Note that $x - x^* \perp x^* - y$ by definition.

$$\|x - x^* + x^* - y\|^2 = \|x - x^*\|^2 + \|x^* - y\|^2 \tag{10}$$

We know that $\|x^* - y\|^2$ is a not zero value, which means that:

$$\|x - x^*\|^2 \leq \|x - x^*\|^2 + \|x^* - y\|^2 = \|x - y\|^2 \tag{11}$$

$\therefore x^*$ is the minimized projection onto S.

Given affine set A is a subspace and $y - x^* \in S,$ where S is a subspace. If $(x - x^*) \perp S,$ then $(x - x^*) \perp A$

6.2. Projection onto n-Dimensional Spaces

Given a 1-dimensional subspace S, let v be the basis for S and x be an arbitrary vector in $\mathbb{R}^n$. We know that $\Pi_S(x) = x^*$. Additionally, $x^* \in S \longrightarrow \exists \alpha^* : x^* = \alpha^* \cdot \vec{v}$.

$$\text{Let } y = \vec{v} \tag{12}$$

$$\begin{aligned} \Rightarrow &< x - \alpha^* \cdot \vec{v}, \vec{v} >= < x, \vec{v} > -\alpha^* \cdot \|\vec{v}\|^2 = 0 \\ &\Rightarrow \alpha^* = \frac{< x, \vec{v} >}{\|\vec{v}\|^2} \end{aligned} \tag{13}$$

This means:

$$x^* = \Pi_S(x) = \alpha^* \cdot v = \frac{< x, v >}{\|v\|^2} \cdot v \tag{14}$$

Example:
A data set of votes $n = 100$ senators on $m = 645$ bills in the period of 2004-2006. For $j = 1, ..., n$ define $x^i \in \mathbb{R}^m$ of votes of senator j.

Answer:
What we want to do is to project the data on to a 1-dimensional space, Which means we need to define a subspace and its basis vectors. Assume the senators vote "for" or "against" at the same rate, 50%.
Lets define $\hat{x}$ as the average vote of the senators. This means:

$$\hat{x} = \frac{1}{n}(x_1, ..., x_n) \tag{15}$$

To remove bias, we substract each vote by the average $\hat{x}$:

$$X = (x_1 - \hat{x}, ..., x_n - \hat{x}) \tag{16}$$

We now want to project this unbiased score onto the subspace:

$$\Pi_S x_j - \hat{x} = \frac{< x_j - \hat{x}, v >}{\|v\|^2} \cdot v \tag{17}$$

The main obstacle is to find the subspace which gives the most accurate data.

Gradients, Taylor Series, and Other Stuff

Lecture 4

9/9/25

6.2.1. Generalization of Projection

Let assume S is a d-dimension subspace with basis: $x^{(0)}, x^{(1)}, ..., x^{(d)}$
Let x^* be the project of x onto S
This means $x^* \in S$, which implies $x^* = \sum_{i=0}^{d} \alpha_i x_i$ ($\exists$ Linear Combination)
From this projection, we know

$$< x - x^*, y >= 0, \forall y \in S \tag{18}$$

Let y be a basis of S, $y = x^{(j)}$. y is now a linear combination of the basis vectors:

$$\begin{aligned} < x - \sum_{i=0}^{d} \alpha_i x_i, x^{(j)} >= 0 \\ < \sum_{i=0}^{d} \alpha_i x_i, x^{(j)} >=< x, x^{(j)} > \end{aligned} \tag{19}$$

Because $x^{(j)}$ can be written as a linear combinations of the basis of S:

$$\sum_{i=0}^{d} \alpha_i < x^{(i)}, x^{(j)} >=< x, x^{(j)} > \tag{20}$$

Expanding this will give d equations with d unknowns

$$\begin{aligned} \alpha_1 < x^{(1)}, x^{(1)} > + ... + \alpha_d < x^{(1)}, x^{(d)} >=< x, x^{(1)} > \\ ... \\ \alpha_1 < x^{(d)}, x^{(1)} > + ... + \alpha_d < x^{(d)}, x^{(d)} >=< x, x^{(j)} > \end{aligned} \tag{21}$$

If $\{x^{(1)}, ..., x^{(d)}\}$ are chosen to be orthonormal with each other, then the dot product of:

$$< x^{(i)}, x^{(j)} >= 0, i \neq j \tag{22}$$

6.3. Hyperplanes & Half-Spaces

6.3.1. Hyperplanes

A **Hyper Plane** is an n-dimensional affine set

$$H = \{z \in \mathbb{R}^n \mid a^T z = b, \text{where } a, b \in \mathbb{R}\} \tag{23}$$

Example:
In $\mathbb{R}^3$ a hyperplane is a 2-dimensional plane with basis $z^{(1)}, z^{(2)} \in H$
Using the definition, we can write this as $a^T z^{(1)} = b, a^T z^{(2)} = b \Rightarrow a^T \left(z^{(1)} - z^{(2)}\right) = 0$
This means that $a \perp \left(z^{(1)} - z^{(2)}\right)$, we call a the normal vector

6.3.2. Half-Spaces

A **Half-Space** is a space that occupy the space outside the plane. We define the two spaces as H_+ and H_-:

$$\begin{aligned} H_+ &= \{x \mid a^T x \geq b\} \\ H_- &= \{x \mid a^T x \leq b\} \end{aligned} \tag{24}$$

This means that H_+ occupies the space that the normal vector points towards while H_- occupies the space that the normal vector points away from.

6.4. Gradients

A gradient is defined as the partial derivative of the each dimension with the corresponding variable:

$$\nabla f(x) = \begin{bmatrix} \frac{\delta f(x_1)}{\delta x_1} \\ \frac{\delta f(x_2)}{\delta x_2} \\ \frac{\delta f(x_3)}{\delta x_3} \end{bmatrix} \tag{25}$$

Two important notions in convex optimization:

6.4.1. Linear functions

A **Linear Function** takes the form of $f : \mathbb{R}^n \longrightarrow \mathbb{R}$

$$f(\alpha_1 x_1 + \alpha_2 x_2) = \alpha_1 f(x_1) + \alpha_2 f(x_2) \tag{26}$$

6.4.2. Affine Function

An **Affine Function** takes the form of $f : \mathbb{R}^n \longrightarrow \mathbb{R},$ if $f(x) - f(0)$

$$\exists a \in \mathbb{R}^n \ \& \ b \in \mathbb{R} \mid f(x) = a^T x = b, \text{ where } b = f(0) \tag{27}$$

NOTE: If $f(x) = a^T x = b$, then $\nabla f(x) = a$

6.4.3. Chain Rule for Gradients

Consider a functions $\varphi(x) = f(g(x))$, where:

$$\begin{aligned} f &: \mathbb{R}^m \longrightarrow \mathbb{R} \\ g &: \mathbb{R}^n \longrightarrow \mathbb{R}^m \end{aligned} \tag{28}$$

We can use this composition to perform a change in coordinate.
Example:

$$\nabla f(g(x)) = \nabla f(z) \cdot \nabla g(x), \text{ where } z = g \tag{29}$$

6.5. Taylor Series

Given $f(x) : \mathbb{R}^n \longrightarrow \mathbb{R}$ and is differentiable at $x_0 \in \mathbb{R}^n$, we can approximate the function by an affine function in a neighborhood of x_0.

$$f(x) = f(x_0) + \nabla f(x_0)^T (x - x_0) + \varepsilon(x) \tag{30}$$

where $\varepsilon(x)$ is the error term.
Theorem: If x^* is a local minimum of $\min(f(x))$, where $f(x)$ is differentiable, then nabla $\nabla f(x^*) = 0$
NOTE: This the same concept as calculus optimization, where the minimum of $f(x)$ is at $f'(x) = 0$.

Ranges and Null Space

Lecture 5

9/11/25

7. More Subspaces

7.1. Range

The **Range** of a matrix can be though of as the solution space of a given equation

$$Ax = y \tag{31}$$

Determining the range of A requires us to find S so that:

$$S = \{x \in \mathbb{R}^n \mid Ax = y\} \tag{32}$$

The $Range(A)$ is the set of solution, whose dimension is called the **Rank** of A.

Theorem: Rank(A) = # of Linearly independent columns of A. This implies that $0 \leq rank(A) \leq \min(m, n)$. If $rank(A) = 0$, then $A = 0$.

7.2. Null Space

The **Null Space** of a matrix can be though of as the solution space of a given equation

$$Ax = 0 \tag{33}$$

Determining the range of A requires us to get the subspace S so that:

$$S = \{x \in \mathbb{R}^n \mid Ax = 0\} \tag{34}$$

Because it passes through the origin, by definition, the null space is a subspace.
To find the null space of a matrix, we setup a system of equations:

Example in R^3:

$$\begin{gathered} Ax = 0 \\ ax_1 + bx_2 + cx_3 = 0 \end{gathered} \tag{35}$$

We then write all x_i as in terms of others except for one:

$$\begin{aligned} x_1 &= -\frac{bx_2 + cx_3}{a} \\ x_2 &= -\frac{ax_1 + cx_3}{b} \end{aligned} \tag{36}$$

We now setup the vector form:

$$Nul(A) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -\frac{bx_2+cx_3}{a} \\ -\frac{ax_1+cx_3}{b} \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{a}{b} \\ 0 \end{bmatrix} x_1 + \begin{bmatrix} -\frac{b}{a} \\ 0 \\ 0 \end{bmatrix} x_2 + \begin{bmatrix} \frac{c}{a} \\ \frac{c}{b} \\ 1 \end{bmatrix} x_1 \tag{37}$$

Therefore:

$$Nul(A) = span\left(\begin{bmatrix} 0 \\ -\frac{a}{b} \\ 0 \end{bmatrix}, \begin{bmatrix} -\frac{b}{a} \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{c}{a} \\ \frac{c}{b} \\ 1 \end{bmatrix} \right) \tag{38}$$

7.3. Fundamental Theorem of Linear Algebra

The Range and Null Space can be related to each other using the Fundamental Theorem of Linear Algebra:

- $N(A) \perp R(A^T)$
- $N(A) \oplus R(A^T) = \mathbb{R}^n$
- $\forall h \in \mathbb{R}^n, \exists x \in N(A), \exists w \in R(A^T) : h = v + w$
- $\dim(N(A)) + rank(A) = n$

Example Proof:

$$\begin{gathered} v \in N(A) \implies Av = 0 \\ w \in R(A^T) \implies \exists u : A^T u = w \\ < u, w >= v^T w \implies v^T A^T u = (Av)^T u = 0 \end{gathered} \tag{39}$$

7.4. Applications of Range and Null Space

Studying the solution space of $Ax = y$, we must make sure that it is not empty. We can find this by checking if y is a linear combination of A.

S is non-empty iff:

- $y \in R(A)$
- $y \in span(col(A))$
- $rank(A) = rank([A + y])$

7.5. Solution Spaces

Sometimes find the solution could seem extremely difficult, which may be because there isn't one. Or maybe there is a separate valid solution. This means that we need to determine the **Solution Space** of $Ax = y$ to better understand the equation.

Let $\bar{x}$ be an arbitrary solution to the equation $Ax = y$, meaning $A\bar{x} = y$

$$\begin{aligned} Ax &= y \\ Ax - A\bar{x} &= y - A\bar{x} \\ A(x - \bar{x}) &= 0 \end{aligned} \tag{40}$$

let $z = x - \bar{x} \in Nul(A)$

This means that each solution x is the sum of $\bar{x}$ and a vector in the null space:

$$S = \bar{x} + Nul(A) \tag{41}$$

7.5.1. Tall Matrix

A tall matrix is a matrix where there are more rows then columns. This means that $Ax = y$ has m equations & n unknowns, where A is a $m \times n$ matrix and $n < m$. This is a overdetermined case, where:

- $\dim(N(A)) + Rank(A) = n$
- $N(A) = \{0\}$

This means that there is only one solution iff $y \in R(A)$

7.5.2. Wide Matrix

A wide matrix is a matrix where there are more columns then rows. This means that $Ax = y$ has m equations & n unknowns, where A is a $m \times n$ matrix and $n > m$. This is a overdetermined case, where:

- $\dim(N(A)) = n - + Rank(A) > 0$

This means that there are an infinite amount of solutions since for solution set:

$$S = \bar{x} + N(A) \tag{42}$$

We can add any arbitrary value in $N(A)$ to $\bar{x}$ to get a new distinct solution.

7.5.3. Optimal Solution

In the case where we have an infinite number of solutions, how do we know the more optimal solution? This can be found by minimizing the input vector as it could represent using less resources:

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} \|x\| \\ &Ax = y \end{aligned} \tag{43}$$

This can be achieved using projection x^* from the origin onto the solution space. Since the solution space is an affine set of the null space

$$S = \bar{x} + N(A) \tag{44}$$

the project of the origin onto the solution space is both perpendicular to the null space and solution space. Using this fact, we can write it in terms of an inner product:

$$< x^*, x^* - \bar{x} >= 0 \tag{45}$$

where $x^* - \bar{x} \in Nul(A)$, since the point $S - \bar{x} = Nul(A)$ and $x^* \in S$
Because $x^* \perp Nul(A)$ and using the **FToLA**, we formulate the following conclusion:

$$x^* \perp Nul(A) \ \wedge \ Nul(A) \perp R(A^T) \Longrightarrow x^* \perp R(A^T) \tag{46}$$

This then means:

$$\exists u \in R(A^T) : A^T u = x^* \tag{47}$$

We can now express x^* as such:

$$x^* = A^T u = A^T (AA^T)^{-1} y \tag{48}$$

Least Squares

Lecture 6

9/16/25

7.5.4. Square Matrix

If A is a full rank square matrix, then the solution to $Ax = y \implies x^* = A^{-1}y$

7.6. Least Squares

Consider an equation: $Ax = b$ and we want to find:

$$\min_{x \in \mathbb{R}^n} f(x) \tag{49}$$

Example:
In an MRI machine, there are beams that passes those boxes. The intensity of the beams is dependant on the angle and how many boxes we passes through. We can model this behavior by setting the number of beams equal to y and the box position as indices of a matrix x and the A matrix represents the intensity.

$$\begin{aligned} x_1 + x_2 &= 2 \\ 2x_1 + 2x_2 &= 5 \\ 4x_1 - x_2 &= 3 \end{aligned} \tag{50}$$

The solution will be $x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. But, if there is notice in the system, $y_i = a + \varepsilon$, there will never be a definite solution. This is where the least square comes in. We will try to minimize the difference between $Ax - y$.

$$\min(\|Ax = y\|) \tag{51}$$

where A is a $m \times n$ matrix and $y \in \mathbb{R}^m$

7.6.1. Linear Regression

In **Linear Regression**, we are trying to find the line $z_2 = az_1 + b$ that best find the data points given. Using the concepts from least squares, we can treat $a = x_1$ and $b = x_2$ indices of a matrix.
We can rearrange this to:

$$z_2^{(i)} - x_1 z_1^{(i)} + x_2 \tag{52}$$

Which is not necessarily equal to 0 since it can be an estimate.

We can then write it in least square form:

$$\begin{aligned} &\min_{x \in \mathbb{R}^2} \sum_{i=1}^{m} \left(x_1 z_1^{(i)} + x_2 - z_2^{(i)} \right) \\ &\min_{x \in \mathbb{R}^2} \sum_{i=1}^{m} \left(\begin{bmatrix} z_1^{(i)} & 1 \end{bmatrix} x - z_2^{(i)} \right) \end{aligned} \tag{53}$$

This is now a least squares problem, where:

$$\begin{gathered}\begin{bmatrix} z_1^{(i)} & 1 \end{bmatrix} = A \\ x = x \\ z_2^{(i)} = y \end{gathered} \tag{54}$$

In a general **Linear Regression** problem:

$$\text{Input } a \in \mathbb{R}^n \longrightarrow \boxed{\text{System}} \longrightarrow \text{Output } a^T x \in \mathbb{R} \tag{55}$$

and we want to formulate it in least squares to find the coefficient x.
We would map each $a^{(i)}$ to it corresponding $y^{(i)}$

7.6.2. Non-Linear Regression

Unlike linear regression's model of $z_2 = az_1 + b$, **Non-Linear Regression** allows us to work with a more accurate model of

$$z_2 = az_1^2 + bz_2 + c \tag{56}$$

This means we will be working with an $x \in R^3$ since our unknowns are a, b, c.
Therefore, we will have the following optimization.

$$\begin{gathered}\min_{x \in \mathbb{R}^2} \sum_{i=1}^{m} \left(x_1 \left(z^{(i)} \right)_1^2 + z^{(i)} x_2 + x_3 - z_2^{(i)} \right) \\ \therefore \min \|Ax - y\|_2^2 \end{gathered} \tag{57}$$

According to **Gaussian IIDs**, the 2-norm is the best estimate, also called the max likelihood.

7.7. Solving Least Squares

Assuming that the exact solution y is outside the range of A, we can find the closest estimation to that solution by projection u onto $range(A)$.
Let the projection of y onto $range(A)$ be y^*. This means that the vector $(y - y^*) \perp range(A)$. According to the **FToLA**, $Nul(A^T) \perp Range(A)$. This then implies that $(y - y^*) \in Nul(A^T)$, which then means $A(y - y^*) = 0$. We can then rearrange the equations to get:

$$\begin{gathered} A(y - y^*) = 0 \\ Ay = Ay^* \\ A^T y = A^T A x^* \\ x^* = \left(A^T A \right)^{-1} A^T y \end{gathered} \tag{58}$$

This is equivalent to $Ax^* = y \implies x^* = A^{-1} y$ since the transpose matrices cancel out.

7.8. Quadratic Equations

Another way to obtain this formula:

$$\min f(x) \longrightarrow \nabla f(x) = 0 \tag{59}$$

The issue is that $\|Ax-y\|$ is a norm function and $\|\cdot\|$ is not differentiable at the origin. But, we know that $\min\|Ax-y\|$ is equivalent to $\min\|Ax-y\|^2$, which is differentiable.

$$\begin{aligned}\min\|Ax-y\|^2 &= \langle Ax-y, Ax-y\rangle \\ &= (Ax-y)^T(Ax-y) \\ &= (x^TA^T-y^T)(Ax-y) \\ &= x^TA^TAx - x^TA^Ty - y^TAx + y^Ty \\ &= x^TA^TAx - \left(y^TAx\right)^T - y^TAx + y^Ty \\ &= x^TA^TAx + 2(A^Ty)x + y^Ty\end{aligned} \tag{60}$$

Let $p = A^TA,\ q = -2A^Ty,\ r = y^Ty$

$$\min\|Ax-y\|^2 = x^TPx + q^Tx + r \tag{61}$$

In the quadratic form, we always choose P to be symmetric, $P = P^T$. This will allow us to differentiate the equation:

$$\begin{aligned}f(x) &= x^TPx - q^Tx - r \\ &= \sum_{i=1}^n\sum_{j=1}^n p_{ij}x_ix_j + \sum_{i=1}^n q_ix_i + r \\ &= \sum_{i=1}^n p_{ii}x_i^2 + \sum_{i<j}^n (p_{ij}+p_{ji})x_ix_j + \sum_{i=1}^n q_ix_i + r \\ &= \sum_{i=1}^n p_{ii}x_i^2 + 2\sum_{i<j}^n p_{ij}x_ix_j + \sum_{i=1}^n q_ix_i + r \\ \frac{\partial f(x)}{\partial x_k} &= p_{kk}x_k + 2\sum_{k<j}^n p_{kj}x_j + 2\sum_{k<j}^n p_{ik}x_i + q_k \\ &= 2\sum_{i=1}^n p_{ki}x_i + q_k\end{aligned} \tag{62}$$

Therefore:

$$\nabla f(x) = 2Px + q \tag{63}$$

This is equivalent to the normal equation seen before:

$$\begin{aligned}&\Rightarrow \nabla f(x^*) = 2Px^* + q \\ &\Rightarrow 2A^TAx^* - 2A^Ty = 0 \\ &\Rightarrow A^TAx^* = 2A^Ty \\ &\Rightarrow x^* = \left(A^TA\right)^{-1}A^Ty\end{aligned} \tag{64}$$

Second Order Optimization Conditions

Lecture 7

9/18/25

8. Eigenvalues, Eigenvectors, and Their Applications

An **Eigenvalues λ** is a scalar that describes the "stretch" of an Eigenvector given a transformation. More formally, the relationship between the eigenvalues and the transformation matrix is as such:

$$Ax = \lambda x \tag{65}$$

Where x is the eigenvector
Using this fact, we can find the eigenvalues of any transformation matrix by:

$$\begin{aligned} Ax &= \lambda x \\ \lambda x - Ax &= 0 \\ x(\lambda I - A) &= 0 \\ \det(\lambda I - A) &= 0 \end{aligned} \tag{66}$$

Example: Given a matrices:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{67}$$

Find the Eigenvalue(s)

$$\begin{aligned} \det(\lambda I - A) &= (\lambda - 1)(\lambda - 1) \\ \therefore \lambda &= 1 \end{aligned} \tag{68}$$

Find the Eigenvector(s)

$$\begin{aligned} (1)I - A &= \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \\ \therefore \vec{\lambda} &= \begin{bmatrix} -1 \\ 0 \end{bmatrix} \end{aligned} \tag{69}$$

8.1. Diagonalizable Matrix

Given a matrix with eigenvectors, we can **Diagonalize** the matrix to the form of:

$$A = PDP^{-1} \tag{70}$$

where

$$P = [\vec{u_1}\ \vec{u_2}\ ...\ \vec{u_3}], \text{where } \vec{u_i} \text{ is an eigenvector}$$
$$D = \begin{bmatrix} \lambda_1 & 0 & ... & 0 \\ 0 & \lambda_2 & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & \lambda_n \end{bmatrix} \tag{71}$$

8.2. Orthogonal Matrix

Any given vector U is orthogonal if and only if:

$$U^T U = UU^T = I_n \tag{72}$$

and all its vectors are orthonormal.

8.3. Symmetrical Matrix

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

$$A = A^T \tag{73}$$

According to the **Spectral Theorem**, if A is a **Symmetrical Matrix**, then its eigenvectors are orthonormal.

8.3.1. Hessian Matrices

A **Hessian Matrix** is symmetric if for $f(x) : \mathbb{R}^n \longrightarrow \mathbb{R}$, then:

$$\begin{gathered} \nabla f(x) \in \mathbb{R}^n \\ \nabla^2 f(x) \in \mathbb{R}^{n \times n} \end{gathered} \tag{74}$$

Example:

$$\nabla f(x) = \begin{bmatrix} \frac{\delta f}{\delta x_1} \\ ... \\ \frac{\delta f}{\delta x_n} \end{bmatrix}$$
$$\nabla^2 f(x) = \begin{bmatrix} \frac{\delta f}{\delta x_1 \delta x_1} & \frac{\delta f}{\delta x_1 \delta x_2} & \cdots & \frac{\delta f}{\delta x_1 \delta x_n} \\ \frac{\delta f}{\delta x_2 \delta x_1} & \frac{\delta f}{\delta x_2 \delta x_2} & \cdots & \frac{\delta f}{\delta x_2 \delta x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta f}{\delta x_n \delta x_1} & \frac{\delta f}{\delta x_n \delta x_2} & \cdots & \frac{\delta f}{\delta x_n \delta x_n} \end{bmatrix} \tag{75}$$

This matrix is symmetrical since the (i,j) and (j,i) index of the matrix is the same based on the symmetry of second derivative.

8.3.2. Defining Symmetric Matrices

Based on the eigenvalues of the **Symmetric** matrices, we can group matrices into different classes:

- **Positive Semi-Definite:**

$$\begin{gathered} \lambda_1, ..., \lambda_n \geq 0 \\ A \in S^n \text{ is PSD if } x^T A x \geq 0, \forall x \in \mathbb{R}^n \end{gathered} \tag{76}$$

- **Positive Definite:**

$$\begin{gathered}\lambda_1, ..., \lambda_n > 0 \\ A \in S^n \text{ is PD if } x^T Ax > 0, \forall x \in \mathbb{R}^n\end{gathered} \tag{77}$$

- **Negative Semi-Definite:**

$$\begin{gathered}\lambda_1, ..., \lambda_n \leq 0 \\ A \in S^n \text{ is NSD if } x^T Ax \leq 0, \forall x \in \mathbb{R}^n\end{gathered} \tag{78}$$

- **Negative Definite:**

$$\begin{gathered}\lambda_1, ..., \lambda_n < 0 \\ A \in S^n \text{ is ND if } x^T Ax < 0, \forall x \in \mathbb{R}^n\end{gathered} \tag{79}$$

- **Sign Indefinite**

$$\lambda_1, ..., \lambda_n \in \mathbb{R} \tag{80}$$

8.4. Convexity and Local Minimums

Theorem:
Consider $\min_{x \in \mathbb{R}^n} f(x)$
Assume $\nabla f(x) \geq 0, \forall x \in \mathbb{R}^n$
All local min = global min when x^* is a local/global min iff $\nabla^2 f(x^*)$ is positive semi-definite.

This highlights that if the function has the shape of a bowl (Convex), then there is only one local min, which is also the global minimum.

Regression continued and Singular Values

Lecture 8

9/23/25

8.5. Mid-Value Theorem

The **Mid-Value Theorem** states that:

$$\exists v \in \mathbb{R}^n : f(x_0) < f(v) < f(x_1) \tag{81}$$

This means that there exist a point in between the two inputs that is outputs a value also in between the two inputs. We can then express the function $f(x)$ as such:

$$f(x) = f(x^*) + \nabla f(x^*)^T (x - x^*) + \frac{1}{2}(x - x^*)\nabla^2 f(v)(x - x^*) \tag{82}$$

We know that $\nabla f(x^*)^T = 0$ from our definition of convex functions and $\nabla^2 f(v) \geq 0$. This means that:

$$\begin{gathered} f(x) = f(x^*) + a, \text{where } a \geq 0 \\ f(x) \geq f(x^*) \\ \therefore x^* \text{ is the global minimum} \end{gathered} \tag{83}$$

9. Ridge Regression

In **Ridge Regression**, we find

$$x \in \mathbb{R}^n : \min \|Ax - y\|^2 + \alpha \|x\|^2 \tag{84}$$

Where α is a **Regularization Term** is a user defined variable that defines the relationship between the variable and its output.

9.1. Solutions of Ridge Regression

We have

$$f(x) = \|Ax - y\|^2 + \alpha \|x\|^2 = x^T (A^T A + \alpha I_n) x + \left(-A^T y\right)^T x + (y^T y) \tag{85}$$

We then find the gradient and hessian of this function:

$$\begin{aligned} \nabla f(x) &= 2(A^T A + \alpha I_n)x - 2A^T y \\ \nabla^2 f(x) &= 2(A^T A + \alpha I_n) \end{aligned} \tag{86}$$

We can then use this to find the solution set:

$$\begin{gathered} \nabla f(x^*) = 2(A^T A + \alpha I_n)x^* - 2A^T y = 0 \\ 2(A^T A + \alpha I_n)x^* = 2A^T y \end{gathered} \tag{87}$$

This gives us the solution set:

$$\mathcal{S} = \left\{ x^* \mid A^T A x^* = A^T y \right\} \tag{88}$$

Corollary:

1. $S = A^T y + N(A)$
2. $\min \|x\| : s \in S = A^T y$ is unique

9.2. Eigenvalue for Non-Square Matrices (Singular Values)

Remember that a matrix A is diagonalizable if and only if:

$$A = U^{-1} \Lambda U \tag{89}$$

But what do we do for when $m \neq n$
This is where **Singular Values** are used:

Singular Value Decomposition (SVD) for a given $A \in \mathbb{R}^{m \times n}$ is:

$$\exists U \in \mathbb{R}^{m \times m}, \forall V \in \mathbb{R}^{n \times n}, \Sigma \in \mathbb{R}^{m \times n} \tag{90}$$

Such that:

- $A = U \Sigma V^T$

- U, V are orthogonal matrices
- If $n \geq m, \Sigma$ does not occupy every column
- If $n \leq m, \Sigma$ does not occupy every row
- $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r > \sigma_{r+1} = ... = 0$

Every matrix $A \in \mathbb{R}^{m \times n}$ can be decomposed using SVD into the form of:

$$A = U\Sigma V^T \tag{91}$$

In the case where some or all the singular values are zero, this means that the rank of the matrix A is k, where k is the number of non-zero singular values. Intuitively, this means that dimensions $k+1$ to n map to the null space of A. This then means that the eigenvectors for the zero singular values is the null space of A. This process is a series of change of basis and scaling.

Example:

$$\begin{aligned} A &= U\Sigma V^T \\ Ax &= U\Sigma V^T x \\ &= U\Sigma \begin{bmatrix} v_n^T x \\ ... \\ v_n^T x \end{bmatrix} \text{ writing x in v-coordinate} \\ &= U \begin{bmatrix} \sigma v_n^T x \\ ... \\ \sigma v_n^T x \end{bmatrix} \text{ scaling each coordinate by } \sigma \\ &= \sum_{i=1}^{m} u_i(\sigma v^T x) \end{aligned} \tag{92}$$

Theorem:
For matrix A, $rank(A) = r$. Additionally, if A is Positive, Semi-Definite, then $\Sigma = \Lambda$ and $U = V$

9.3. Pseudo-Inverse Matrices

The Pseudo-Inverse, also called the **Moore-Penrose Inverse**, of matrix A is defined as:

$$A = V\Sigma^+ U^T \tag{93}$$

Where every entry of Σ^+ is $\frac{1}{\sigma_i}$
If A is a wide and full row rank matrix or tall and full row rank matrix, then $A^T = A^T(AA^T)^{-1}$

9.4. Finding Singular Values

Consider the matrix:

$$A = U\Sigma V^T \tag{94}$$

We can multiple it by its transpose:

$$\begin{aligned} AA^T &= U\Sigma V^T V \Sigma U^T \\ &= U\Sigma\Sigma^T U^T, \text{since } V^T V = I_n \end{aligned} \tag{95}$$

We know that $\Sigma\Sigma^T$ is also a diagonal matrix since in both case of $m \geq n$ and $m \leq n$, $\Sigma\Sigma^T$ will be a square matrix with the singular values $\sigma_1 \geq \sigma_2 \geq ...$ as the diagonal indices. This then means that AA^T is a diagonalizable matrix in the form of:

$$\begin{aligned} AA^T &= U\Sigma\Sigma^T U^T \\ A^TA &= V\Sigma\Sigma^T V^T \end{aligned} \tag{96}$$

This establish that $\sigma_i^2 = \lambda_i$.
Going back to **Pseudo-Inverse Matrices**, we now could conclude that for the formula:

$$A = U\Sigma V^T \tag{97}$$

We now know that:

- U is the columns of the **normalized** Eigenvectors of AA^T
- V is the columns of the **normalized** Eigenvectors of A^TA
- Σ is the non-zero singular values of A, where $\sigma_i = \sqrt{\lambda_i}$ of AA^T or A^TA

Low-Rank Optimization

Lecture 9

9/25/25

9.5. Applications of Singular Value Decomposition

Lets assume we have a matrix representing an image. This matrix consists of values representing pixel values [0, 255]. We can call $A \in \mathbb{R}^{n\times n}$, the first picture, and represent it as:

$$\begin{aligned} A = U\Sigma V^T &= \begin{bmatrix} U^{(1)} & U^{(2)} & ... & U^{(n)} \end{bmatrix} \Sigma \begin{bmatrix} V^{(1)} & V^{(2)} & ... & V^{(n)} \end{bmatrix} \\ &= \sigma_1 U^{(1)} V^{((1))^T} + ... + \sigma_n U^{(n)} V^{((n))^T} \end{aligned} \tag{98}$$

We can then approximate the matrix:

$$A \approx \sigma_1 U^{(1)} V^{((1))^T} \tag{99}$$

This will reduce the size to $1 + n + n = 2n + 1 < n^2$, where n^2 is the size of the original image.
This demonstrates that we can approximate a matrix $A \approx \sqrt{\sigma_1}U^{(1)}\sqrt{\sigma_1}V^{((1))^T}$ We can represent $\sqrt{\sigma_1}U^{(1)} = x$ and $y = \sqrt{\sigma_1}V^{((1))^T}$:

$$A \approx xy^T \tag{100}$$

Now our goal is to find x and y that best approximates A. This is then a **Least Squares** problem:

$$\min_{x,y\in\mathbb{R}} \left\| A - xy^T \right\| \tag{101}$$

9.5.1. Matrix Norms

The problem above requires taking the norm of a $m \times n$ matrix, which will output a scalar $\in \mathbb{R}$ with 3 properties:

1. $\|x\| \geq 0 \forall x \in \mathbb{R}^{m \times n}, \|x\| = 0 \longleftrightarrow x = 0$
2. $\|\alpha x\| = |\alpha| \|x\|, \forall \alpha \in \mathbb{R}, \forall x \in \mathbb{R}^{m \times n}$
3. $\|x + y\| \leq \|x\| + \|y\|, \forall x, y \in \mathbb{R}^{m \times n}$

Frobenius Norm:

$$\|A\|_F = \|Vec(A)\|_2 = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2} \qquad \text{where } a_{ij} \text{ is the } ij^{\text{th}} \text{ element of } A \tag{102}$$

LP-Induced Norm:

$$\|A\|_P = \max_{z \neq 0 \in \mathbb{R}^n} \frac{\|Az\|_P}{\|z\|_P} \tag{103}$$

Theorem:
Given an $m \times n$ matrix, $\|A\|_2 = \sigma_1(A) = \sqrt{\lambda_{\max}(AA^T)}$.

9.5.2. Solving the Optimization

Using the norms we discussed, we can use the **Frobenius Norm** in place of the L-2 norm since they will give the same solution even though they might give different arguments.

Theorem:
A non-zero matrix $B \in \mathbb{R}^{m \times n}$ is rank 1 if $\exists x \in \mathbb{R}^m, \exists y \in \mathbb{R}^n : B = xy^T$

Lets assume that B is a rank 1 matrix:

$$B = \begin{bmatrix} B^{(1)} & B^{(2)} & \dots & B^{(n)} \end{bmatrix} \tag{104}$$

with basis $x \in \mathbb{R}^m = span\big(B^{(1)}, B^{(2)}, ..., B^{(n)}\big)$. This means that:

$$\begin{gathered} \exists y_i : B^{(i)} = y_i x, \text{ where } B^{(i)} \text{ is the } i^{\text{th}} \text{ basis vector} \\ \therefore B = \begin{bmatrix} y_1 x & \dots & y_n x \end{bmatrix} = xy^T \end{gathered} \tag{105}$$

With this, we can change the objective function to:

$$\min_{B \in \mathbb{R}^{m \times n}} \|A - B\| \tag{106}$$

This is called **Low Rank Optimization**.

Example: The Netflix Problem

> Lets assume we have a matrix representing a user's rating of a movie. We will then have an extremely large matrix where the ij^{th} index represents user's i rating for movie j. If we say that Netflix has 1 billion users and 50,000 movies, a naive matrix setup will be $10^8 \times 5 \cdot 10^4$. But, we can approximate this using low rank matrix optimization, giving us a $10^8 + 5 \cdot 10^4$ matrix. Even though this greatly decrease the data size, it is unrealistic if we make $rank(B) \leq 1$, which says only one person's rating matters. So, we make $rank(B) \leq k$, where k is very small, saying that k people's rating are important.

Theorem:
$B \in \mathbb{R}^{m\times n}$ satisfies $Rank(B) \leq k$ iff

1. $\exists C \in \mathbb{R}^{m\times k}, \exists D \in \mathbb{R}^{n\times k} : B = CD^T$
2. $\exists C^{(1)}, ..., C^{(k)} \in \mathbb{R}^m, \exists d^{(1)}, ..., d^{(k)} \in \mathbb{R}^n : B = C^{(1)}D^{((1))^T} + ... + C^{(n)}d^{((n))^T}$

C and D are rank 1 matrixes, $m \times 1$ and $n \times 1$, respectively. Adding k of them together will result in a matrix of rank at most k. This means that each column $C^{(i)}d^{((i))^T}$ are not necessarily linearly independent. Additionally, approximating A as a rank-k matrix, we will have $k(m+n)$ amount of data since $C^{(i)}D^{((i))^T}$ has $(m+n)$.

Low-Rank Optimization cont.

Lecture 10

9/30/25

10. Eckart-Young-Mirsky Theorem:

- $b = A_K$ optimal solution of $\min\|A-B\|$ s.t $rank(B) \leq k$
- Optimal solution is unique iff $\sigma_k \neq \sigma_{k+1}$

Proof:
Let rank(A) = r

$$\begin{aligned} A &= \sigma_1 u^{(1)} v^{((1))^T} + ... + \sigma_1 u^{(r)} v^{((r))^T} \\ A_K &= \sigma_1 u^{(1)} v^{((1))^T} + ... + \sigma_1 u^{(k)} v^{((k))^T} \end{aligned} \tag{107}$$

Where $k \leq \min(m,n) \wedge k \leq r$

$$\|A - A_K\|_F^2 = \text{trace } \tilde{A}\tilde{A}^T = \sum_{i=k+1}^{n} \sigma_i^2 \tag{108}$$

This means that:

$$\|A-B\|_F^2 \geq \sum_{i=k+1}^{r} \sigma_i^2, \quad \forall B : \text{rank} \leq k \tag{109}$$

Lemma 1: Trace (X) = $\sum$ eigenvalues of X
Lemma 2: Given $Y, Z \in \mathbb{R}^{m\times n} \forall i,j \geq 1, \sigma_{i(Y)} + \sigma_{i(Z)} \geq \sigma_{1+j-i}(Y+Z)$

From **Lemma 1**:

$$\begin{aligned} \|A-B\|_F^2 &= \text{trace } CC^T - \sum \text{eigenvalues of } CC^T \\ &= \sigma_1^2(C) + ... + \sigma_{\min(m,n)}^2(C) \\ &= \left(\sigma_1(C) + \sigma_{k+1}(B)\right)^2 + ... + \left(\sigma_{\min(m,n)}(C) + \sigma_{k+1}(B)\right)^2 \end{aligned} \tag{110}$$

Remember that B is at most rank k. Therefore, the $(k+1)^{\text{th}}$ singular value is equal to zero.
From **Lemma 2**:

$$\begin{aligned}\sigma_1(C) + \sigma_{k+1}(B) &\geq \sigma_{k+1}^2(A) \\ \sigma_k(C) + \sigma_{\min(m,n)}(B) &\geq \sigma_{\min(m,n)+k}^2(A)\end{aligned} \tag{111}$$

With these two Lemmas, we can conclude:

$$\therefore \|A - B\|_F^2 \geq \sum_{i=k+1}^{r} \sigma_i^2, \quad \forall B : \text{rank} \leq k \tag{112}$$

This means that $A_K = \sum_{i=k+1}^{r} \sigma_i^2(A)$ is the optimal solution.

10.1. Error of Optimal Solution

Let e_k be the error of the approximation:

$$\begin{aligned}e_k &= \frac{\|A - A_K\|_F^2}{\|A\|_F^2} \\ &= \frac{\sigma_{k+1}^2 + ... + \sigma_r^2}{\sigma_1^2 + ... + \sigma_r^2}\end{aligned} \tag{113}$$

Example:
We have a grey scaled image, 256 possible values, of 266×400 pixels. This gives us a matrix A of rank 266. If we set k to be 9, we can approximate the matrix with rank at most 9, which means you are multiplying by a 9×400 matrix to get the original matrix. To know if this is a good approximation, can calculate the error.

$$e_k = \frac{\|A - A_K\|_F^2}{\|A\|_F^2} = \frac{\sigma_{k+1}^2 + ... + \sigma_r^2}{\sigma_1^2 + ... + \sigma_r^2} \tag{114}$$

We could plot the σ_i in terms of the error to find the best k for the approximation.

10.2. Principle Component Analysis (PCA)

Principle Component Analysis handles approximating the matrix without losing too much data.

We can do this be creating a lower dimensional space then projecting the data points onto that space.
Example: Consider data points:

$$x^{(1)}, ..., x^{(m)} \in \mathbb{R}^n \tag{115}$$

Where the average is

$$\frac{1}{m}\left(x^{(1)}, ..., x^{(m)}\right) = \tilde{x} \tag{116}$$

This givens us the centered data point:

$$\tilde{x}^{(i)} = x^{(i)} - \tilde{x}, \quad i = 1, ..., m \tag{117}$$

Find Z, which is the direction of maximum variance
Define line:

$$L : \{\alpha z \mid \alpha \in \mathbb{R}\} \tag{118}$$

This means:

$$\begin{gathered} \alpha_i z \text{ is the projection of } \tilde{x}^{(i)} \\ \alpha_i = \frac{< \tilde{x}^{(i)}, z >}{\|z\|} = z^T \tilde{x}^{(i)} \\ \|z\| = 1, \text{normal vector} \end{gathered} \tag{119}$$

Let $A = Z^T \tilde{X}$

$$AA^T = Z^T \tilde{X} \tilde{X}^T Z, \quad \text{where } \tilde{X} = \begin{bmatrix} \tilde{x}^{(1)} & \dots & \tilde{x}^{(m)} \end{bmatrix} \tag{120}$$

This is not an optimization problem:

$$\max_z Z^T \tilde{X} \tilde{X}^T Z \tag{121}$$

Remember how:

$$\begin{aligned} \|A\|_2 &= \max_{\|w\|_2=1} \|Aw\|_2 \\ &= \max_{\|w\|_2=1} \sqrt{w^T A^T A w} \\ &= \sqrt{\lambda_{\max}(A^T A)} = \sigma_1(A) \end{aligned} \tag{122}$$

Which looks like our optimization.
This means that the optimal objective for our optimization problem is: $\sigma_1^2(\tilde{X}) = \lambda_{\max}(\tilde{X})$ and the optimal solution is $U^{(2)}$.

10.2.1. Solving PCA

To solve PCA, we first have to center the data. This means that we subtract the row entry by the mean of the row:

$$\begin{gathered} x^{(1)} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad x^{(2)} = \begin{bmatrix} 3 \\ 3 \end{bmatrix} \quad x^{(3)} = \begin{bmatrix} 4 \\ 4 \end{bmatrix} \quad x^{(4)} = \begin{bmatrix} 5 \\ 5 \end{bmatrix} \\ X = \begin{bmatrix} 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 \end{bmatrix} \end{gathered} \tag{123}$$

The centralize matrix $\bar{X}$ is given as:

$$\bar{X} = \begin{bmatrix} x_{1,1} - \mu_1 & x_{1,2} - \mu_1 & \dots & x_{1,n} - \mu_1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{n,1} - \mu_n & x_{n,2} - \mu_n & \dots & x_{n,n} - \mu_n \end{bmatrix} \tag{124}$$

This means:

$$\bar{X} = \begin{bmatrix} -1.5 & -0.5 & 0.5 & 1.5 \\ -1.5 & -0.5 & 0.5 & 1.5 \end{bmatrix} \tag{125}$$

We then find the SVD of the matrix:

$$\begin{aligned} \bar{X}\bar{X}^T &= \begin{bmatrix} -1.5 & -0.5 & 0.5 & 1.5 \\ -1.5 & -0.5 & 0.5 & 1.5 \end{bmatrix} \begin{bmatrix} -1.5 & -1.5 \\ -0.5 & -0.5 \\ 0.5 & 0.5 \\ 1.5 & 1.5 \end{bmatrix} \\ &= \begin{bmatrix} 5 & 5 \\ 5 & 5 \end{bmatrix} \end{aligned} \tag{126}$$

$$\begin{aligned} \det\left(\begin{bmatrix} 5-\lambda & 5 \\ 5 & 5-\lambda \end{bmatrix} \right) = 0 \\ \lambda_1 = -10;\ \lambda_2 = 0 \\ \vec{\lambda_1} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix};\quad \vec{\lambda_2} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \end{aligned} \tag{127}$$

We know that the first principal component is the eigenvector corresponding to σ_1^2. This eigenvector points in the direction of the most variances so that the residual of the projection is minimized.

$$\begin{aligned} Z &= \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}^T \begin{bmatrix} -1.5 & -1.5 \\ -0.5 & -0.5 \\ 0.5 & 0.5 \\ 1.5 & 1.5 \end{bmatrix} \\ &= \begin{bmatrix} -\frac{3}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{3}{\sqrt{2}} \end{bmatrix} \end{aligned} \tag{128}$$

10.3. Deflated Data Matrix

$\hat{X} = \begin{bmatrix} \hat{x}^{(1)} & \dots & \hat{x}^{(m)} \end{bmatrix} = \left(I_n - U^{(1)} U^{((1))^T} \right)$ We now need to find the best direction for $\hat{X}$

$$\max_z Z^T \hat{X} \hat{X}^T Z \tag{129}$$

Using the concept before, the optimal objective is $\sigma^2\left(\hat{X}\right) = \sigma^2\left(\tilde{X}\right)$, making the optimal solution $U^{(2)}$

Conditional Numbers

Lecture 11

10/2/25

10.4. Condition Numbers

The **Condition Number** describes how sensitive the output is to changes in the input. If the condition number of a matrix is close to 1, we say it is well conditioned. If the condition number of a matrix is very large, we say it is ill conditioned.

$$\mathrm{K}(A) = \frac{\sigma_1}{\sigma_n} \tag{130}$$

Instead of having $Ax = y$, we now have:

$$\begin{aligned}
(A + \Delta A)(x + \Delta x) &= y \\
\underbrace{Ax}_{y} + A\Delta x + \Delta A x + \Delta A \Delta x &= y \\
\Delta x &= A^{-1}\Delta A(x + \Delta x) \\
\|\Delta x\|_2 &= \left\| \underbrace{A^{-1}}_{B} \underbrace{\Delta A(x + \Delta x)}_{y} \right\|_2 \\
\|\Delta x\|_2 = \|By\|_2 &\le \|B\|_2 \cdot \|y\|_2 \\
&\le \left\|A^{-1}\right\|_2 \left\| \underbrace{\Delta A}_{\bar{B}} \underbrace{(x + \Delta x)}_{y} \right\|_2 \\
&\le \underbrace{\left\|A^{-1}\right\|_2 \cdot \|A\|_2}_{\mathrm{K}(a)} \cdot \|x + \Delta x\|_2
\end{aligned} \tag{131}$$

10.4.1. Condition Numbers for Leasts Squares

Remember that Least Square, we want to minimize:

$$\min_{x \in \mathbb{R}^n} \|Ax - y\| \tag{132}$$

And the solution space is $S = A^{+}y + N(A)$, where:

$$\begin{aligned}
A^{+} &= \left(AA^T\right)^{-1} A \\
&= \left(U\Sigma V^T V \Sigma^T U^T\right)^{-1} U \Sigma V^T \\
&= \left(U \Sigma \Sigma^T U^T\right)^{-1} U \Sigma V^T \\
&= V \Sigma^{+} U^T
\end{aligned} \tag{133}$$

Now we can take $y = y + \Delta y$

$$\min_{x\in\mathbb{R}^n}\|Ax-(y+\Delta y)\| \tag{134}$$

Now the solution space becomes $S_P = A^+(y+\Delta y)+N(A)$
The main difference is the pseudo inverse of S and S_P, A^+y vs $A^+(y+\Delta y)$
Lets assume x^* is the minimum solution to LS and $x^*+\Delta x$ is the minimum solution to LSP. This means:

$$\begin{aligned} S &= A^+y+N(A) = x^*+N(A) \\ S_P &= A^+(y+\Delta y)+N(A) = x^*+\Delta x+N(A) \\ &= S+A^+\Delta y \end{aligned} \tag{135}$$

Now Assume Δy is an arbitrary vector where $\|\Delta y\|\le 1$, we can then define:

$$\mathrm{E} = \{\Delta x = A^+\Delta y \mid \|\Delta y\|\le 1\} \tag{136}$$

Which is an ellipsoid

Example:
We have a circle

$$\begin{aligned} B &= \left\{x\in\mathbb{R}^2 \mid x_1^2+x_2^2\le r^2\right\} \\ &= \left\{x\in\mathbb{R}^2 \mid \frac{x_1^2+x_2^2}{r^2}\le 1\right\} \\ &= \left\{x\in\mathbb{R}^2 \mid [x_1 \quad x_2][x_1 \quad x_2]\begin{bmatrix}\frac{1}{r^2} & 0\\ 0 & \frac{1}{r^2}\end{bmatrix}\le 1\right\} \\ &= \left\{x\in\mathbb{R}^2 \mid \underbrace{[x_1 \quad x_2]\begin{bmatrix}\frac{1}{r} & 0\\ 0 & \frac{1}{r}\end{bmatrix}}_{y^T}\underbrace{\begin{bmatrix}\frac{1}{r} & 0\\ 0 & \frac{1}{r}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}}_{y}\le 1\right\} \\ &= \left\{x\in\mathbb{R}^2 \mid y=\begin{bmatrix}\frac{1}{r} & 0\\ 0 & \frac{1}{r}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}, y^Ty\le 1\right\} \\ &= \left\{x\in\mathbb{R}^2 \mid \begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}\frac{1}{r} & 0\\ 0 & \frac{1}{r}\end{bmatrix}^{-1}\begin{bmatrix}y_1\\ y_2\end{bmatrix}, \|y\|_2\le 1\right\} \end{aligned} \tag{137}$$

Therefore, a Ball in $\mathbb{R}^n$:

$$\begin{gathered} \left\{x\in\mathbb{R}^n \mid x_1^2+\ldots+x_n^2\le r^2\right\} \\ \left\{x\in\mathbb{R}^n \mid x=\underbrace{\begin{bmatrix}\frac{1}{r} & 0\\ 0 & \frac{1}{r}\end{bmatrix}}_{B}^{-1}y, \quad \|y\|_2\le 1\right\} \\ \left\{x\in\mathbb{R}^n \mid x=B^{-1}y, \quad \|y\|_2\le 1\right\} \end{gathered} \tag{138}$$

We can expand:

$$\|Bx\|_2 = x^T BB^T x \tag{139}$$

We can define $P = BB^T$. Since we assume that B is invertible, P is also invertible. This means we can rewrite the set as:

$$\left\{x \in \mathbb{R}^n \mid x^T P^{-1} x \leq 1\right\} \tag{140}$$

This is the definition if an n-dimensional ellipsoid.
The characteristic of the ellipsoid is that each eigenvector are the direction of a point in the ellipsoid surface and the length is the corresponding singular values.

What is B is not square or invertible, consider:

$$\{x \in \mathbb{R}^n \mid x = By, \|y\|_2 \leq 1\} \tag{141}$$

where $B \in \mathbb{R}^{n \times m}$, $y \in \mathbb{R}^m$, and the rank(B) = r
The ellipsoid now is in a lower dimensional space r, the semi-axis are the still the eigenvectors and the lengths are the singular values.

In the case of

$$\mathrm{E} = \{\Delta x = A^+ \Delta y \mid \|\Delta y\| \leq 1\} \tag{142}$$

This is an ellipsoid with semi axis: $v^{(1)}, ..., v^{(n)}$ with lengths: $\frac{1}{\sigma_1}, ..., \frac{1}{\sigma_r}, 0, 0$

This says that any change in the solution space belongs to a different ellipsoid. The main difference between $y = Ax$ and leasts square is that leasts squares takes into account all singular values instead of only the largest and smallest.

Sets

Lecture 13

10/9/25

11. Sets and Combinations

11.1. Types and Properties of Sets

A set $\mathcal{S}$ **Open** if

$$\forall x \in \mathcal{S}, \exists \varepsilon > 0 : \forall \alpha \leq \varepsilon, \sqrt{x^2 - (\alpha\varepsilon)^2} \subseteq \mathcal{S} \tag{143}$$

Intuitively, this means that for a set $\mathcal{S}$ to be open, we can create a circle around any element in the set, no matter how small the radius is, and every element within that circle is a subset of the set.

Let C be the edge of $\mathcal{S}$, then we know that $\forall x \in \mathcal{S}, x < C$, this means that, no matter how close x is the C, we can take $\varepsilon = C - x > 0$

A set is $\mathcal{S}$ **Closed** if

$$\forall \varepsilon > 0, \exists x \in \mathcal{S} : (x + \varepsilon) \nsubseteq \mathcal{S} \tag{144}$$

Notice that this is the negation of the open set.

This means that if $\mathbb{R} \setminus \mathcal{S}$ is open, then $\mathcal{S}$ is closed. By definition, $\mathbb{R}^n$ & $\emptyset$ are open and closed.

The **Interior** of a set is defined as the set points strictly inside $\mathcal{S}$

$$\text{int}(\mathcal{S}) = \{Z \in S \mid B(z) \subseteq \mathcal{S}\} \tag{145}$$

The **Closure** of a set is defined as

$$\text{cls}(\mathcal{S}) = \left\{z \in \mathbb{R}^n \mid z = \lim_{k \to \infty} x^{(k)}; \forall k, x^{(k)} \in \mathcal{S}\right\} \tag{146}$$

The **Boundary** of a set $\mathcal{S}$ is

$$\delta\mathcal{S} = \text{cls}(\mathcal{S}) \setminus \text{int}(\mathcal{S}) \tag{147}$$

A set $\mathcal{S}$ is **Bounded** if:

$$\exists \text{ ball of radius r} : S \subseteq \text{ball} \tag{148}$$

A set $\mathcal{S}$ is **Compact** if it is both **Closed** and **Bounded**

11.2. Linear and Affine Combinations

A **Linear Combination** of a point x is the set of point and its coefficient that could express x:

$$X = \alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_n x_n \tag{149}$$

The **Linear Hull** is the set of all linear combinations of $\{x_1, x_2, ..., x_n\}$ = span$(x_1, x_2, ..., x_n)$

The **Affine Combination** of a set of points $\{x_1, x_2, ..., x_n\}$ is

$$\begin{gathered} X = \alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_n x_n \\ \alpha_1 + \alpha_2 + ... + \alpha_n = 1 \\ \forall \alpha_i \in \mathbb{R} \end{gathered} \tag{150}$$

The **Affine Hull** is the set of all affine combinations

Theorem:
The Affine hull of any set is an affine set.

11.3. Convexity

A **Convex Combination** for points $\{x_1, x_2, ..., x_n\} \in \mathbb{R}^n$ is

$$\begin{gathered} X = \alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_n x_n \\ \alpha_1 + \alpha_2 + ... + \alpha_n = 1 \\ \forall \alpha_i \geq 0 \in \mathbb{R} \end{gathered} \tag{151}$$

The **Convex Hull** is the set of all convex combinations of $\{x_1, x_2, ..., x_n\} \in \mathbb{R}^n$

A **Convex Set** $\mathcal{S}$ is a set where for every two points $x_1, x_2 \in \mathcal{S}$ in the set, the convex combination is also in the set.

$$\mathcal{S} \text{ is convex if } \forall \alpha \in [0,1], \forall x_1, x_2 \in \mathcal{S}, \alpha x_1 + (1-\alpha)x_2 \in \mathcal{S} \tag{152}$$

Theorem:
The **Convex Hull** is the smallest convex set that contains all points in the set

The dimension of **Convex Set** is the dimension if the affine hull of the set The union of two **Convex Sets** is not always convex The intersection of two **Convex Sets** is always convex

Hyperplanes

Lecture 14

10/14/25

12. Hyperplanes

12.1. Supporting Hyperplane

Given a convex set C and a boundary point $z \in \delta C$, the hyperplane

$$H = \{x \in \mathbb{R}^n \mid a^T x = b\} \tag{153}$$

is called a supporting hyperplane for C at z if z is on the hyperplane and the boundary of C and the set C is only on one side of the hyperplane.

- $z \in H(a^T z = b)$
- $C \subset H_-(H_- = \{x \in \mathbb{R}^n \mid a^T x \leq b\})$, meaning $a^T x \leq b, \forall x \in C$

The following theorem states that the supporting hyperplane always exists.

if $C \subset \mathbb{R}^n$ is a convex set and z is on the boundary of C, then $\exists$ a supporting hyperplane containing z.

For a given boundary point z, the supporting hyperplane at z may or may not be unique.

12.2. Separating Hyperplane

Given $C_1, C_2 \subset \mathbb{R}^n$: Convex Set

$$H = \{x \in \mathbb{R}^n \mid a^T x = b\} \tag{154}$$

is a separating hyper plane if C_1 is on one side of H and C_2 is on the other.

1. if $C_1 \wedge C_2 = \emptyset$, then a separating hyperplane exists.
2. if C_1 and C_2 are closed & one bounded $C_1 \wedge C_2 = \emptyset$, then a strict separation exists.

If the intersection of H and $C_1, C_2 = \emptyset$, this is called a strict separation.

12.3. Convex Functions

A **Convex Function** $f : \mathbb{R}^n \to \mathbb{R}$ is convex if the domain of f is defined as the following set

$$f = \{x \in \mathbb{R}^n \mid -\infty < f(x) < \infty\} \tag{155}$$

Assume that $f(x) = \log(x)$, then the domain $dom(f) = (0, \infty)$ Assume that $f(x) = \frac{a^T x + b}{c^T x + d}$, then the domain $dom(f) = \{x \in \mathbb{R}^n \mid c^T x + d \neq 0\}$

Formally, we say that f is a convex function if:

1. $dom(f)$ is a convex set.
2. $\forall x, y \in dom(f),\ \alpha \in [0,1],\ f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y)$
   - if the inequality is only less than, then it is a strictly convex function
3. The complement is a concave function.

12.4. Convexity of Norms

Theorem: Let $f : \|\cdot\|\ \mathbb{R}^n \to \mathbb{R}$, $f(x)$ is convex if

- $dom(f) : \mathbb{R}^n \in$ Convex Set
- $\|\alpha x + (1-\alpha)y\| \leq \alpha\|x\| + (1-\alpha)\|y\|,\ \alpha \in [0,1]$

If $f_1(x),\ ...,\ f_{m(x)}$ are convex functions, $\omega_1,\ ...,\ \omega_m > 0$, then

$$\begin{gathered} f(x) = \omega_1 f_1(x) + ... + \omega_m f_{m(x)} \text{ is convex} \\ dom(f) = dom(f_1) \cap ... \cap dom(f_m) \end{gathered} \tag{156}$$

Convexity

Lecture 15

10/16/25

13. Properties of Convexity

A convex function is not just a unique that allows us to simplify optimization, it is also easier to work with because the summation of convex functions f_1, f_2 is still a convex function. There are other operations that preserves convexity, such as the **Affine Transformation**.

An **Affine Transformation** is a geometric transformation that preserves straight lines, parallel lines, and the ratios of distances between points, but does not necessarily preserve angles or lengths. The affine transformation on $f(x)$ is $g(x) = f(Ax + b)$.

Let $f : \mathbb{R}^n \to \mathbb{R}$, $A \in \mathbb{R}^{n \times m}$, $B \in \mathbb{R}^n$, we can define $g(x) = f(Ax + b)$, which is a convex function:

$$domain(g) = \{x \mid Ax + b \in f\} \tag{157}$$

This is a convex set since $\alpha x_1 + (1-\alpha)x_2 \in domain(g)$. Let $x = \alpha x_1 + (1-\alpha)x_2$

$$A(\alpha x_1 + (1-\alpha)x_2) + b = \alpha A x_1 + (1-\alpha) A x_2 + b \tag{158}$$

Let $b = \alpha b + (1-\alpha)b$

$$\begin{aligned} &\alpha Ax_1 + (1-\alpha)Ax_2 + \alpha b + (1-\alpha)b \\ &\alpha\underbrace{(Ax_1+b)}_{\in f} + (1-\alpha)\underbrace{(Ax_2+b)}_{\in f} \in domain(g) \end{aligned} \tag{159}$$

This is also a convex function:

$$\begin{aligned} f(\alpha x_1 + (1-\alpha)x_2) &\leq \alpha f(x_1) + (1-\alpha)f(x_2) \\ g(\alpha x_1 + (1-\alpha)x_2) &= f(A(\alpha x_1 + (1-\alpha)x_2) + b) \\ &\leq \alpha f(Ax_1+b) + (1-\alpha)f(Ax_2+b) \\ &= \alpha g(x_1) + (1-\alpha)g(x_2) \\ g(\alpha x_1 + (1-\alpha)x_2) &\leq \alpha g(x_1) + (1-\alpha)g(x_2) \text{ Convex} \end{aligned} \tag{160}$$

What we have been doing so far is the **Zeroth Order Convexity Condition**:

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha)f(x_2) \tag{161}$$

We also have **First Order Convexity Condition**, which require the function to be differentiable. Assume that the domain of the function f is open, convex, and differentiable on the domain, we say f is convex **if and only if**

$$\forall x, y \in domain(f), f(y) \geq f(x) + \nabla f(x)^T(y-x) \tag{162}$$

Notice that the right side is the first order Taylor approximation of f(y). This means intuitively that this condition holds if the tangent line on any point in the domain must be below the function.

Example:
Let f be a convex function, using the zeroth order condition:

$$\begin{aligned} f(\alpha x_1 + (1-\alpha)x_2) &\leq \alpha f(x_1) + (1-\alpha)f(x_2) \\ \frac{f(\alpha x_1 + (1-\alpha)x_2)}{1-\alpha} &\leq \frac{\alpha f(x_1) + (1-\alpha)f(x_2)}{1-\alpha} \\ \frac{f(\alpha x_1 + (1-\alpha)x_2)}{1-\alpha} &\leq f(x_2) - f(x_1) \end{aligned} \tag{163}$$

Let $\beta = 1-\alpha$ approach zero:

$$\begin{aligned} f(x_2) - f(x_1) &\geq \lim_{\beta\to 0} \frac{f(x_1 + \beta(x_2 - x_1)) - f(x_1)}{\beta} \\ &= \lim_{\beta\to 0} \frac{\cancel{f(x)} + \nabla f(x)^T\beta(y-x) + \text{higher order terms} - \cancel{f(x)}}{\beta} \\ &= \nabla f(x)^T\beta(y-x) + \cancel{\lim_{\beta\to 0} \frac{\text{higher order terms}}{\beta}} \\ &\therefore f(x_2) - f(x_1) \geq \nabla f(x)^T\beta(y-x) \end{aligned} \tag{164}$$

An easier condition to use is the **Second Order Convexity Condition**. This condition states that a function f is convex **if and only if** $\forall x \in domain(f), \nabla^2 f \geq 0$

In the case if $\nabla^2 f > 0$, positive definite, then f is strictly convex.

$$\nabla^2 f > 0 \longrightarrow f \text{ is strictly convex} \tag{165}$$

This is a one way implication, so f being strictly convex does not imply that $\nabla^2 f > 0$

Convex Optimization

Lecture 16

10/21/25

14. Convex Optimization

A canonical form of general optimization problems:

$$\begin{gathered} \min_{x \in \mathbb{R}^n} f_0(x) \\ f_i(x) \leq 0; \qquad h_j(x) = 0 \end{gathered} \tag{166}$$

These optimization problems are convex or non-convex.

For an optimization to be convex, the following conditions must be true:

- $f_0(x)$ is a convex function
- $f_i(x)$ are convex functions
- $h_j(x)$ must be an affine function

The reason why $h_j(x)$ must be an affine function rather than a convex function is because when putting it in standard form with two inequalities, one of the inequalities will be concave.

Example:

$$\begin{gathered} \min_{x \in \mathbb{R}^3} e^{x_1 - x_2} + (2x_1 + x_2)^2 \\ (x_1 + x_2)^6 \leq 5x_1 - x_2 + 1 \\ -1 \leq x_3 \leq 1 \\ x_1 + x_2 + x_3 = 5 \end{gathered} \tag{167}$$

First put in canonical form:

$$\begin{gathered} \min_{x \in \mathbb{R}^3} e^{x_1 - x_2} + (2x_1 + x_2)^2 \\ f_i \begin{cases} (x_1 + x_2)^6 - 5x_1 + x_2 - 1 \leq 0 \\ x_3 - 1 \leq 0 \\ -x_3 + 1 \leq 0 \end{cases} \\ h\{x_1 + x_2 + x_3 - 5 = 0 \end{gathered} \tag{168}$$

We then check the conditions highlight above to check if the optimization problem is convex.

14.1. Feasible Set

A **Feasible Set** is a set of possible solutions::

$$\mathcal{S} = \left\{ x \in \mathbb{R}^n \,\middle|\, f_i(x) \leq 0; h_j(x) = 0 \right\} \tag{169}$$

Theorem: The feasible set of a convex optimization problem is a convex set.

Proof:
Let $x, y \in \mathcal{S}; i = 1, 2, ..., m; j = 1, 2, ..., k$
if $x \in \mathcal{S} \rightarrow f_i(x) \leq 0; h_j(x) = 0$
if $y \in \mathcal{S} \rightarrow f_i(y) \leq 0; h_j(y) = 0$
Let $z = \alpha x + (1 - \alpha) y \in \mathcal{S}$, where $\alpha \in [0, 1]$

$$\begin{aligned} f_i(z) = f_i(\alpha x + (1-\alpha)y) &\leq \alpha \underbrace{f_i(x)}_{\leq 0} + (1-\alpha) \underbrace{f_i(y)}_{\leq 0} \leq 0 \\ h_j(z) = h_j(\alpha x + (1-\alpha)y) &\leq \alpha \underbrace{h_j(x)}_{=0} + (1-\alpha) \underbrace{h_j(y)}_{=0} = 0 \end{aligned} \tag{170}$$

$$\therefore z \in \mathcal{S} \text{ is a convex set}$$

Theorem: For convex optimization:

- All local solutions are global solutions
- The set of all global minimums is convex

Coercive Functions

Lecture 17

10/30/25

14.2. Coercive Function

A function is **Coercive** if:

$$f : \mathbb{R}^n \rightarrow \mathbb{R} : \lim_{\|x\| \rightarrow \pm\infty} f(x) = +\infty \tag{171}$$

This describes the behavior of the function's extreme points rather the how the graph between. This means the the graph can be convex or non-convex and still be coercive.

Example:
Given $f(x) = \alpha_1 x_1^4 + \alpha_2 x_2^4$, this function is coercive if $\alpha_1, \alpha_2 > 0$. Proof:

$$\begin{aligned} f(x) &\geq \min(\alpha_1, \alpha_2)(x_1^4 + x_2^4) \\ &\geq \min(\alpha_1, \alpha_2) \frac{\left(x_1^2 + x_2^2\right)^2}{2} \\ &= \min(\alpha_1, \alpha_2) \frac{\|x\|^2}{2} \\ \therefore \lim_{\|x\| \rightarrow \pm\infty} f(x) &= +\infty \end{aligned} \tag{172}$$

Theorem:
Consider $f : \mathbb{R}^n \to \mathbb{R}$, domain $f \in \mathbb{R}^n$, if f is continuous and coercive, then $\min f(x)$ has a solution.

1. $\min f(x) \to dom(f) \in \mathbb{R}^n \quad s.t. x \in S$, if f is continuous & coercive and S is closed, then it has a finite solution.
2. $\min f_0(x) \quad s.t. \quad f_i(x) \leq 0; h_j(x) = 0$. Assume f_0 is continuous and coercive, if f_i, h_j is continuous and has domain $\mathbb{R}^n$, then it has a finite solution.

14.3. Weierstrass Theorem
Consider minimizing a continuous function $f(x) : \mathbb{R}^n \to \mathbb{R} \quad s.t. \quad x \in S$, there exist a finite solution if the feasible set S is compact (closed & bounded).

Example:

$$\begin{aligned} &\min e^{x_1} - x_1^2 + x_2^2 + e^{x_1 + x_2} \\ s.t. \quad &\cos(x_1) + \sin(x_2) \leq 1 \\ &x_1^4 + e^{x_2} \leq 5 \end{aligned} \tag{173}$$

Notice that all the functions are defined in $\mathbb{R}^n$, which means its continuous and therefore closed. We can also show that the feasible set is bounded by observing that $x_1^4 + e^{x_2} \leq 5$.

All these theorems are sufficient conditions of the existence of a solution.

14.4. Strictly Convex
If f_0 is strictly convex, then there is either a unique solution or no solution.

Proof:
Assume at least two solutions $x^*,\ y^*$, due to convexity, $\frac{1}{2}x^* + \frac{1}{2}y^* \in S$:

$$\begin{aligned} f_0\left(\frac{1}{2}x^* + \frac{1}{2}y^*\right) &< \frac{1}{2}f_0(x^*) + \frac{1}{2}f_0(y^*) \\ &= f_0(x^*) \end{aligned} \tag{174}$$

This shows that there exists a point more optimal than the optimal solution. **Contradiction**.

15. Linear Programming
For an optimization problem to be a **Linear Program**, the objective function and constraints must be affine.

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} a_0^T x - b_0 \\ s.t. \quad &a_i^T x - b_i = 0 \\ &a_j^T x - b_j \leq 0 \end{aligned} \tag{175}$$

Rearranging in matrix form, we get:

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} a_0^T x \\ s.t. \quad &Ax = B \\ &Cx \leq d \quad \text{element-wise} \end{aligned} \tag{176}$$

The special property of linear programming is that the feasible set, the intersection of all the constraint, is a polyhedron since the constraints are hyperplanes and half-spaces.

15.1. Solving Linear Programs

Given a convex set S, we say a point $y \in S$ is an **Extreme Point** is $\nexists \alpha \in (0,1)\ s.t.\ \alpha U + (1-\alpha)V = y, U, V \in S$. This means that y is not a convex combination of any points in S. Usually, extreme points are at the border of the set.

Linear Programming

Lecture 18

11/4/25

15.2. Standard Linear Program

For Linear Programs, our constraints must be an equality. To convert inequality constraints to equality constraints, we add a **Stack Variable** as an equalizer:

$$c_1 x \leq d \implies c_1 x + \overset{\text{stack}}{\widehat{s}} = d, \quad s \geq 0 \tag{177}$$

Additionally, all the variables used must be non-negative:

$$x_1, ..., x_2 \geq 0 \tag{178}$$

This means our constraints can take the form of:

$$Ay = b \tag{179}$$

and we can find a unique solution if the matrix A is linearly independent.

15.3. Vertices of a Linear Program

If all the constraints of the Linear Program are affine, then there exist a solution vertex if the LP has a solution. Why this is the case is because any point in a polyhedron can be expressed as a convex combination of all vertices. This means that there is a vertex at least as small as any point on the polyhedron.

Consider a non-degenerate Linear Program with $v_1, ..., v_n$ as the vertices of the feasible set. Two vertices v_i, v_j are adjacent if and only if there are $n - m - 1$ zero entries in both vectors on the same dimension, where n is the number of dimensions and m is the number of constraints.

15.4. The Simplex Algorithm

Simplex is the most commonly used algorithm for solving Linear Program by utilizing the concept of adjacent vertices. It does this by:

1. Find an arbitrary vertex y
2. Find adjacent vertices of y
3. If $\exists \tilde{y}\ s.t.\ a_0^T \tilde{y} < a_0^T y$, go to $\tilde{y}$
4. repeat from step two on $\tilde{y}$

15.5. Epigraph Formulation

When the function is does not fit the linear model, we cen use an **Epigraph** to formulate it into such:

$$\min f(x) \ s.t. \ x \in S \quad \Longrightarrow \min t \ s.t. \ x \in S, \ f(x) \leq t \tag{180}$$

Quadratic Programming

Lecture 19

11/6/25

16. Quadratic Programming

A **Quadratic Program** is an optimization with a quadratic objective function and linear constraints:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & x^\top P_0 x + q_0^\top x + r_0 \\ s.t. \quad & a_i^\top x = b_i \\ & c_j^\top x \leq d_j \end{aligned} \tag{181}$$

For convexity, we need the Hessian $2P_0 \succeq 0$. One thing to point out is that if $P_0 = 0$, then the quadratic program reduces to a linear program. $LP \subset QP$

16.1. Quadratically Constrained Quadratic Program

This is a subset of a Quadratic Program, taking the form of:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & x^\top P_0 x + q_0^\top x + r_0 \\ s.t. \quad & x^\top P_i x + q_i^\top x + r_i \leq 0 \\ & a_j^\top x = b_j \end{aligned} \tag{182}$$

For convexity, we need the Hessian $2P_i \succeq 0$. If $P_i = 0$, then the Quadratically Constrained Quadratic Program reduces to a Quadratic Program. $QP \subset QCQP$.

From the constraints given, the quadratic constraints describe an ellipsoid and the linear constraints describe a hyperplane. This means that the feasible set is the intersection of ellipsoids and hyperplanes.

17. Relaxation

17.1. Convex relaxation

Convex relaxation is the technique of reducing non-convex optimization problem to convex optimization problem. Let $\tilde{S}$ be convex and $S \subseteq \tilde{S}$

$$\begin{aligned} &\min f_0(x) \to \text{convex} && \min f_0(x) \\ &s.t \quad x \in S \to \text{non-convex} && s.t \quad x \in \tilde{S}; \end{aligned} \tag{183}$$

Theorem:
Let x^*: arbitrary global min of non-convex optimization
Let $\tilde{x}$: arbitrary global min of convex relaxation

1. $f(\tilde{x}) \leq f(x^*)$
2. if $\tilde{x} \in S$, then $\tilde{x}$ is the solution to the original problem

If the feasible set of a non-convex problem (I) is a subset of a convex problem (II), then (II) is the convex relaxation of (I).

Example:

$$\begin{aligned} \min \quad & f_0(x) \\ s.t. \quad & x_1^2 + x_2^2 = 1 \\ & x_1 + x_2 \leq 1 \end{aligned} \tag{184}$$

We can relax this by converting the equality into an inequality.

17.2. Integer Programing

Integer Programming is an optimization problem that constraints the feasible points to integers:

$$\begin{aligned} \min \quad & a^\top x \\ s.t. \quad & Ax = b \\ & x \geq 0 \in \mathbb{Z} \end{aligned} \tag{185}$$

A convex relaxation on an IP is simply turning the integer constraint into an interval.

Theorem: If all vertices of the LP are integral, all integers elements, then the convex relaxation is exact.

Optimality Conditions

Lecture 20

11/13/25

17.2.1. Application of Integer Programming

Assume we have m suppliers and n customers. Each supplier can produce a certain number of units, and each customer has a certain demand. The cost of transporting goods from supplier i to customer j is given by c_{ij}. The goal is to minimize the total transportation cost while satisfying supply and demand:

$$\begin{aligned} \min_{x \in \mathbb{R}^{m \times n}} \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \\ s.t. \quad & \sum_{j=1}^{n} x_{ij} \leq s_i \quad \forall i = 1, ..., m \\ & \sum_{i=1}^{m} x_{ij} \geq d_j \quad \forall j = 1, ..., n \end{aligned} \tag{186}$$

where s_i is the supply of supplier i and d_j is the demand of customer j. The variable x_{ij} represents the number of units transported from supplier i to customer j.

Another example is compressed sensing, where we want to recover a sparse signal $x \in \mathbb{R}^n$ from a limited number of linear measurements $y = Ax +$ noise, where $A \in \mathbb{R}^{m \times n}$ is the measurement matrix with $m < n$. The goal is to find the sparsest solution that fits the measurements:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \|x\|_0 \\ s.t. \quad & \|Ax - y\|_2 \leq \varepsilon \end{aligned} \tag{187}$$

where $\|x\|_0$ counts the number of non-zero entries in x, and ε is a small tolerance for the measurement noise. This problem can be relaxed to an l_1-norm minimization problem, which is convex and can be solved efficiently:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \|x\|_1 \\ s.t. \quad & \|Ax - y\|_2 \leq \varepsilon \end{aligned} \tag{188}$$

One note is that the l_1-norm minimization is a relaxation of the l_0 minimization only if the feasible set is restricted to

$$x \in \{x \in \mathbb{R}^n \mid -1 \leq x_i \leq 1 \quad \forall i = 1, ..., n\} \tag{189}$$

Theorem:
if A is random enough and m is sufficiently large (on the order of $k \log\left(\frac{n}{k}\right)$ where k is the sparsity level of the true signal), then the solution to the l_1-norm minimization problem is equal to the solution of the original l_0-norm minimization problem.

Duality of LP and QP

Lecture 21

11/18/25

17.3. Applications of LASSO: Cyber Attack Detection for Power Networks

We can model a Power Network as a graph where nodes represent buses (points of power generation, consumption, or distribution) and edges represent transmission lines. We can think of flow on each edge from node i to node j as traffic in the network, which we denote as P_{ij}. This means that $P_{ij} = -P_{ji}$. To simplify the model, we assume conservation of flow on each node, in other words:

$$\sum_{i,j \in \text{ edges}} P_{ij} = P_i \tag{190}$$

where P_i is the net flow at node i.

Going back to notations, generators are nodes that produce power, so $P_i > 0$ for generators. Consumers are nodes that consume power, so $P_i < 0$ for consumers. Transmission nodes are nodes that neither produce nor consume power, so $P_i = 0$ for transmission nodes.

We know that the votage at a node i is defined as:

$$V_i = \text{mag} \times \cos(\omega t + \theta_i) \tag{191}$$

where "mag" is the magnitude, ω is the frequency, t is time, and θ_i is the unknown phase $\in [-\pi, \pi]$.

Let $\theta_1^*, ..., \theta_n^*$ be the unknown phases of nodes $1, ..., n$ at time $t = 0$. We can define the reactance z_{ij} as:

$$z_{ij} = \frac{\theta_i^* - \theta_j^*}{P_{ij}} \tag{192}$$

We can take a node k as the reference node, and set $\theta_k^* = 0$. Then, $\hat{N}$ is the set of nodes with senosrs and $\hat{E}$ is the set of edges with sensors, with $\hat{P}_i$ is the flow of nodes with sensors. We can express $\hat{P}_i$ as:

$$\begin{aligned} \hat{P}_i &= \sum_{j \in \hat{N}} P_{ij} + w_i^* + v_i^* \\ &= \sum_{j \in \hat{N}} \frac{\theta_i^* - \theta_j^*}{z_{ij}} + w_i^* + v_i^* \end{aligned} \tag{193}$$

where w_i^* is the sensor noise and v_i^* is the attack vector.

Using LASSO, we can estimate the attack vector v^* by solving the following optimization problem:

$$\begin{aligned} \min_{\theta, \omega, v} \quad & \|\omega\|_2^2 + \lambda \|v\|_1 \\ s.t. \quad & \hat{P}_i = \sum_{j \in \hat{N}} \frac{\theta_i - \theta_j}{z_{ij}} + \omega_i + v_i, \forall i \in \hat{N} \\ & \hat{P}_{ij} = \frac{\theta_i - \theta_j}{z_{ij}} + \omega_i + v_i, \forall i, j \in \hat{E} \\ & \theta_1 = 0 \end{aligned} \tag{194}$$

where $\lambda > 0$ is a regularization parameter that controls the sparsity of the attack vector v.

17.4. More Optimality Conditions

For an unconstrained optimization problem:

$$\min_x f(x) \tag{195}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a differentiable function, we can use the gradient condition to find the necessary condition for optimality.

17.4.1. Relative Interior

Assume we have a convex set D and a affine hull of the set $\text{aff}(D)$, we say that a point $x \in D$ is in the relative interior of D, denoted as $\text{relint}(D)$, if there exists a ball $B(x, r)$ with radius $r > 0$ such that:

$$B(x, r) \in \text{aff}(D) \subseteq D \tag{196}$$

In other words, there exists a neighborhood around x that lies entirely within the set D when restricted to the affine hull of D.

Assume that in $\mathbb{R}^3$, your feasible set is a raised disc. The affine hull of this set is the plane containing the disc. The reason why we use the affine hull is so that the ball can be "flattened" to fit within the disc.

Recall that the standard Optimization Problem is defined as:

$$\begin{aligned} &\min_x f_0(x) \to \text{convex} \\ s.t. \quad &f_{i(x)} \leq 0, \quad \forall i = 1, ..., k \\ &h_{j(x)} = 0, \quad \forall j = 1, ..., m \end{aligned} \tag{197}$$

17.5. Optimality Conditions for Convex Optimization (KKT Conditions)

The Karush-Kuhn-Tucker (KKT) conditions provide necessary and sufficient conditions for a solution to be optimal in a convex optimization problem. It says that if Slater's condition is satisfied, then x^* is a global minimum iff

$$\begin{aligned} \exists \lambda^* = \begin{bmatrix} \lambda_1^* & ... & \lambda_k^* \end{bmatrix}^\top \\ \exists \mu^* = \begin{bmatrix} \mu_1^* & ... & \mu_m^* \end{bmatrix}^\top \end{aligned} \tag{198}$$

Which are Lagrange multipliers such that the following conditions are satisfied:

1. **Primal Feasibility**:

$$\begin{aligned} f_i(x^*) \leq 0, \quad \forall i = 1, ..., k \\ h_j(x^*) = 0, \quad \forall j = 1, ..., m \end{aligned} \tag{199}$$

2. **Dual Feasibility**:

$$\lambda_i^* \geq 0, \quad \forall i = 1, ..., k \tag{200}$$

3. **Complementary Slackness**:

$$\lambda_i^* f_i(x^*) = 0, \quad \forall i = 1, ..., k \tag{201}$$

4. **Stationarity**:

$$\nabla f_0(x^*) + \sum_{i=1}^{k} \lambda_i^* \nabla f_i(x^*) + \sum_{j=1}^{m} \mu_j^* \nabla h_j(x^*) = 0 \tag{202}$$

Consider the following unconstrained optimization problem:

$$\min_x f(x) \tag{203}$$

In this case, the KKT conditions reduce to the gradient condition:

$$x^* \text{ is a global min} \iff \nabla f(x^*) = 0 \tag{204}$$

because there are no constraints, so the primal feasibility, dual feasibility, and complementary slackness conditions are trivially satisfied.

Example:

$$\begin{aligned} \min_{x} \quad & f_0(x) \\ s.t. \quad & a \leq x \leq b \end{aligned} \tag{205}$$

We can rewrite it into standard form:

$$\begin{aligned} \min_{x} \quad & f_0(x) \\ s.t. \quad & x - b \leq 0 \\ & a - x \leq 0 \end{aligned} \tag{206}$$

We can represent $f_1(x) = x - b$ and $f_2(x) = a - x$. We would first need to check if Slater's condition holds. If there exists a point y such that:

$$a < y < b \longrightarrow y = \frac{a+b}{2} \tag{207}$$

then Slater's condition holds. The first KKT condition is primal feasibility:

$$\begin{cases} f_1(x^*) = x^* - b \leq 0 \\ f_2(x^*) = a - x^* \leq 0 \end{cases} \tag{208}$$

The second KKT condition is dual feasibility:

$$\begin{cases} \lambda_1^* \geq 0 \\ \lambda_2^* \geq 0 \end{cases} \tag{209}$$

The third KKT condition is complementary slackness:

$$\begin{cases} \lambda_1^*(x^* - b) = 0 \\ \lambda_2^*(a - x^*) = 0 \end{cases} \tag{210}$$

The fourth KKT condition is stationarity:

$$\begin{aligned} \nabla f_1(x^*) = 1; \quad \nabla f_2(x^*) = -1 \\ \nabla f_0(x^*) + \lambda_1^*(1) + \lambda_2^*(-1) = 0 \end{aligned} \tag{211}$$

17.5.1. Optimality Conditions for Quadratic Programming

Consider the following Quadratic Programming (QP) problem:

$$\begin{aligned} \min_{x} \quad & x^\top P_0 x + x^\top Q x + r \\ s.t. \quad & Ax = b \end{aligned} \tag{212}$$

Slater's condition holds if there exists a point y such that:

$$Ay = b \tag{213}$$

and the KKT conditions are as follows:

1. **Primal Feasibility:** $Ax^* = b$
2. **Dual Feasibility:** (trivially satisfied since there are no inequality constraints)
3. **Complementary Slackness:** (trivially satisfied since there are no inequality constraints)
4. **Stationarity:** $2P_0x^* + q_0 + A^\top \mu^* = 0$

Therefore, we can check the conditions by solving the following matrix equation:

$$\begin{bmatrix} A & 0 \\ 2P_0 & A^\top \end{bmatrix} \begin{bmatrix} x^* \\ \mu^* \end{bmatrix} = \begin{bmatrix} b \\ -q_0 \end{bmatrix} \tag{214}$$

Karush-Kuhn-Tucker Conditions

Lecture 22

11/20/25

18. Karush-Kuhn-Tucker Conditions Conceptual Overview

The general form of a convex optimization problem is given by:

$$\begin{aligned} &\min_x f_0(x) \\ &s.t. f_i(x) \leq 0, \quad \forall i = 1, ..., k \\ &\quad h_j(x) = 0, \quad \forall j = 1, ..., m \end{aligned} \tag{215}$$

where f_0 is the objective function, f_i are the inequality constraint functions, and h_j are the equality constraint functions. What we can do is to get rid of the hard constraints by introducing Lagrange multipliers. The Lagrangian function is defined as:

$$L(x, \lambda, \mu) = f_0(x) + \sum_{i=1}^{k} \lambda_i f_i(x) + \sum_{j=1}^{m} \mu_j h_j(x) \tag{216}$$

where $\lambda = \begin{bmatrix} \lambda_1 & ... & \lambda_k \end{bmatrix}^\top$ and $\mu = \begin{bmatrix} \mu_1 & ... & \mu_m \end{bmatrix}^\top$ are the vectors of Lagrange multipliers associated with the inequality and equality constraints, respectively. This transforms the constrained optimization problem into an unconstrained one by incorporating the constraints into the objective function through the Lagrange multipliers.

Note that the function is affine in λ and μ for fixed x.

The dual function is defined as the infimum of the Lagrangian over x:

$$g(\lambda, \mu) = \inf_x L(x, \lambda, \mu) \tag{217}$$

18.1. Duality in Convex Optimization

The dual function provides a lower bound on the optimal value of the primal problem. Assume that p^* is the optimal value of the primal problem and d^* is the optimal value of the dual problem. We define the duality gap as:

$$\text{duality gap} = p^* - d^* \tag{218}$$

Weak duality states that for any feasible solution x of the primal problem and any feasible solution (λ, μ) of the dual problem, the following inequality holds:

$$f_0(x) \geq g(\lambda, \mu) \tag{219}$$

This implies that the duality gap is always non-negative:

$$p^* \geq d^* \tag{220}$$

Strong duality, on the other hand, the duality gap is zero:

$$p^* = d^* \tag{221}$$

This means that the optimal values of the primal and dual problems are equal.

Theorems:

- If x^* is an optimal solution to the primal problem and (λ^*, μ^*) is an optimal solution to the dual problem, and if strong duality holds, then KKT conditions are satisfied.
- Assume that the primal problem is a convex optimization problem and that Slater's condition holds. If (λ^*, μ^*) and x^* satisfy the KKT conditions, then x^* is an optimal solution to the primal problem and (λ^*, μ^*) is an optimal solution to the dual problem.
- The Dual Problem of a Linear Program is also a Linear Program.

18.2. Slater's Condition

Slater's condition is a sufficient condition for strong duality to hold in convex optimization problems. It states that if there exists a point y in the relative interior of the domain of the inequality constraint functions such that:

$$\begin{gathered} \begin{cases} f_i(y) \leq 0 & \text{if } f_i \text{ is affine} \\ f_i(y) < 0 & \text{if } f_i \text{ is non-affine} \end{cases} \quad \forall i = 1, ..., k \\ h_j(y) = 0, \quad \forall j = 1, ..., m \end{gathered} \tag{222}$$

then strong duality holds for the optimization problem.

18.3. Feasibility of Primal and Dual Problems

When solving a primal optimiztion problem, there are 3 possible outcomes:

1. The problem is unbounded below: $p^* = -\infty$.
2. The problem is infeasible: there is no x satisfying the constraints, $p^* = +\infty$.
3. The problem has an optimal solution: p^* is finite and attained at some x^*.

When solving a dual optimization problem, there are also 3 possible outcomes:

1. The problem is unbounded above: $d^* = +\infty$.
2. The problem is infeasible: there is no (λ, μ) satisfying the constraints, $d^* = -\infty$.
3. The problem has an optimal solution: d^* is finite and attained at some (λ^*, μ^*).

In the case of weak duality, if the primal problem is unbounded below, then the dual problem is infeasible and vice versa. A practical use is proving whether a set exists.

Consider the following cases:

$$\begin{aligned} \min \quad & 0 \\ s.t. \quad & f(x) \leq 0 \\ & h(x) = 0 \end{aligned} \tag{223}$$

By weak duality, the primal is infeasible if the dual problem is unbounded above. This means:

$$g(\lambda, \mu) = \inf_x L(x, \lambda, \mu) = +\infty \tag{224}$$

Example:
Prove that the following set is empty:

$$\mathcal{S} = \{x \in \mathbb{R}^3 \mid x_1^2 + x_2^2 + x_3^2 = 1, \quad x_1 + 2x_2 + 3x_3 \geq 5\} \tag{225}$$

We can formulate this as a primal optimization problem:

$$\begin{aligned} \min_x \quad & 0 \\ s.t. \quad & x_1^2 + x_2^2 + x_3^2 - 1 = 0 \\ & -x_1 - 2x_2 - 3x_3 + 5 \leq 0 \end{aligned} \tag{226}$$

The corresponding dual problem is:

$$\max_{\lambda \geq 0, \mu} g(\lambda, \mu) \tag{227}$$

where:

$$\begin{gathered} g(\lambda, \mu) = \inf_x L(x, \lambda, \mu) \\ L(x, \lambda, \mu) = 0 + \mu(x_1^2 + x_2^2 + x_3^2 - 1) + \lambda(-x_1 - 2x_2 - 3x_3 + 5) \end{gathered} \tag{228}$$

To find the infimum, we take the gradient of L with respect to x and set it to zero:

$$\nabla_x L = [2\mu x_1 - \lambda 2\mu x_2 - 2\lambda 2\mu x_3 - 3\lambda] = 0 \tag{229}$$

Solving for x, we get:

$$x_1 = \frac{\lambda}{2\mu}, \quad x_2 = \frac{\lambda}{\mu}, \quad x_3 = \frac{3\lambda}{2\mu} \tag{230}$$

Substituting these values back into the Lagrangian, we have:

$$\begin{aligned} L(x, \lambda, \mu) &= \mu\left(\left(\frac{\lambda}{2\mu}\right)^2 + \left(\frac{\lambda}{\mu}\right)^2 + \left(\frac{3\lambda}{2\mu}\right)^2 - 1\right) + \lambda\left(-\frac{\lambda}{2\mu} - 2\frac{\lambda}{\mu} - 3\frac{3\lambda}{2\mu} + 5\right) \\ &= \mu\left(\frac{\lambda^2}{4\mu^2} + \frac{\lambda^2}{\mu^2} + \frac{9\lambda^2}{4\mu^2} - 1\right) + \lambda\left(-\frac{8\lambda}{2\mu} + 5\right) \\ &= \mu\left(\frac{14\lambda^2}{4\mu^2} - 1\right) + \lambda\left(-\frac{8\lambda}{2\mu} + 5\right) \\ &= \frac{14\lambda^2}{4\mu} - \mu - \frac{8\lambda^2}{2\mu} + 5\lambda \\ &= -\frac{2\lambda^2}{4\mu} - \mu + 5\lambda \end{aligned} \tag{231}$$

Infeasibility Problem

Lecture 23

11/25/25

19. Duality

Consider the following primal optimization problem:

$$\begin{aligned} \min_{x} \quad & f_0(x) \\ s.t. \quad & f_i(x) \leq 0, \quad i = 1, ..., m \\ & h_j(x) = 0, \quad j = 1, ..., p \end{aligned} \tag{232}$$

and its corresponding dual problem:

$$\begin{aligned} \max_{\lambda, \mu} \quad & d(\lambda, \mu) \\ s.t. \quad & \lambda \geq 0 \end{aligned} \tag{233}$$

If we denote p^* and d^* as the optimal values of the primal and dual problems respectively, then the following properties hold:

1. $d(\lambda, \mu)$ is a concave function, even if the primal problem is not convex.
2. $d^* \leq p^*$

Sensitivity Analysis

Lecture 24

11/27/25

19.1. Constraint Elimination

Consider the following optimization problem:

$$\begin{aligned} \min_{x} \quad & f_0(x) \\ s.t. \quad & f_i(x) \leq 0, \quad i = 1, ..., m \\ & h_j(x) = 0, \quad j = 1, ..., p \end{aligned} \tag{234}$$

Let x^* be the optimal solution, then:

$$\begin{aligned} & f_i(x^*) = 0 \rightarrow \text{active/binding constriants} \\ & f_i(x^*) < 0 \rightarrow \text{inactive/non- binding constriants} \end{aligned} \tag{235}$$

If we remove the inactive constraints from the problem, the optimal solution x^* will remain unchanged. This process is called **constraint elimination**:

$$\begin{aligned} \min_x \quad & f_0(x) \\ s.t. \quad & f_i(x) \leq 0, \quad i \in A(x^*) \\ & h_j(x) = 0, \quad j = 1, ..., m \end{aligned} \tag{236}$$

If you do this, then x^* is also the optimal solution of the reduced problem.

If we assume Slater's condition holds for the problem, then we can use the KKT conditions to find the optimal solution x^*. Note that complementary slackness means:

$$\begin{gathered} \lambda_i^* f_i(x^*) = 0, \quad \forall i = 1, ..., m \\ \text{if } \begin{cases} f_i(x^*) = 0 \to \lambda_i^* = 0 \to f_i \text{ is not important} \\ f_i(x^*) < 0 \to \lambda_i^* = 0 \to f_i \text{ is not important} \end{cases} \end{gathered} \tag{237}$$

19.2. Sensitivity Analysis

Consider the following perturbed optimization problem:

$$\begin{aligned} \min_x \quad & f_0(x) \\ s.t. \quad & f_i(x) \leq u_i, \quad i = 1, ..., k \\ & h_j(x) = w_j, \quad j = 1, ..., m \end{aligned} \tag{238}$$

where $u \in \mathbb{R}^k$ and $w \in \mathbb{R}^m$ are perturbation parameters.

Theorem:

1. $p^*(u, w)$ is convex in (u, w)
2. If Slater's condition holds, then $p^*(u, w)$ is differentiable at $(0, 0)$ and the optimal Lagrange multipliers λ^* and μ^* corresponding to the original problem (i.e., when $u = 0$ and $w = 0$) satisfy:

$$\begin{aligned} \lambda_i^* &= -\frac{\partial p^*(0,0)}{\partial u_i}, \quad i = 1, ..., k \\ \mu_j^* &= -\frac{\partial p^*(0,0)}{\partial w_j}, \quad j = 1, ..., m \end{aligned} \tag{239}$$

This means that the optimal Lagrange multipliers can be interpreted as the sensitivity of the optimal value with respect to perturbations in the constraints.

The Taylor expansion of $p^*(u, w)$ around $(0, 0)$ is:

$$p^*(u, w) \approx p^*(0, 0) - \sum_{i=1}^{k} \lambda_i^* u_i - \sum_{j=1}^{m} \mu_j^* w_j \tag{240}$$

This shows that if $\lambda_i^*, \mu_i^* = 0$ for some i, then small perturbations in the corresponding constraints will not affect the optimal value to the first order. On the other hand, if λ_i^* or μ_j^* is large in magnitude, then small perturbations in the corresponding constraints can lead to significant changes in the optimal value.

So far, our variable x is a vector in $\mathbb{R}^n$. If we has a matrix as a variable, we can simply vectorize it:

$$\begin{aligned} &X \in \mathbb{R}^{m \times n} \\ \rightarrow &\tilde{f}_i(x) = f_i(X); i = 0, ..., k \\ \rightarrow &\tilde{h}_j(x) = h_j(X); j = 0, ..., m \end{aligned} \tag{241}$$

Exmaple:
Consider the following optimization problem:

$$\begin{aligned} \min_{X \in \mathbb{R}^{2\times 2}} \quad & \left\| \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} X - \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \right\|_2 \\ s.t. \quad & X_{1,1} + X_{1,2} \leq 5 \\ & X_{2,1} + 5X_{2,2} = 4 \end{aligned} \tag{242}$$

we can rewrite the optimization problem as:

$$\begin{aligned} \min_{x \in \mathbb{R}^4} \quad & \left\| \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} X - \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \right\|_2 \\ s.t. \quad & \overbrace{x_1 + x_3 - 5}^{\tilde{f}_i(x)} \leq 0 \\ & \underbrace{x_2 + 5x_4 - 4}_{\tilde{h}_i(x)} = 0 \end{aligned} \tag{243}$$

recall that in PCA, we had the optimization problem:

$$\begin{aligned} \max_{U \in \mathbb{R}^{n \times k}} \quad & \|X - Y\|_F \\ s.t. \quad & rank(X) \leq r \\ & X = CD^\top \end{aligned} \tag{244}$$

19.3. Matrix Completion Problem

Lets say we have a matrix $M \in \mathbb{R}^{m \times n}$ with some missing entries. We define the set of observed entries as:

$$\Omega \subseteq \{(i, j) \mid i = 1, ..., m; j = 1, ..., n\} \tag{245}$$

The matrix completion problem aims to fill in the missing entries of M based on the observed entries in Ω. One common approach is to assume that the underlying complete matrix has low rank. The optimization problem can be formulated as:

$$\begin{aligned} \min_{X \in \mathbb{R}^{m \times n}} \quad & rank(X) \\ s.t. \quad & X_{i,j} = X^*_{i,j}, \quad \forall (i, j) \in \Omega \end{aligned} \tag{246}$$

Where the rank is the cardinality of the set of non-zero singular values of X.

$$
\begin{aligned}
rank(X) &= \left\| \begin{bmatrix} \sigma_1(x) \\ \vdots \\ \sigma_p(x) \end{bmatrix} \right\|_0 \\
&= \sum_{i=1}^{p} \begin{cases} 1 & \text{if } \sigma_i(x) \neq 0 \\ 0 & \text{if } \sigma_i(x) = 0 \end{cases}
\end{aligned} \tag{247}
$$

However, this problem is non-convex and NP-hard. A common convex relaxation is to minimize the nuclear norm of X instead of its rank:

$$
\begin{aligned}
\min_{X \in \mathbb{R}^{m \times n}} \quad & \|X\|_* \\
s.t. \quad & X_{i,j} = X^*_{i,j}, \quad \forall (i,j) \in \Omega
\end{aligned} \tag{248}
$$

where the nuclear norm $\|X\|_*$ is defined as the sum of the singular values of X.

Matrix Optimization

Lecture 25

12/2/25

20. Matrix Optimization

In normal PCA, we try to find a low-rank approximation of a data matrix $X \in \mathbb{R}^{m \times n}$ by solving the optimization problem:

$$
\min_{rank(B) \leq k} \|X - B\|_F^2 \tag{249}
$$

where $B \in \mathbb{R}^{m \times n}$ is the low-rank approximation of X and k is the desired rank. However, in matrix optimization, we can have more complex objectives and constraints. Let $\gamma \in \mathbb{R}^{m \times n}$ be the low-rank approximation of X. We can formulate the matrix optimization problem as:

$$
\begin{aligned}
\min_{\gamma \in \mathbb{R}^{m \times n}} \quad & \|X\|_* + \lambda \|Z\|_F \\
s.t \quad & \gamma = X + Z
\end{aligned} \tag{250}
$$

where $\|X\|_*$ is the nuclear norm of X, which is the sum of its singular values, and $\lambda > 0$ is a regularization parameter that controls the trade-off between the nuclear norm and the Frobenius norm of Z.

Assume we have $Y = \begin{bmatrix} u^{(1)} & \dots & u^{(p)} \end{bmatrix} = X + Z$ where $X = \begin{bmatrix} u_b^{(1)} & \dots & u_b^{(p)} \end{bmatrix}$ and $Z = \begin{bmatrix} u_f^{(1)} & \dots & u_f^{(p)} \end{bmatrix}$. The optimization problem can be rewritten as:

$$
\min_{X,Z} \|X\|_* + \lambda \sum_i \sum_j |Z_{ij}| \tag{251}
$$

21. Numerical Algorithms

To solve the matrix optimization problem, we can use numerical algorithms such as the Alternating Direction Method of Multipliers (ADMM) or Proximal Gradient Descent. These algorithms iteratively update the variables X and Z to minimize the objective function while satisfying the constraints.

Lets say we can generate the iteratively sequence X^k, where k is finite, then in finite steps we can reach the optimal solution. A small issue with this is that we will rarely reach the exact optimal solution, but we can get very close to it. We define $\varepsilon > 0$ as the precision error tolerance, and we can stop the iterations when the change in the objective function is less than ε.

21.1. Descent Algorithms

Descent algorithms are a class of optimization algorithms that iteratively update the variables in the direction of the negative gradient of the objective function. The basic idea is to take small steps in the direction that reduces the objective function value. The update rule for descent algorithms can be expressed as:

$$x_{k+1} = x_k - S\Delta x_k \tag{252}$$

where $\alpha > 0$ is the step size, and Δx_k is the change in x at iteration k. For an algortihm to be considered a descent algorithm, it must satisfy the following conditions:

1. The objective function $f(x)$ is differentiable.
2. $f(x_{k+1}) < f(x_k)$

Theorem:
For alll small enough step size α

1. If $\nabla f(x_k)^\top \Delta x_k < 0$ then $f(x_{k+1}) < f(x_k)$
2. If $\nabla f(x_k)^\top \Delta x_k > 0$ then $f(x_{k+1}) > f(x_k)$
3. If $\nabla f(x_k)^\top \Delta x_k = 0$ then $f(x_k)$ is the solution

If $\nabla f(x_k) \neq 0$, then we can choose $\Delta x = -\nabla f(x_k)$. this means that $\nabla f(x_k)\Delta x_k = -\|\nabla f(x_k)\|^2 < 0$

These concepts lead to the Gradient Descent Algorithm, which is a specific type of descent algorithm where the update direction is the negative gradient of the objective function. The algorithm can be summarized as follows:

$$x_{k+1} = x_k - s_k \nabla f(x_k) \tag{253}$$

In gradient descent, we give

$$x_0 \quad s.t. \quad p = \{x \in \mathbb{R}^m \mid f(x) \leq f(x_0)\} \tag{254}$$

This is Lipschitz continuous gradient with constant $L > 0$ if $\forall x, y \in \mathbb{R}^m$ we have:

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \tag{255}$$

21.1.1. Newton's Method

Newton's Method is an optimization algorithm that uses second-order information (the Hessian matrix) to find the minimum of a function. The update rule for Newton's Method is given by:

$$x_{k+1} = x_k - s_k(\nabla^2 f(x_k))^{-1} \nabla f(x_k) \tag{256}$$

where s_k is the step size at each iteration k. Newton's Method typically converges faster than gradient descent, especially for functions that are well-approximated by a quadratic function near the minimum.

Lemma:
If f is twice differentiable and $\forall x \in \mathbb{R}^m$ and P is compact, then L exists.

Theorem:
Consider the gradient algorithm with an arbitrary precision error $\varepsilon > 0$, if the step size $s_0, s_1, \ldots$ are chosen such that $s_k \in \left(\varepsilon, \frac{2}{L}\right)$, then $\|\nabla f(x_k)\|$ converges to 0 as k goes to infinity.