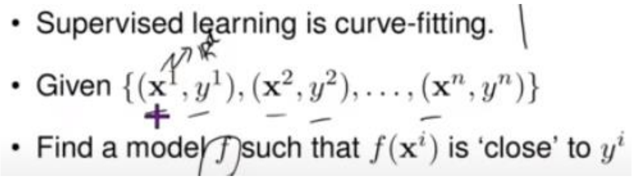
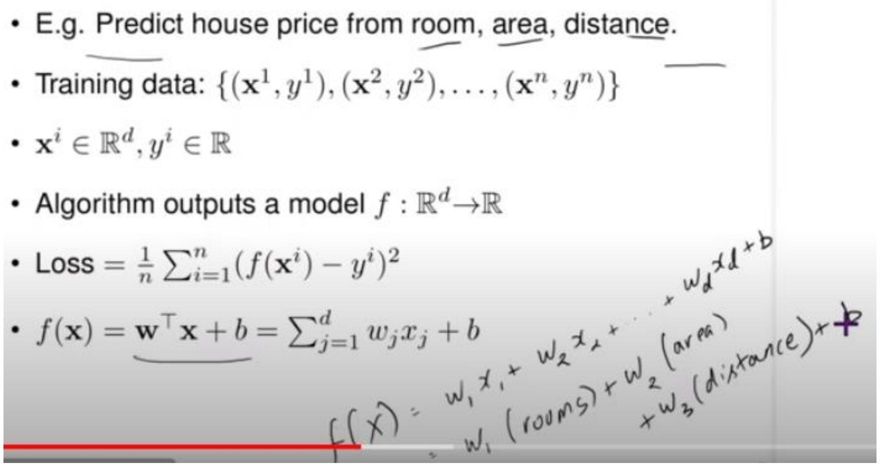
Week1

Supervised learning: Regression



- Supervised learning is curve-fitting.
- Given $\{(\mathbf{x}^1, y^1), (\mathbf{x}^2, y^2), \ldots, (\mathbf{x}^n, y^n)\}$
- Find a model f such that $f(\mathbf{x}^i)$ is 'close' to y^i



- E.g. Predict house price from room, area, distance.
- Training data: $\{(\mathbf{x}^1, y^1), (\mathbf{x}^2, y^2), \ldots, (\mathbf{x}^n, y^n)\}$
- $\mathbf{x}^i \in \mathbb{R}^d, y^i \in \mathbb{R}$
- Algorithm outputs a model $f : \mathbb{R}^d \to \mathbb{R}$
- Loss $= \frac{1}{n} \sum_{i=1}^{n} (f(\mathbf{x}^i) - y^i)^2$
- $f(\mathbf{x}) = \underline{\mathbf{w}^\top \mathbf{x} + b} = \sum_{j=1}^{d} w_j x_j + b$

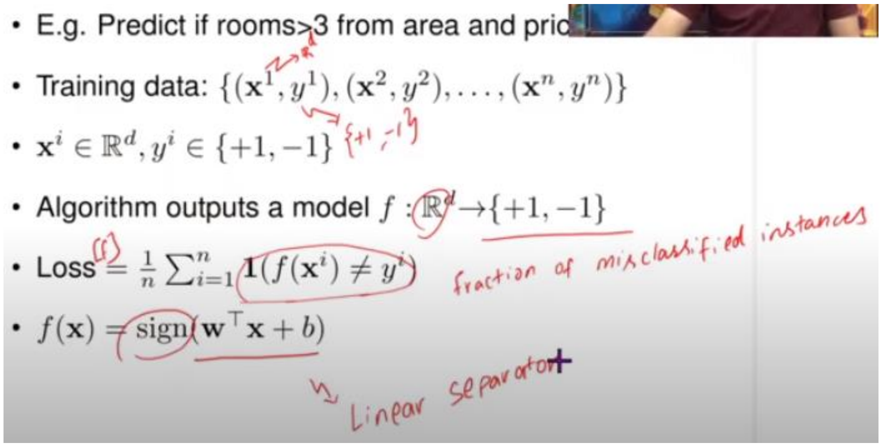
$f(x) = w_1 x_1 + w_2 x_2 + \ldots + w_d x_d + b$

$= w_1 (\text{rooms}) + w_2 (\text{area}) + w_3 (\text{distance}) + b$

The learning algorithm attempts to find the best parameters (w1, w2…wd and b) that yields the correct value of f(x) for any given x vector. In other words, it chooses the model that gives the least loss among many potential models. Choosing the potential models is typically a manual process.

The output of regression model is continuous and with any range.

Supervised Learning: Classification

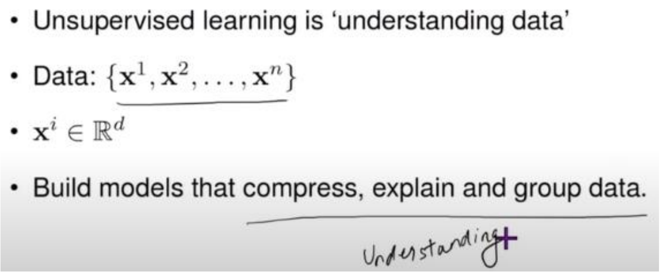


- E.g. Predict if rooms>3 from area and pric
- Training data: $\{(\mathbf{x}^1, y^1), (\mathbf{x}^2, y^2), \ldots, (\mathbf{x}^n, y^n)\}$
- $\mathbf{x}^i \in \mathbb{R}^d, y^i \in \{+1, -1\}$ {+1, -1}
- Algorithm outputs a model $f : \mathbb{R}^d \to \{+1, -1\}$
- Loss(f) $= \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(f(\mathbf{x}^i) \neq y^i)$ fraction of misclassified instances
- $f(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b)$

Linear separator

Evaluation shouldn’t be done on the training data, and instead on test data. Similarly, the selection of potential models that are input to the learning model, should be done using validation data.

Unsupervised learning

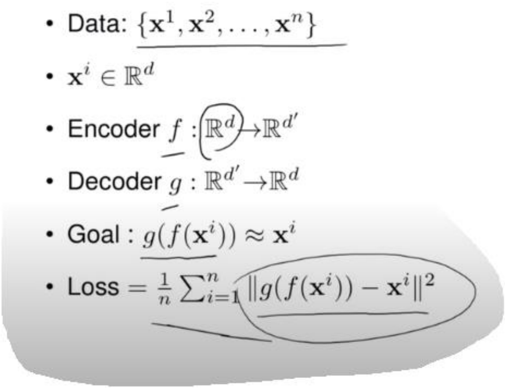


- Unsupervised learning is ‘understanding data’
- Data: $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n\}$
- $\mathbf{x}^i \in \mathbb{R}^d$
- Build models that compress, explain and group data.

Understanding

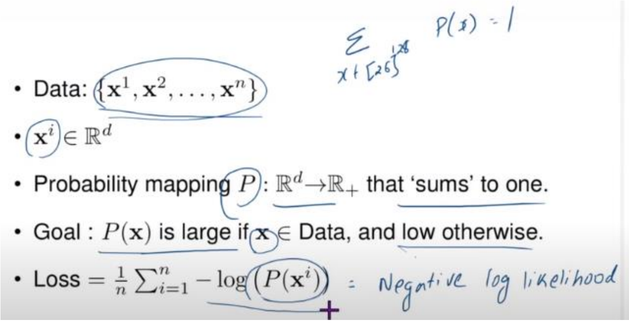
Typically, unsupervised learning is a pre-processing step, and the interpretation of the output is performed manually.

In ***dimensionality reduction*** technique, the goal is to compress and simplify data. Instead of using all of it, we choose part of it.



- Data: $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n\}$
- $\mathbf{x}^i \in \mathbb{R}^d$
- Encoder $f : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$
- Decoder $g : \mathbb{R}^{d'} \rightarrow \mathbb{R}^d$
- Goal : $g(f(\mathbf{x}^i)) \approx \mathbf{x}^i$
- Loss $= \frac{1}{n} \sum_{i=1}^{n} \|g(f(\mathbf{x}^i)) - \mathbf{x}^i\|^2$

In the ***density estimation*** technique, it assigns a probability value to each valid input, such that the sum of all of it is 1.



$\sum_{x \in [256]^{128}} P(x) = 1$

- Data: $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n\}$
- $\mathbf{x}^i \in \mathbb{R}^d$
- Probability mapping P: $\mathbb{R}^d \rightarrow \mathbb{R}_+$ that ‘sums’ to one.
- Goal : $P(\mathbf{x})$ is large if $\mathbf{x} \in$ Data, and low otherwise.
- Loss $= \frac{1}{n} \sum_{i=1}^{n} -\log(P(\mathbf{x}^i))$ = Negative log likelihood

The training set is used to fit the model, the validation set is used for model selection and the test set is used for computing the generalization error

Week-2

- Definition of an open ball B.

$$B(x,\epsilon) = \{ y \in \mathbb{R}^d : D(x,y) < \epsilon \}$$
$$\overline{B}(x,\epsilon) = \{ y \in \mathbb{R}^d : D(x,y) \leq \epsilon \}$$

  Note that D(x, y) indicates the distance between the points x and y, and is represented mathematically as follows.

$$D(x,y) = \|x-y\| = \sqrt{(x_1-y_1)^2 + \cdots + (x_d-y_d)^2}$$

- De-Morgan's laws of sets

$$(A \cup B)^c = A^c \cap B^c$$
$$(A \cap B)^c = A^c \cup B^c$$

- A sequence is said to converge, if

$$\lim_{i \to \infty} x_i = x^*$$
$$\Updownarrow$$
$$\forall\, \epsilon > 0,\ \exists\, N \text{ s.t. } x_n \in B(x^*, \epsilon)\ \forall\, n \geq N$$

  .

  The above two statements are equivalent, and the bottom one is a definition using the concept of an open ball centered at x* and having radius ξ

- Vector space is a set of vectors wherein linear combinations of any two of its vectors (u and v) is another vector in the same space. $\alpha u + \beta v \in V$, where α and β are two real numbers.
- Dot product of vectors is sum of product of its parts.

$$x \cdot y = x^T y = \sum_{i=1}^{d} x_i y_i$$

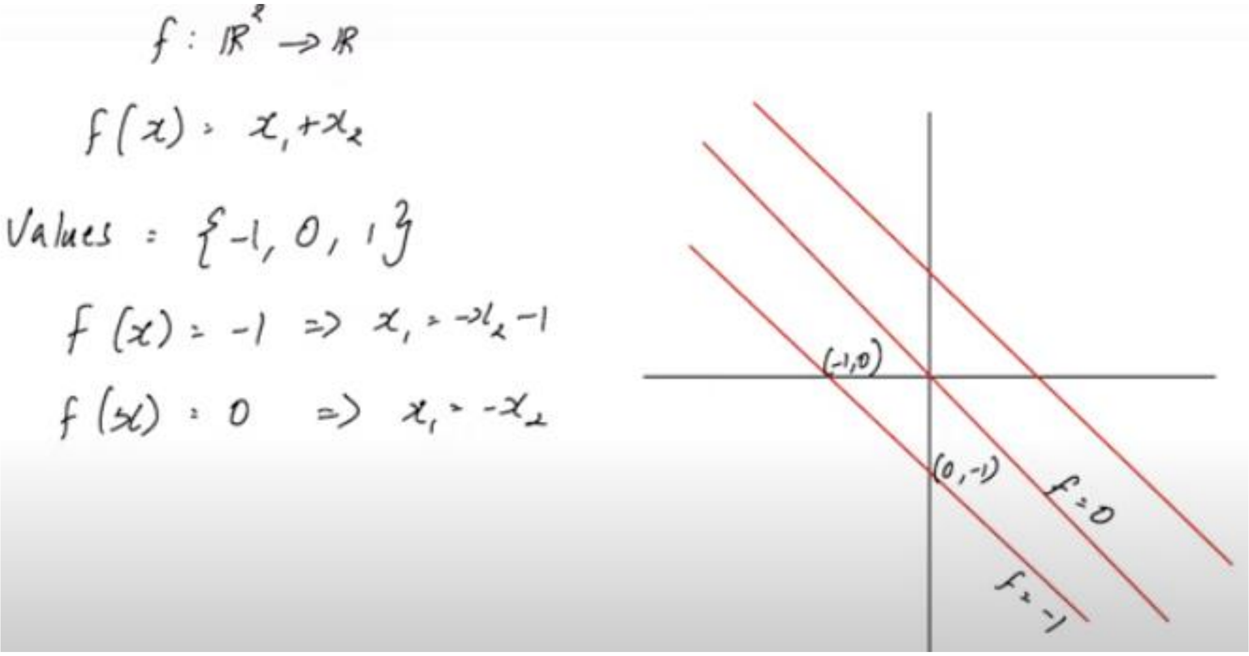
- Norm of a vector x is defined as

$$\|x\|^2 = x \cdot x = x^T x = \sum_{i=1}^{d} x_i^2$$

- Two vectors are said to be orthogonal(perpendicular), if the dot product is 0
- Graph of a function defined on a d-dimensional vector x_d is a (d+1)-dimensional vector and is mathematically represented as

$$G_f = \{ (x, f(x)) : x \in \mathbb{R}^d \}$$

- In order to plot two dimensional functions, use a range of values that the function can assume and draw contours for each value. Here's a contour plot for the function f(x) = $x_1 + x_2$



$f : \mathbb{R}^2 \to \mathbb{R}$

$f(x) = x_1 + x_2$

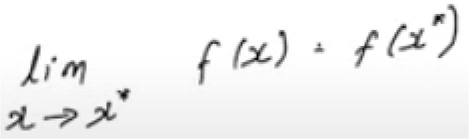
Values : $\{-1, 0, 1\}$

$f(x) = -1 \Rightarrow x_1 = -x_2 - 1$

$f(x) = 0 \Rightarrow x_1 = -x_2$

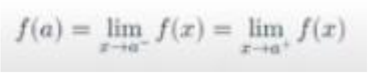
Univariate calculus

- Function is said to be continuous at x* if and only if



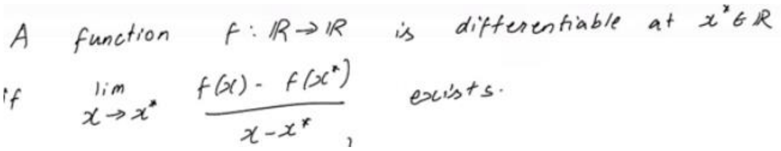
$$\lim_{x \to x^*} f(x) = f(x^*)$$

Can also be written as



$$f(a) = \lim_{x \to a^-} f(x) = \lim_{x \to a^+} f(x)$$

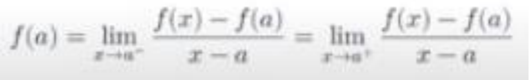
- Function is said to be continuous if it's continuous at all points in the domain.
- When is a function differentiable?



A function $f : \mathbb{R} \to \mathbb{R}$ is differentiable at $x^* \in \mathbb{R}$

if $\lim_{x \to x^*} \dfrac{f(x) - f(x^*)}{x - x^*}$, exists.

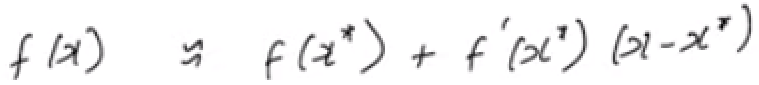
Also written as



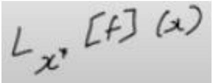
$$f(a) = \lim_{x \to a^-} \frac{f(x) - f(a)}{x - a} = \lim_{x \to a^+} \frac{f(x) - f(a)}{x - a}$$

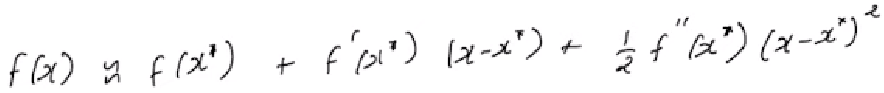
- A discontinuous function is not differentiable.
- Slope of a function is given by its derivative at a given point.

- Linear approximation of a function f at x* is given by the equation

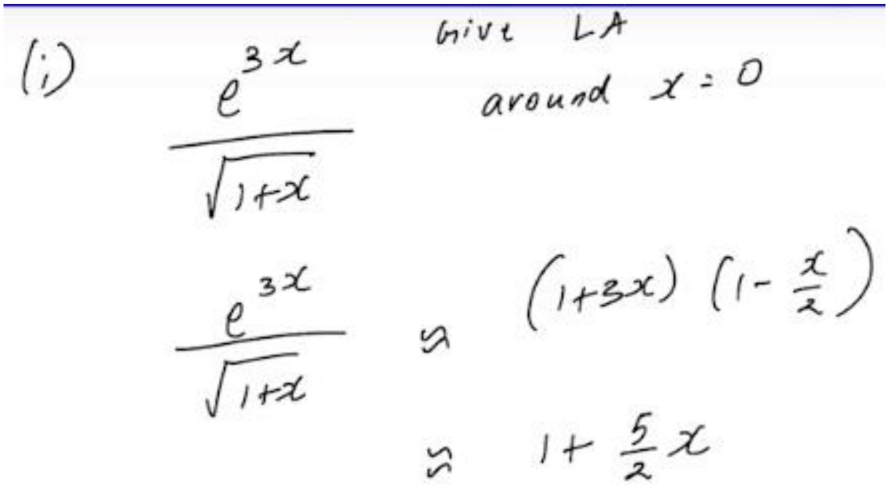
  $$f(x) \simeq f(x^*) + f'(x^*)(x-x^*)$$

  This is represented mathematically as  $L_{x^*}[f](x)$
- Typically, linear approximations are taken around x*=0, so f(x) = f(0) + f`(0) (x - 0) = f(0) + f`(0).x
- Quadratic approximation is given by the equation

  $$f(x) \simeq f(x^*) + f'(x^*)(x-x^*) + \frac{1}{2}f''(x^*)(x-x^*)^2$$

- Thus, to solve the following problem,

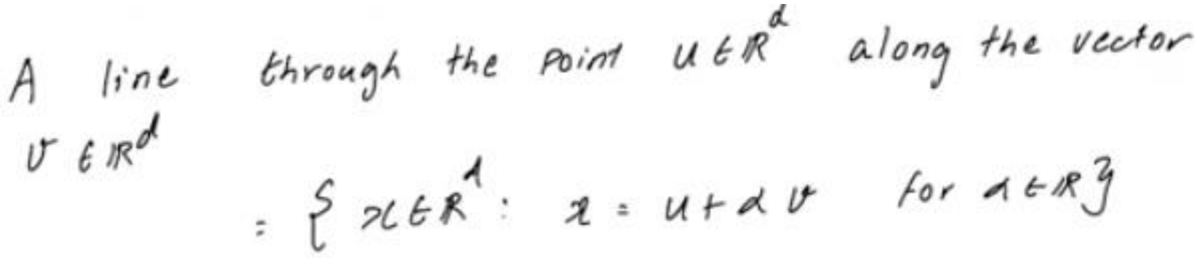
  (i) $\dfrac{e^{3x}}{\sqrt{1+x}}$ Give LA around $x=0$

  $$\frac{e^{3x}}{\sqrt{1+x}} \simeq (1+3x)\left(1-\frac{x}{2}\right)$$

  $$\simeq 1+\frac{5}{2}x$$

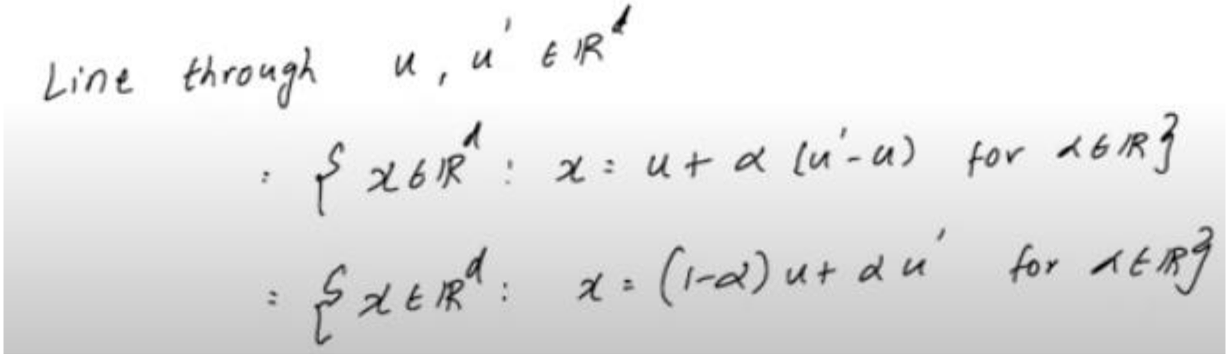
  we computed the linear approximation of e^3x and 1/sqrt(1+x) separately, multiply them and ignore the quadratic term.
- The points where the derivative of a function is zero are called critical points.

Multivariate calculus

- 

  A line through the Point $u \in \mathbb{R}^d$ along the vector $v \in \mathbb{R}^d$

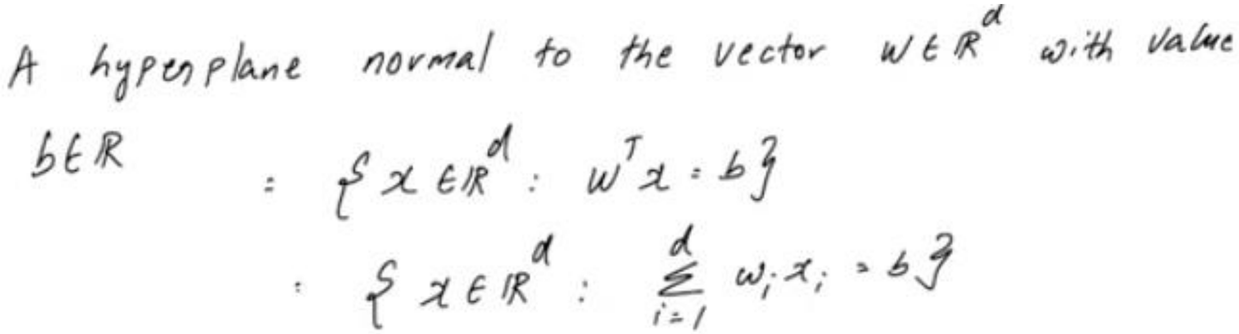
  $$: \{x \in \mathbb{R}^d : x = u + \alpha v \quad \text{for } \alpha \in \mathbb{R}\}$$

- 

  Line through $u, u' \in \mathbb{R}^d$

  $= \{x \in \mathbb{R}^d : x = u + \alpha(u' - u) \text{ for } \alpha \in \mathbb{R}\}$

  $= \{x \in \mathbb{R}^d : x = (1-\alpha)u + \alpha u' \text{ for } \alpha \in \mathbb{R}\}$

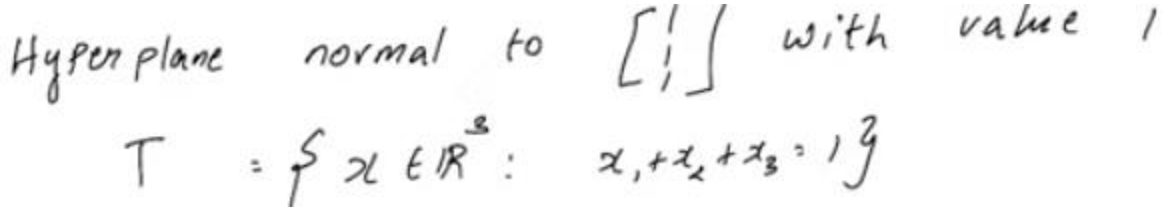
- 

  A hyperplane normal to the vector $w \in \mathbb{R}^d$ with value $b \in \mathbb{R}$

  $= \{x \in \mathbb{R}^d : w^T x = b\}$

  $= \{x \in \mathbb{R}^d : \sum_{i=1}^{d} w_i x_i = b\}$

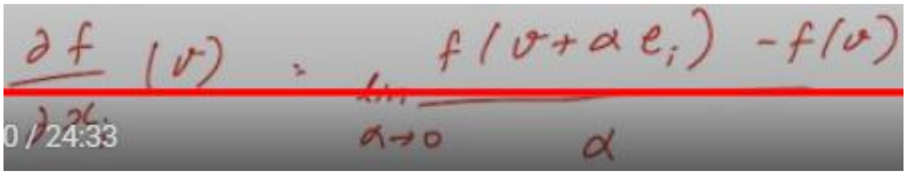
- As an example, consider this:

  Hyperplane normal to $\begin{bmatrix}1\\1\\1\end{bmatrix}$ with value 1

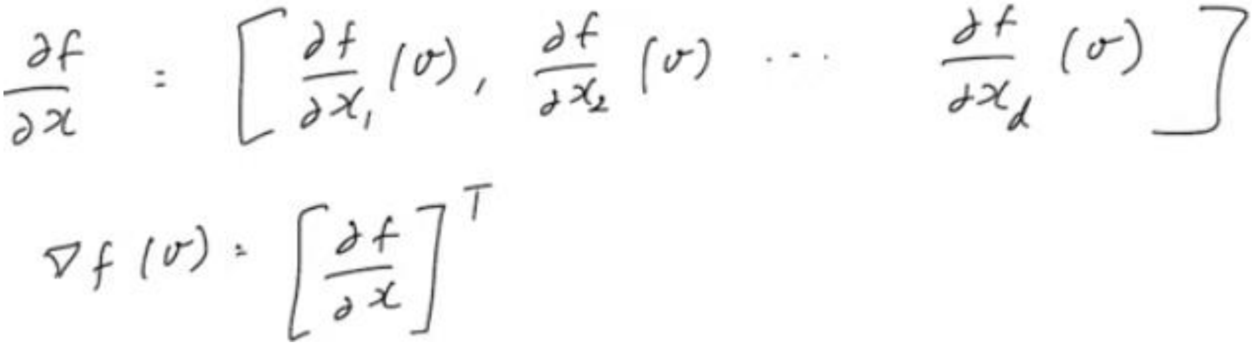
  $T = \{x \in \mathbb{R}^3 : x_1 + x_2 + x_3 = 1\}$

  Here, w is the vector all of whose components x_1, x_2 and x_3 are 1. Applying the previous mathematical definition of hyperplane normal to the vector w, we get $x_1 + x_2 + x_3 = 1$
- Partial derivative of function f evaluated at the point v is defined as

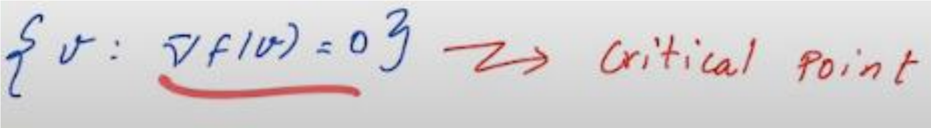
  $$\frac{\partial f}{\partial x_i}(v) = \lim_{\alpha \to 0} \frac{f(v + \alpha e_i) - f(v)}{\alpha}$$

- Gradient of a function evaluated at v is defined as a column vector containing Its partial derivatives based on each component of x.

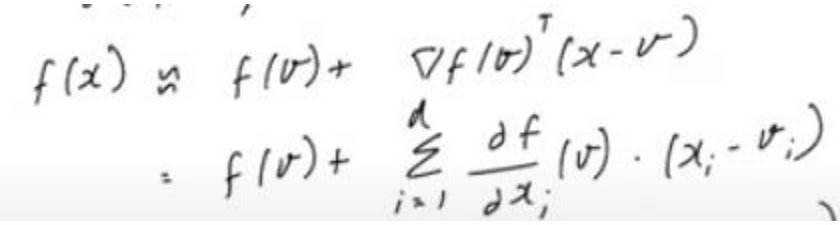
  $$\frac{\partial f}{\partial x} = \left[\frac{\partial f}{\partial x_1}(v), \frac{\partial f}{\partial x_2}(v) \cdots \frac{\partial f}{\partial x_d}(v)\right]$$

  $$\nabla f(v) = \left[\frac{\partial f}{\partial x}\right]^T$$
- The points where the gradient of a function is a zero vector are called critical points.



$$\{v : \underline{\nabla f(v) = 0}\} \Rightarrow \text{Critical point}$$

- For a multi-variable function f (dimension d), linear approximation is given as

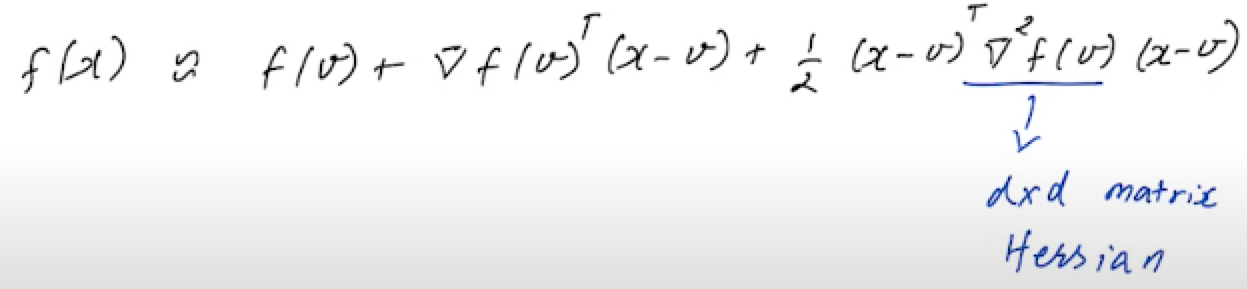


$$f(x) \approx f(v) + \nabla f(v)^T (x - v)$$

$$= f(v) + \sum_{i=1}^{d} \frac{\partial f}{\partial x_i}(v) \cdot (x_i - v_i)$$

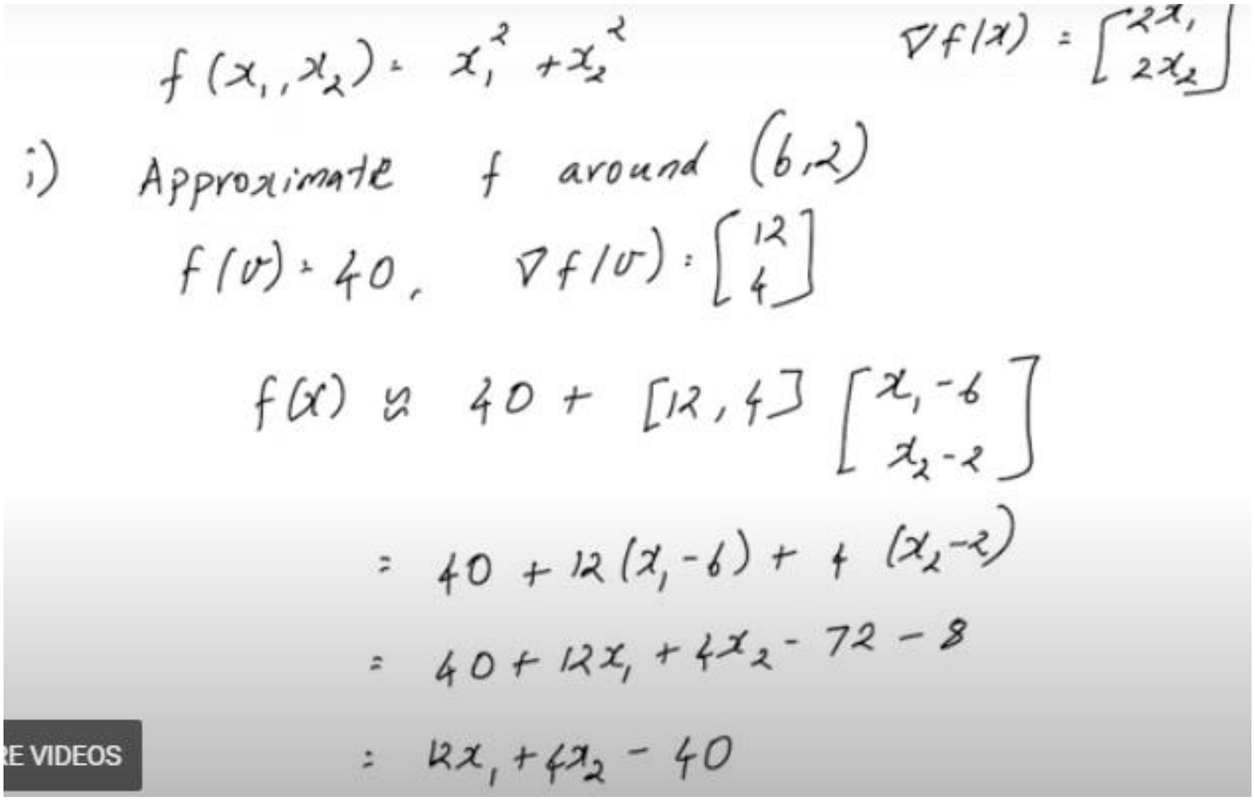
where x and v are vectors in Rd and are approximately equal to each other.

- Higher order approximation is given by



$$f(x) \approx f(v) + \nabla f(v)^T (x - v) + \frac{1}{2} (x - v)^T \underbrace{\nabla^2 f(v)}_{\substack{d \times d \text{ matrix} \\ \text{Hessian}}} (x - v)$$

- The gradient of a function f (denoted as ∇f) is a collection of all its partial derivatives (on each dimension) into a vector.
- A worked-out example of linear approximation of a 2-dimension function



$$f(x_1, x_2) = x_1^2 + x_2^2 \qquad \nabla f(x) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix}$$

i) Approximate f around $(6,2)$

$$f(v) = 40, \quad \nabla f(v) = \begin{bmatrix} 12 \\ 4 \end{bmatrix}$$

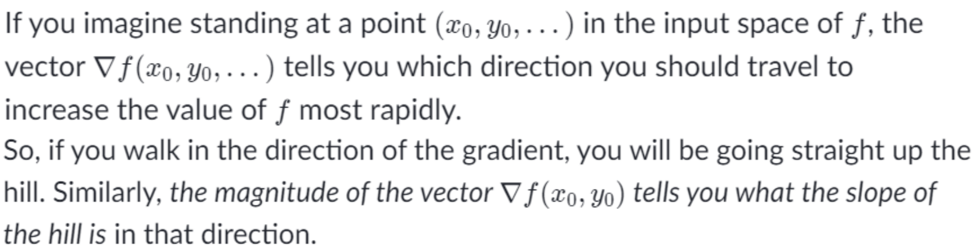
$$f(x) \approx 40 + [12, 4] \begin{bmatrix} x_1 - 6 \\ x_2 - 2 \end{bmatrix}$$

$$= 40 + 12(x_1 - 6) + 4(x_2 - 2)$$

$$= 40 + 12x_1 + 4x_2 - 72 - 8$$

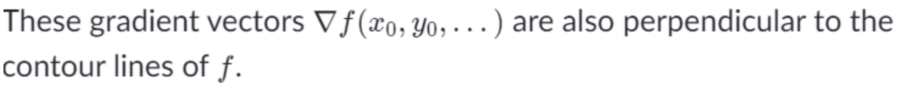
$$= 12x_1 + 4x_2 - 40$$

- Gradient is perpendicular to the function’s contour plot at a specific point.
- The most important thing to remember about the gradient: The gradient of *f*, if evaluated at an input (x_0, y_0) points in the direction of the steepest ascent. When the function *f* accepts more than two inputs, the interpretation of a gradient is similar.



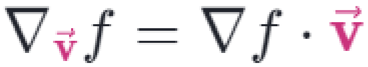
If you imagine standing at a point $(x_0, y_0, \dots)$ in the input space of f, the vector $\nabla f(x_0, y_0, \dots)$ tells you which direction you should travel to increase the value of f most rapidly.

So, if you walk in the direction of the gradient, you will be going straight up the hill. Similarly, *the magnitude of the vector* $\nabla f(x_0, y_0)$ *tells you what the slope of the hill is* in that direction.



These gradient vectors $\nabla f(x_0, y_0, \dots)$ are also perpendicular to the contour lines of f.

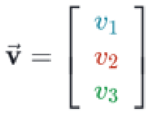
- For a function f that varies with *x, y and z*, the directional derivative along the **unit** vector *v* is



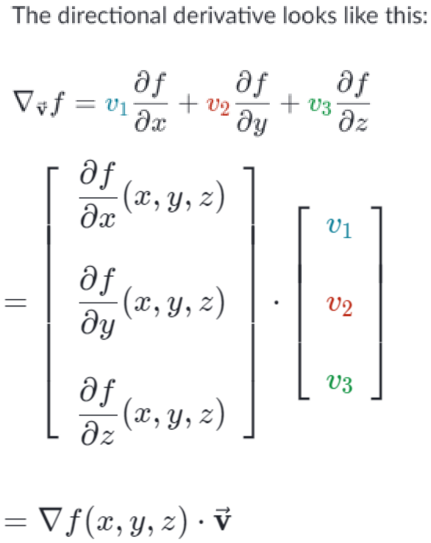
$$\nabla_{\vec{\mathbf{v}}} f = \nabla f \cdot \vec{\mathbf{v}}$$

More details on this follows:

If v is denoted by the vector,



$$\vec{\mathbf{v}} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$



The directional derivative looks like this:

$$\nabla_{\vec{\mathbf{v}}} f = v_1 \frac{\partial f}{\partial x} + v_2 \frac{\partial f}{\partial y} + v_3 \frac{\partial f}{\partial z}$$

$$= \begin{bmatrix} \frac{\partial f}{\partial x}(x, y, z) \\ \frac{\partial f}{\partial y}(x, y, z) \\ \frac{\partial f}{\partial z}(x, y, z) \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

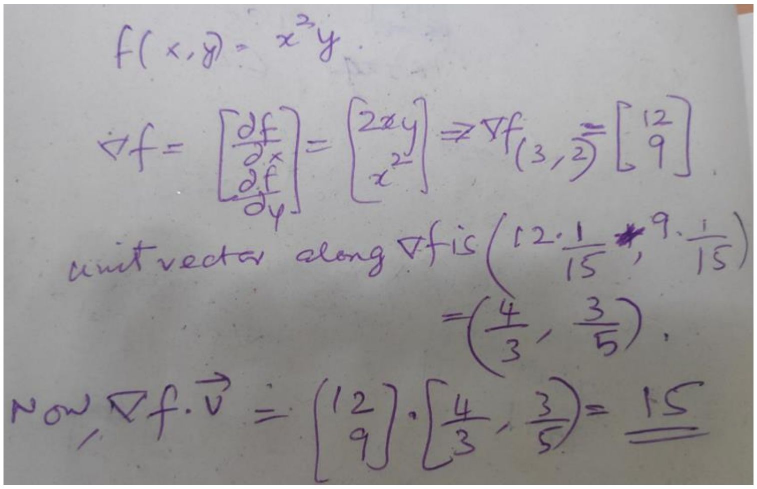
$$= \nabla f(x, y, z) \cdot \vec{\mathbf{v}}$$

NOTE: If the vector *v* is not a unit vector, normalize it before applying the above method, or divide the above expression by magnitude of vector *v*.

- In order to find the maximal directional derivative, find the unit vector along the gradient. Then apply the same formula

$$\nabla_{\vec{\mathbf{v}}} f = \nabla f \cdot \vec{\mathbf{v}}$$

For example, to find the maximal derivative of f(x,y) = x^{2}y at (3,2), refer to the following working.



$f(x,y) = x^2y$.

$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} 2xy \\ x^2 \end{bmatrix} \Rightarrow \nabla f_{(3,2)} = \begin{bmatrix} 12 \\ 9 \end{bmatrix}$.

unit vector along ∇f is $\left(12 \cdot \frac{1}{15}, 9 \cdot \frac{1}{15}\right)$

$= \left(\frac{4}{3}, \frac{3}{5}\right)$.

Now, $\nabla f \cdot \vec{v} = \begin{bmatrix} 12 \\ 9 \end{bmatrix} \cdot \left[\frac{4}{3}, \frac{3}{5}\right) = \underline{\underline{15}}$

- Maximum derivative can also be computed by finding the magnitude of the gradient vector.
- Cauchy-Shwarz inequality states that given 2 d-dimensional vectors a and b,

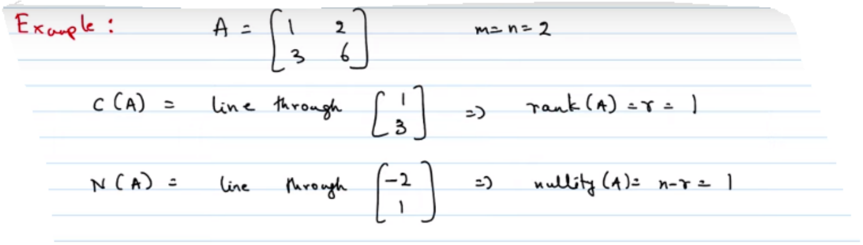
$$-\|a\| \cdot \|b\| \leq a^T b \leq \|a\| \, \|b\|$$

Note that the equality holds when vector a is a scalar multiple of vector b.

Week-3

Linear Algebra

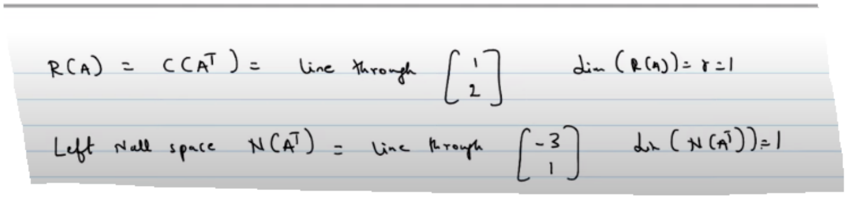
- C(A) = Vector space spanned by columns of given matrix. This is also called rank of the matrix.
- R(A) = C(A^T) = Vector space spanned by rows of given matrix (or by columns of the transpose of the matrix)
- N(A) = basis for the solution set of a homogeneous linear system derived from the given matrix.
- Left null space = basis for the solution set of a homogeneous linear system derived from the transpose of the given matrix.



Example : $A = \begin{bmatrix} 1 & 2 \\ 3 & 6 \end{bmatrix}$ $m = n = 2$

$C(A)$ = line through $\begin{bmatrix} 1 \\ 3 \end{bmatrix}$ $\Rightarrow$ rank$(A) = r = 1$

$N(A)$ = line through $\begin{bmatrix} -2 \\ 1 \end{bmatrix}$ $\Rightarrow$ nullity$(A) = n - r = 1$



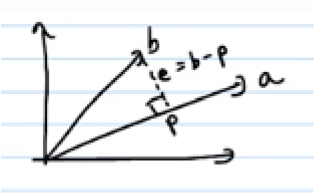
$R(A) = C(A^T)$ = line through $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ dim$(R(A)) = r = 1$

Left Null space $N(A^T)$ = line through $\begin{bmatrix} -3 \\ 1 \end{bmatrix}$ dim$(N(A^T)) = 1$

- Null space of a matrix is a vector space, which implies that all linear combinations of its basis also belong to the null space.
- If the matrix is invertible, the column vectors are linearly independent. In this case, the null space only has 'zero' vector, and column space is the whole space.
- ☑ $\dim(C(A)) + \dim(N(A)) = n$
- ☑ $\dim(C(A^T)) + \dim(N(A^T)) = m$
- ☑ $\text{rank}(A) + \text{nullity}(A) = n$
- Rank + Nullity = number of columns
- Rank + Left nullity = number of rows
- Column rank = Row rank = Rank of the matrix.
- A set consisting of mutually orthogonal vectors is a linearly independent set.
- Two subspaces are orthogonal to each other, when each element in first subspace is orthogonal to each element in the second subspace. For example, row subspace of a given matrix is orthogonal to its null subspace. Similarly, column subspace of a given matrix is orthogonal to the left null subspace of its transpose.
- If {v1, v2…vk} are mutually orthogonal (non-trivial) set of vectors, it's a linearly independent set.
- Inverse of the transpose of matrix A = Transpose of the inverse of matrix A (https://www.youtube.com/watch?v=MsIvs_6vC38)

Projections

- When vector b is not in column space of A, we typically project b into the column space of A.
- Projecting a vector b onto vector a.



Projection matrix : Recall $p = \left(\frac{a^T b}{a^T a}\right) a = \left(\frac{a a^T}{a^T a}\right) b$

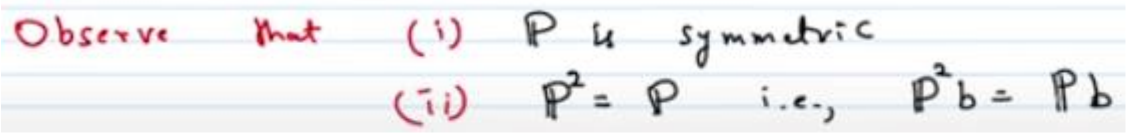
Let $P = \frac{a a^T}{a^T a}$. Then, projection of b onto a is Pb

For example,

Example: $a = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

Projection matrix is $P = \frac{a a^T}{a^T a} = \frac{1}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}.$

Following are a few observations, while projecting.



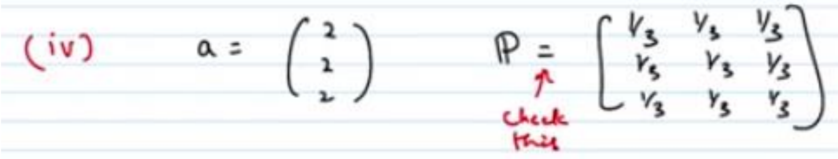
Observe that (i) $\mathbb{P}$ is symmetric

(ii) $\mathbb{P}^2 = \mathbb{P}$ i.e., $\mathbb{P}^2 b = \mathbb{P} b$

The second observation follows from the fact that if you project twice, it's not going to alter the projection matrix, and hence multiplies by itself.

Third observation is that column space of the projection matrix is a line passing through the vector a. Null space of the projection matrix is a plane orthogonal to vector a.

Further, the projection matrix will not change if vector a is a scalar multiple of itself. Thus,

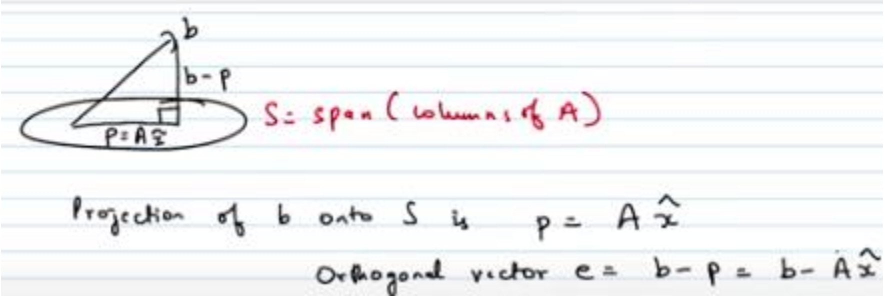


(iv) $a = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}$ $\qquad \underset{\substack{\uparrow \\ \text{check} \\ \text{this}}}{\mathbb{P}} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$

Least square error

- How to find the least square error?

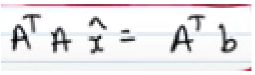
  The following diagram shows vector b projected onto the column space of A. Projection is denoted by $p = A\hat{x}$ (p = Ax_{HAT}). E denotes the error and can be represented as vector (b–p)



Projection of b onto S is $p = A\hat{x}$

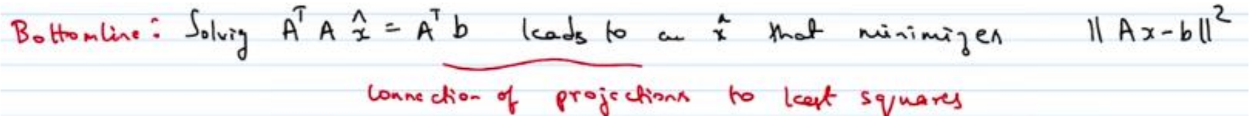
Orthogonal vector $e = b - p = b - A\hat{x}$

Calculations leads to the following equation, from which we can compute the projection.



$$A^T A \hat{x} = A^T b$$

- To find the least square error of $(Ax – b)^2$ is to find the projection of vector b into column space of matrix A.



Bottomline: Solving $A^T A \hat{x} = A^T b$ leads to an $\hat{x}$ that minimizes $\|Ax - b\|^2$

Connection of projections to least squares

Case 1: When columns of A are linearly independent, Solving $A^TA\hat{x} = A^Tb$ when (A^TA) is invertible

$$\boxed{\hat{x} = (A^TA)^{-1}A^Tb}$$

Projection $P = A\hat{x} = A(A^TA)^{-1}A^Tb$

Case 2: If vector b belongs to the column space of A, projection matrix is identity matrix and the projection is same as vector b itself.

$b \in C(A)$ ie., $b = Ax$

$$p = A(A^TA)^{-1}A^Tb = A\underbrace{(A^TA)^{-1}A^TA}_{=I}x = Ax = b$$

Case 3: If vector b is in the null space of transpose of A, projection is 0.

$b \in N(A^T)$

$p = A(A^TA)^{-1}A^Tb = 0$ since $A^Tb = 0$

Case 4: If matrix A is invertible, projection is vector b itself.

A is square & invertible $\Leftrightarrow$ $C(A) = \mathbb{R}^n$

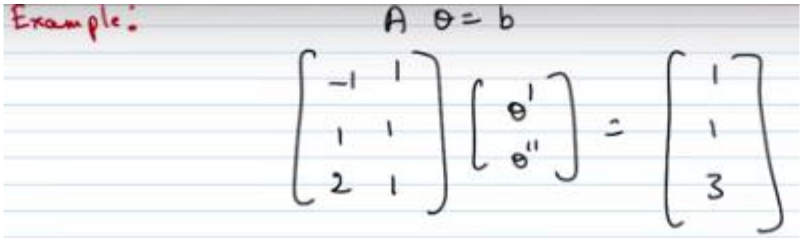
$$p = A(A^TA)^{-1}A^Tb = AA^{-1}(A^T)^{-1}A^Tb = b$$

Case 5: If matrix is of rank 1, projection is the same as with that on a line.

A is rank one i.e., $A = \begin{bmatrix} | \\ a \\ | \end{bmatrix}$

Then, $\hat{x} = \dfrac{a^Tb}{a^Ta}$

- Projection matrix P is symmetric $\mathbb{P}^T = \mathbb{P}$, and satisfies $\mathbb{P}^2 = \mathbb{P}$. Converse is also true.
- Example of projection and least squares method applied.



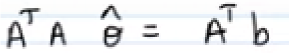
Example:

$$A\ \theta = b$$

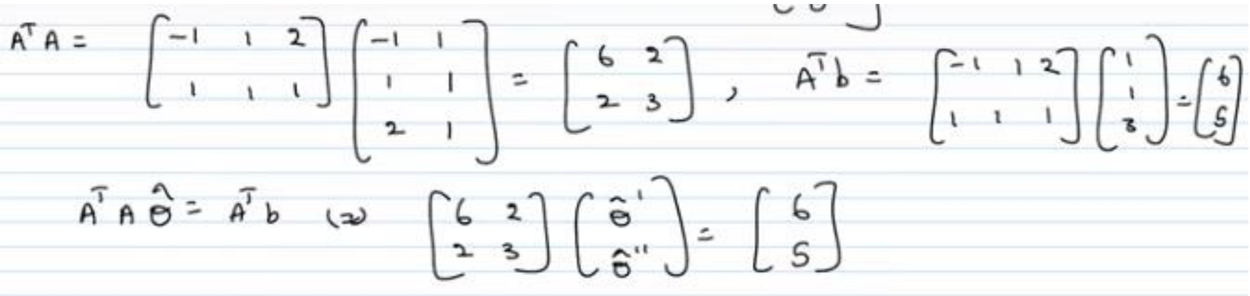
$$\begin{bmatrix} -1 & 1 \\ 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} \theta' \\ \theta'' \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix}$$

This system of linear equations is not solvable, since it’s inconsistent (can be proved by row-reducing).

Let’s apply the equation above



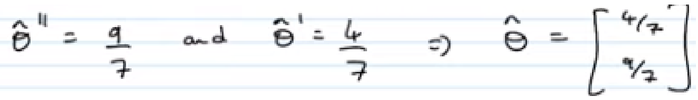
$$A^T A\ \hat{\theta} = A^T b$$



$$A^T A = \begin{bmatrix} -1 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & 1 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 2 \\ 2 & 3 \end{bmatrix}, \quad A^T b = \begin{bmatrix} -1 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \end{bmatrix}$$

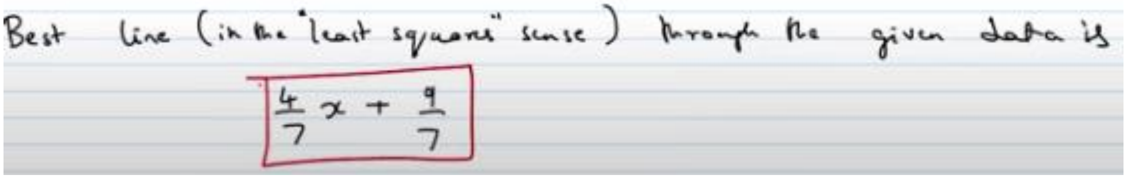
$$A^T A \hat{\theta} = A^T b \iff \begin{bmatrix} 6 & 2 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} \hat{\theta}' \\ \hat{\theta}'' \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \end{bmatrix}$$

Solving the above set of linear equations, we get



$$\hat{\theta}'' = \frac{9}{7} \text{ and } \hat{\theta}' = \frac{4}{7} \Rightarrow \hat{\theta} = \begin{bmatrix} 4/7 \\ 9/7 \end{bmatrix}$$

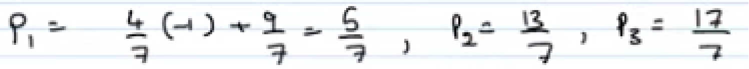
Thus,



Best line (in the "least squares" sense) through the given data is

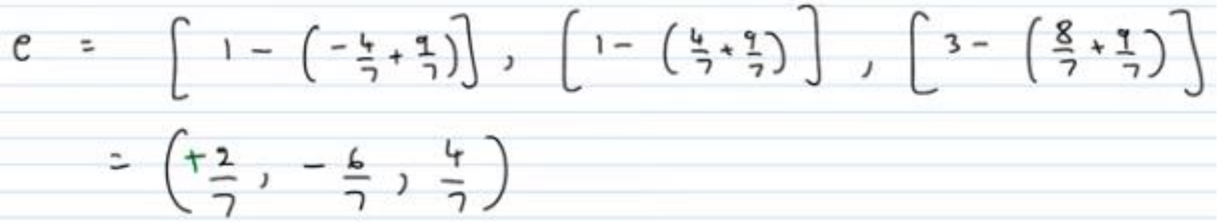
$$\boxed{\frac{4}{7}x + \frac{9}{7}}$$

Using the above equation for the line on each dimension of the input vector ([-1,1,2]), we get



$$p_1 = \frac{4}{7}(-1) + \frac{9}{7} = \frac{5}{7}, \quad p_2 = \frac{13}{7}, \quad p_3 = \frac{17}{7}$$

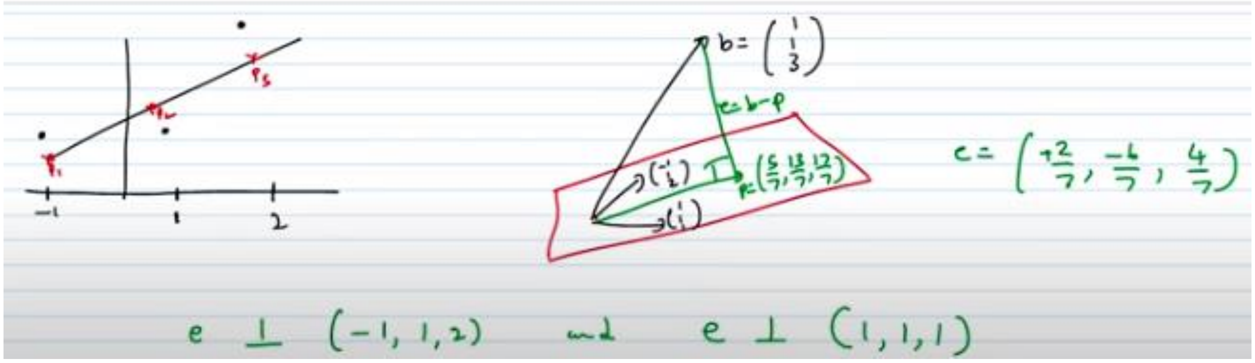
Given b = [1,1,3]



$$e = \left[1 - \left(-\frac{4}{7} + \frac{9}{7}\right)\right], \left[1 - \left(\frac{4}{7} + \frac{9}{7}\right)\right], \left[3 - \left(\frac{8}{7} + \frac{9}{7}\right)\right]$$

$$= \left(+\frac{2}{7}, -\frac{6}{7}, \frac{4}{7}\right)$$

Rough sketch of these points on the X-Y plane looks like this. From this sketch, it's clear that e is orthogonal to both input vectors.



p_1
p_2
p_3
-1
1
2
$b = \begin{pmatrix}1\\1\\3\end{pmatrix}$
$e = b - p$
$\begin{pmatrix}-1\\1\\2\end{pmatrix}$
$\begin{pmatrix}1\\1\\1\end{pmatrix}$
$p = \left(\frac{5}{7}, \frac{13}{7}, \frac{12}{7}\right)$
$e = \left(\frac{+2}{7}, \frac{-6}{7}, \frac{4}{7}\right)$
$e \perp (-1,1,2)$ and $e \perp (1,1,1)$

NOTE: Least square error is well covered in these Khan Academy videos:

Least squares approximation | Linear Algebra | Khan Academy - YouTube

Least squares examples | Alternate coordinate systems (bases) | Linear Algebra | Khan Academy - YouTube

Week4

- For Aθ = Y, loss function is represented by the following equation.

$$L(\theta) = \frac{1}{2}(A\theta - Y)^T (A\theta - Y)$$

and the least squares is given by

$$(A^T A)\theta = A^T Y$$

We can solve $\theta = (A^T A)^{-1} A^T Y$, if A is a full rank matrix.

Minimizing the loss is equivalent to maximizing the likelihood function.

$$\log L(\theta) = \frac{n}{2}\log\beta - \frac{n}{2}\log 2\pi - \beta\left[\frac{1}{2}\sum(y_i - \theta^T x_i)^2\right]$$

In the case of polynomial regression, solution can be obtained by performing linear regression on

$(A^T A)\theta = A^T Y$, where A is a matrix with transformed features $A = \begin{bmatrix} \phi(x_1)^T \\ \vdots \\ \phi(x_n)^T \end{bmatrix}$ and $\hat{y}(x) = \theta^T \phi(x)$.

Solution also can be represented in a regularized form as:

$$(A^T A + \lambda I)\,\theta_{reg} = A^T Y$$

NOTE: This method is called ridge regression.

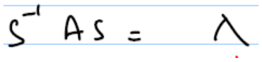
- For $\lambda > 0$, $(A^T A + \lambda I)$ is always invertible

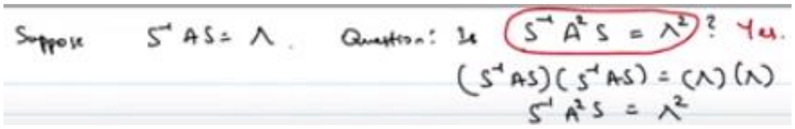
Eigen values/Eigen-vectors

- For a matrix A, eigen-value λ can be obtained by equating determinant of (A–λ*I*) to 0, and the corresponding eigenvector lies in the null space of (A–λ*I*).
- If an eigenvalue of a matrix A is zero, then its corresponding eigen-vector belongs to the null space of A.
- For a real symmetric matrix, all its eigen-values are real, but there could be imaginary eigen-vectors.
- If matrix A results in distinct eigen values, it'll have linearly independent eigen vectors.
- If A = a*I* + B where A and B are matrices and *I* is the identity matrix, both A and B will have same eigen-vectors.
- Eigen-value corresponding to every non-zero vector in the column-space of projection matrix is 1.
- Eigen-value corresponding to every non-zero vector orthogonal to the column-space of projection matrix is 0.
- A permutation matrix has eigen-value(s) 1 or -1.
- If A has r non-zero eigenvalues, then rank of A is at least r.
- If x is an eigenvector of A corresponding to eigenvalue λ1 and x is also an eigenvector of B corresponding to eigenvalue λ2, then x is also an eigenvector of (A+B).
- Some important properties of eigen-values and eigen-vectors:
    - Sum of the eigen-values of a matrix is equal to the sum of its diagonal elements (trace).
    - The product of the eigenvalues of a matrix equals the determinant of the matrix.
    - Matrix is singular (zero determinant) if and only if it has 0 eigen-value.
    - The eigenvalues of an upper (or lower) triangular matrix are the elements on the main diagonal.
    - If λ is an eigenvalue of **A**, then λ is an eigenvalue of $\mathbf{A}^T$.
    - If λ is an eigenvalue of **A**, then λ^k is an eigenvalue of $\mathbf{A}^k$, for any positive integer k.

  - If λ is an eigenvalue of **A** and if **A** is invertible, then 1/λ is an eigenvalue of **A**$^{-1}$ (follows from above statement)
  - If λ is an eigenvalue of **A**, then αλ is an eigenvalue of α**A**, where α is any arbitrary scalar.
  - If **x** is an eigenvector of **A** corresponding to the eigenvalue λ, then **x** is an eigenvector of α**A** corresponding to eigenvalue αλ.
  - If **x** is an eigenvector of **A** corresponding to the eigenvalue λ, then **x** is an eigenvector of **A**^{k} corresponding to the eigenvalue λ^k, for any positive integer *k*
- For more detailed discussion refer to https://www.adelaide.edu.au/mathslearning/system/files/media/documents/2020-03/evalue-magic-tricks-handout.pdf.
- For details on eigen-values and eigen-vectors refer to https://www.khanacademy.org/math/linear-algebra/alternate-bases/eigen-everything/v/linear-algebra-introduction-to-eigenvalues-and-eigenvectors
  or
  https://ocw.mit.edu/courses/mathematics/18-06-linear-algebra-spring-2010/video-lectures/lecture-21-eigenvalues-and-eigenvectors/

Matrix Diagonalizability

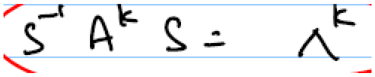
- A matrix is diagonalizable if there exists matrix S such that  $S^{-1}AS = \Lambda$, where λ represents a diagonal matrix with eigen-values along its diagonal. Each column of matrix S is an eigen vector of the matrix A.
- Matrix A is diagonalizable only when there are enough eigen vectors. If the matrix has repeated eigen-values, then it's not diagonalizable.
- If the matrix is not diagonalizable, it can be added with a scalar multiple of identify matrix (λ*I*) to make it diagonalizable.
- S can be used to diagonalize square of the matrix A. Proof below.



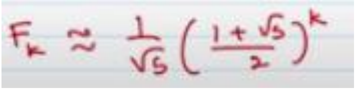
Suppose $S^{-1}AS = \Lambda$. Question: Is $\boxed{S^{-1}A^2S = \Lambda^2}$? Yes.

$(S^{-1}AS)(S^{-1}AS) = (\Lambda)(\Lambda)$

$S^{-1}A^2S = \Lambda^2$

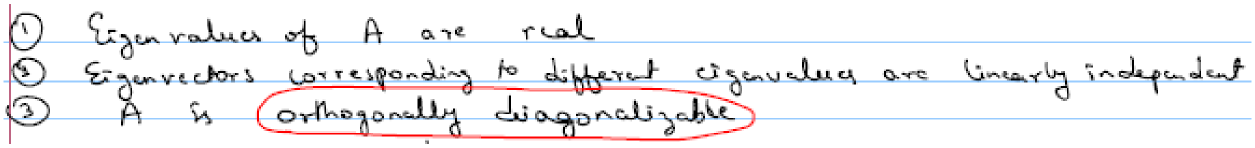
- As an extension to the above argument, following also holds.
  - 

    $S^{-1}A^kS = \Lambda^k$
- S is not unique, since S could contain scaled up eigen vectors as its columns.
- Not all matrices are diagonalizable.
- As per linear algebra, a good approximation of the kth Fibonacci number is



$$F_k \approx \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^k$$

- Any real symmetric matrix A satisfies the following properties:

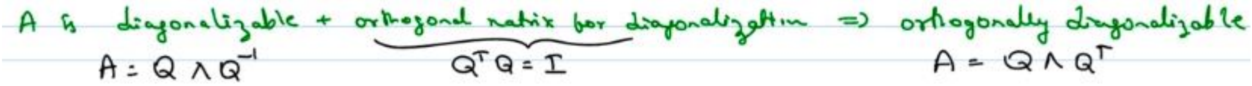


① Eigen values of A are real

② Eigenvectors corresponding to different eigenvalues are linearly independent

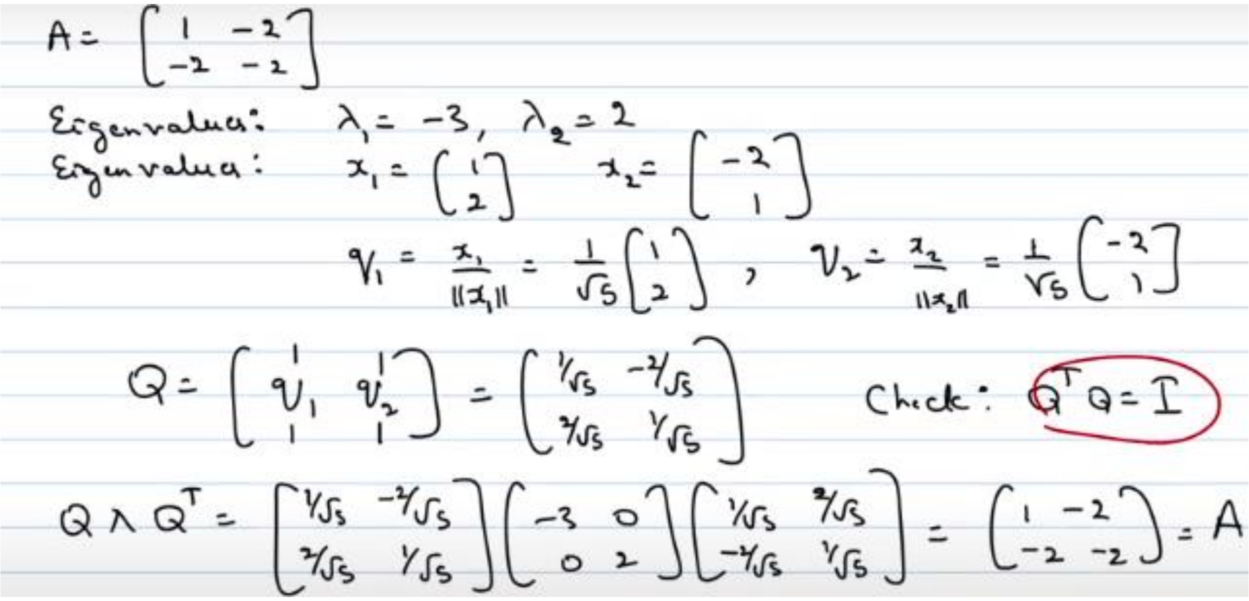
③ A is orthogonally diagonalizable

- Orthogonal matrices satisfy the property $Q^{-1} = Q^T$. Since $QQ^{-1} = I$ for all matrices, in the case of orthogonal matrices, $QQ^T = I$
- Orthogonal diagonalizability can be written as:

  $$\underset{A = Q\Lambda Q^{-1}}{A \text{ is diagonalizable}} + \underbrace{\text{orthogonal matrix for diagonalization}}_{Q^TQ = I} \Rightarrow \underset{A = Q\Lambda Q^T}{\text{orthogonally diagonalizable}}$$

  This is called spectral theorem.
- In order to find Q, find the eigen-vectors and divide them by their corresponding lengths. Following is an example that demonstrates this on a 2*2 matrix.



$$A = \begin{bmatrix} 1 & -2 \\ -2 & -2 \end{bmatrix}$$

Eigenvalua: $\lambda_1 = -3, \quad \lambda_2 = 2$

Eigenvalua: $x_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad x_2 = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$

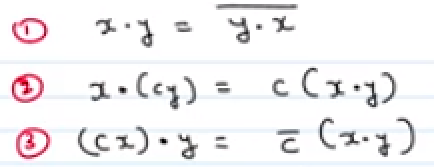
$$v_1 = \frac{x_1}{\|x_1\|} = \frac{1}{\sqrt{5}}\begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad v_2 = \frac{x_2}{\|x_2\|} = \frac{1}{\sqrt{5}}\begin{bmatrix} -2 \\ 1 \end{bmatrix}$$

$$Q = \begin{bmatrix} | & | \\ v_1 & v_2 \\ | & | \end{bmatrix} = \begin{pmatrix} 1/\sqrt{5} & -2/\sqrt{5} \\ 2/\sqrt{5} & 1/\sqrt{5} \end{pmatrix} \qquad \text{Check: } Q^TQ = I$$

$$Q\Lambda Q^T = \begin{bmatrix} 1/\sqrt{5} & -2/\sqrt{5} \\ 2/\sqrt{5} & 1/\sqrt{5} \end{bmatrix}\begin{bmatrix} -3 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 1/\sqrt{5} & 2/\sqrt{5} \\ -2/\sqrt{5} & 1/\sqrt{5} \end{bmatrix} = \begin{pmatrix} 1 & -2 \\ -2 & -2 \end{pmatrix} = A$$

Week5

- If U and V are two symmetric matrices, UV is not symmetric, but U + V is symmetric.
- Complex conjugate of *a + ib* is *a – ib.* Magnitude of the former is given by re^iθ and that of the latter is given by re^-iθ
- In the case of complex vector space, the vector is defined using its complex conjugate. Thus, length of the vector (1, i) is calculated from its complex conjugate (1, -i) and equals 2.
- Given x and y are two vectors in the complex vector space, $\bar{x}^T y \neq \bar{y}^T x$. Thus, x.y is not equal to y.x. Recollect, if the vectors belong to the real space, dot product is commutative.
- Following equations are true when x and y belong to complex vector space.

  ① $x \cdot y = \overline{y \cdot x}$

  ② $x \cdot (cy) = c(x \cdot y)$

  ③ $(cx) \cdot y = \bar{c}(x \cdot y)$

- Other properties of the complex vector space are
  - (x + y).z = x.y + y.z
  - cx.cy = |c|(x.y)

- In the case of a complex vector space, conjugate transpose is defined as

  $A^* = \bar{A}^T = \overline{A^T}$

  Also,

  ① $(A^*)^* = A$

  ② $(AB)^* = B^* A^*$

- In the case of a real vector space, conjugate transpose equivalent is

  Real matrix A, $A^* = A^T$.

  Also,

  Real case equivalents

  ① $(A^T)^T = A$

  ② $(AB)^T = B^T A^T$

- Inner product in a complex space is defined as $x \cdot y = x^* y$
- Matrix A is called Hermitian matrix, if A* = A. In other words, Hermitian matrix is the equivalent of real symmetric matrices in complex space.
- Properties of Hermitian matrices
  - Diagonal entries of a Hermitian matrix are real numbers.
  - All eigen values of a Hermitian matrix are real numbers
  - If eigen-values obtained are distinct, corresponding eigen vectors are orthogonal to each other. Hence, the matrix is orthogonally diagonalizable.
  - Matrix that diagonalizes a Hermitian matrix is called a Unitary matrix. Thus,

    $A = U \Lambda U^*$, Λ is a diagonal matrix with eigenvalues of A.

    In the above case, A is a Hermitian and U is a Unitary matrix.
  - A = UΛU* and Λ = U*AU are equivalent expressions, where A is the Hermitian matrix, U and U* are unitary matrices and Λ is the diagonal matrix.
- Given upper triangular matrix A, AA^* and A^*A are Hermitian matrices.
- Every real diagonal matrix is Hermitian. However, in general, any diagonal matrix is not Hermitian, since the diagonal elements Hermitian matrices cannot be complex.
- Unitary matrices are square matrices with orthonormal columns. For such matrices, $U^* U = I$ or $U^{-1} = U^*$
- If U is unitary matrix, then U^* is also unitary matrix.
- All Hermitian matrices are unitarily diagonalizable but, all unitarily diagonalizable matrices are not Hermitian. Watch https://www.youtube.com/watch?v=VYS9EYZ3gCo for an example diagonalization of a Hermitian matrix.
- Properties of Unitary matrices
  - ||Ux|| = ||x|| (Length unchanged)
  - Eigen-values of unitary matrices have an absolute value equal to 1, although not necessarily real.

    - Eigen-vectors corresponding to the eigen-values of a unitary matrix are orthogonal.
- Given a non-symmetric n * n matrix A, it cannot be diagonalized, but it can be upper-triangularized. There exists a unitary matrix U and an upper triangular matrix T such that A = UTU^*
- To upper-triangularize a matrix, follow these steps.
    - Find the characteristic polynomial of the matrix
    - Find 1st eigen vector. Extend with (n-1) basis, where n is the number of columns
    - Use Gram-schmidt process and arrive at its orthonormal basis. Call this U1.
    - Extract the square matrix at the right-bottom of U1.
    - Repeat all of the above steps on the new matrix, until you get U2.
    - Now, $U_2^* U_1^* A U_1 U_2$ will result in an upper triangular matrix.
- Given an arbitrary basis {u₁, u₂, …uₙ} for a n n-dimensional inner product space V, ***Gram-Schmidt algorithm*** constructs an orthogonal basis {v₁, v₂, …vₙ} for V:

Step 1 Let $\mathbf{v}_1 = \mathbf{u}_1$.

Step 2 Let $\mathbf{v}_2 = \mathbf{u}_2 - \frac{\langle \mathbf{u}_2, \mathbf{v}_1 \rangle}{\|\mathbf{v}_1\|^2}\mathbf{v}_1$.

Step 3 Let

$$\mathbf{v}_3 = \mathbf{u}_3 - \frac{\langle \mathbf{u}_3, \mathbf{v}_1 \rangle}{\|\mathbf{v}_1\|^2}\mathbf{v}_1 - \frac{\langle \mathbf{u}_3, \mathbf{v}_2 \rangle}{\|\mathbf{v}_2\|^2}\mathbf{v}_2.$$

https://www.khanacademy.org/math/linear-algebra/alternate-bases/orthonormal-basis/v/linear-algebra-the-gram-schmidt-process

Singular Value Decomposition

- For a real matrix A, AA^T always results in a real symmetric matrix.
- If $A = Q_1 \Sigma Q_2^T$, then $A^T A$ is $Q_2 \Sigma^T \Sigma Q_2^T$
- If $A = Q_1 \Sigma Q_2^T$, then AA^T is $Q_1 \Sigma \Sigma^T Q_1^T$
- In the case of diagonalizable matrices, we already know how to decompose them using spectral theorem.
- If the matrix is not diagonalizable, we'll still be able to decompose it, as follows.

A can be written as $\underset{m\times n}{A} = \underset{m\times m \text{ matrix}}{Q_1} \Sigma \underset{n\times n \text{ matrix}}{Q_2^T}$

Q_1, Q_2 are orthogonal i.e., $Q_1^T Q_1 = \underset{m\times m}{I}$, $Q_2^T Q_2 = \underset{n\times n}{I}$

$\Sigma = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}$, where $D = \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_r \end{bmatrix}$, $\sigma_i > 0$

NOTE: Note that this works for any real m * n matrix. In this case, Q1 is m * m and Q2 is n * n.

Here Q1 is made up of normalized eigen vectors of AA^T. Q2 is made up of normalized eigen vectors of A^{T}A. Note that both are real symmetric square matrices. These matrices are also orthogonal. σ_i denotes the square root of the eigen values of A^{T}A. The diagonal of the matrix ∑ are normally in a sorted (descending) order.

- Q2^T rotates the input, ∑ stretches it (σ_i gives the semi-major and semi-minor axis length of a stretched unit circle) and Q1^T further rotates it. Effectively, this will produce the same transformation as matrix A will cause.

Week6

- Every quadratic function of the form ax^2+bxy+cy^2 has a stationary point at (0, 0). First derivative of the function becomes zero at this point.
- A function ***f*** that vanishes at (0, 0) and is *strictly* positive at all other points is called ***positive definite***, and is denoted as *f > 0*
- ax^2+bxy+cy^2 is positive definite if and only if a > 0 and ac > b^2
- if ac = b^2, ax^2+bxy+cy^2 is ***positive semi-definite*** if a > 0
- if ac = b^2, ax^2+bxy+cy^2 is ***negative semi-definite*** if a < 0
- If ac < b^2, ax^2+bxy+cy^2 has a saddle point at (0, 0)
- Following is the matrix-based representation of a function in two variables using matrices.

  $$ax^2 + 2bxy + cy^2 = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a & b \\ b & c \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

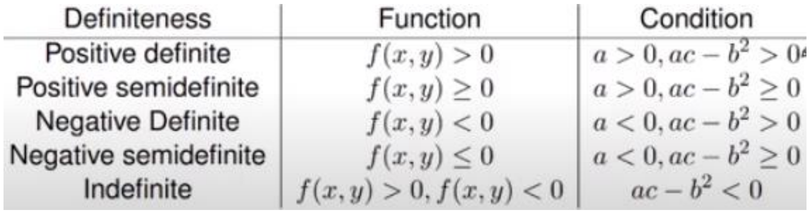
  Let $v = \begin{bmatrix} x \\ y \end{bmatrix}$ and $A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$

  Then $ax^2 + 2bxy + cy^2 = v^T A v$

  NOTE: ***A*** is real-symmetric matrix.
- In general, function in n variables can be represented as

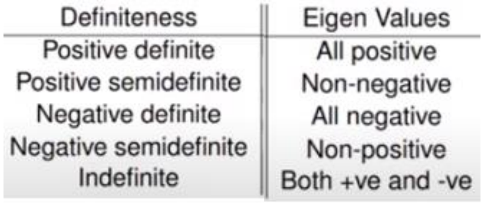
  $$\begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} \begin{bmatrix} a_{11} & - - - & a_{1n} \\ | & \ddots & \\ \vdots & & \ddots \\ a_{n1} & & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j$$

- In general, function ***f*** (in n variables) is positive, if all pivots are positive in its reduced row echelon form.
- Definiteness of a bi-variate function ***f*** represented as a 2 x 2 matrix ***A*** can be determined using the following table, from its elements:



Definiteness	Function	Condition
Positive definite	$f(x,y) > 0$	$a > 0, ac - b^2 > 0$
Positive semidefinite	$f(x,y) \geq 0$	$a > 0, ac - b^2 \geq 0$
Negative Definite	$f(x,y) < 0$	$a < 0, ac - b^2 > 0$
Negative semidefinite	$f(x,y) \leq 0$	$a < 0, ac - b^2 \geq 0$
Indefinite	$f(x,y) > 0, f(x,y) < 0$	$ac - b^2 < 0$

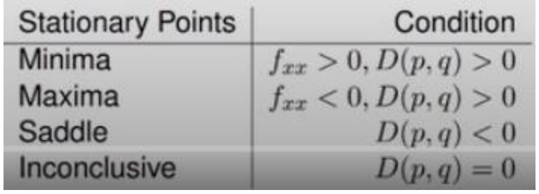
- Definiteness of a function ***f*** represented as a matrix ***A*** can be determined using the following table, from its eigen values:



Definiteness	Eigen Values
Positive definite	All positive
Positive semidefinite	Non-negative
Negative definite	All negative
Negative semidefinite	Non-positive
Indefinite	Both +ve and -ve

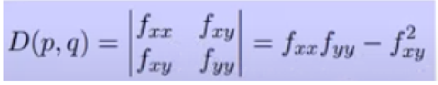
NOTE: Matrix ***A*** represents the function ***f*** in matrix form.

- For any function ***f***, following are true at point (p, q):



Stationary Points	Condition
Minima	$f_{xx} > 0, D(p,q) > 0$
Maxima	$f_{xx} < 0, D(p,q) > 0$
Saddle	$D(p,q) < 0$
Inconclusive	$D(p,q) = 0$

where ***f**_xx_* is the second order derivative of ***f*** and ***D*** is the determinant of the second order derivatives at the given point (given below)



$$D(p,q) = \begin{vmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{vmatrix} = f_{xx}f_{yy} - f_{xy}^2$$

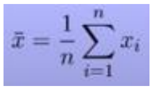
- Example:

① $f(x,y) = 2x^2 + 4xy + y^2$ ⟸ saddle point at origin since $ac = 2 < b^2 = 4$, $A = \begin{bmatrix} 2 & 2 \\ 2 & 1 \end{bmatrix}$

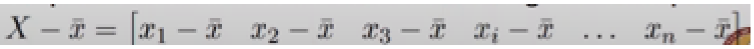
② $f(x,y) = 2xy$ ⟸ saddle at origin. $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

Principal Component Analysis

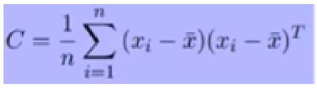
- PCA aims to project given data onto a lower-dimensional subspace, such that reconstruction error is minimized and variance of the projected data is maximized.
- Following are the steps involved:
  - <u>**Step1**</u>: Data matrix X has n data points, each with m features (dimensions)
  - <u>**Step2**</u>: Find the mean vector of data points

    $$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$
  -
  - <u>**Step3**</u>: Subtract mean vector from the given data points.

    $$X - \bar{x} = [x_1 - \bar{x} \quad x_2 - \bar{x} \quad x_3 - \bar{x} \quad x_i - \bar{x} \quad \ldots \quad x_n - \bar{x}]$$
  - <u>**Step4**</u>: Find the covariance matrix C (A symmetric m x m matrix)

    $$C = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^T$$
  - <u>**Step5**</u>: Find the eigen-values and eigen-vectors u_j of C.
  - <u>**Step6**</u>: Choose the eigen-vectors corresponding to the first k eigen-values and derive the transformed data points.

- Scalar projection of a datapoint x_i on u_j: $\alpha_j = x_i^T u_j$
- Transformed data point corresponding to x_i: $\tilde{x}_i = \sum_{j=1}^{k} \alpha_j u_j$

- **Step7**: Calculate the reconstruction error

  $$J = \frac{1}{n}\sum_{i=1}^{n} ||x_i - \tilde{x}_i||^2$$

  and projected variance

  $$\lambda_1 + \lambda_2 + \ldots + \lambda_k$$

For a data set $D = \{x_1, x_2, ...\}$ that is centered around $\bar{x}$, the projected variance along u is

- $\frac{1}{n}\sum_{i=1}^{n}(x_i^T u - \bar{x}^T u)^2$
- Rank of the matrix $C = \frac{1}{n}\Sigma_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^T$ is less than or equal to n.
- If u is an eigenvector of a matrix XX^T, then $X^T u$ is an eigenvector of $X^T X$
- Appending a 1 to the end of every data point does not change the results of performing PCA, except that the useful principal component vectors have an extra 0 at the end and there is one useless component with eigen value 0.
- If you perform a 90-degree clockwise rotation of the data points before performing PCA, the largest eigenvalue does not change.
- If you perform a 90-degree clockwise rotation of the data points before performing PCA, the variance along each eigen vector does not change.

Week7

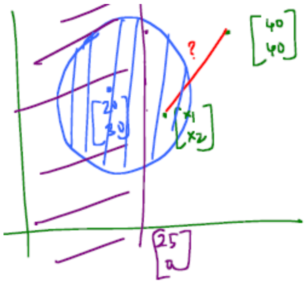
- Pillars of machine learning are Linear algebra, Probability, Optimization
- The structure and relationship between data points is dealt within *Linear Algebra*.
- Modelling noise/uncertainty in data is done using *Probability*.
- *Optimization* is the mathematical tool that helps in converting data to decisions.
- The "best" of something often means the least loss or maximum reward, and found using calculus.
- A typical optimization problem looks like this:

How close can a cow tied to point (20, 30) using a rope that measures 10 units get to the grass at (40, 40), given that a fence is placed along the perpendicular at x = 25.

Given the problem, following equation mathematically represent the objective function that must be minimized.

$$d(x_1, x_2) = (x_1 - 40)^2 + (x_2 - 40)^2$$

And, following are the two equations that mathematically represent the constraints.



Rope restriction $(x_1 - 20)^2 + (x_2 - 30)^2 \le 10^2$

Fence restriction $x_1 \le 25$

- In general, constraints could include inequality and equality constraints and can be represented pictorially as follows:

General form

$$\min_{x \in \mathbb{R}^d} f(x) \quad \leftarrow \text{OBJECTIVE}$$

VARIABLE/PARAMETER

$$\begin{cases} g_i(x) \le 0 & \forall i = 1, \cdots, K \\ h_j(x) = 0 & \forall j = 1, \cdots, L \end{cases}$$

CONSTRAINTS

INEQUALITY CONSTRAINTS

EQUALITY CONSTRAINTS.

- One method to find the minimum/maximum parameters of an equation is to find the first derivative and equate to 0. As an example, to minimize y = $(x - 5)^2$

$$f'(x) = 2(x^* - 5) = 0$$

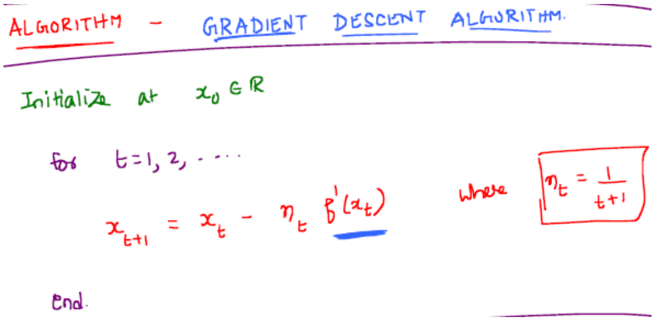
$$\Rightarrow \boxed{x^* = 5}$$

NOTE: However, not all models can be minimized like this, since this might require us to solve equations with degrees higher than 2.

- One of the algorithms that could solve the above equation is as follows:
  - Start with x = x_0
  - Find the derivative of the function at the current x.
  - Add negative of the derivative to the current value of x, to get the new x.

$$\boxed{x_{t+1} = x_t + d} \quad \text{where} \quad \boxed{d = -f'(x)}$$

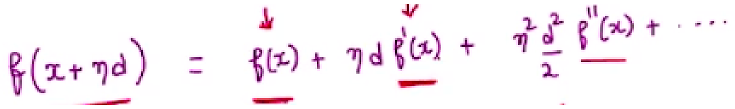
  - Repeat above two steps multiple times.
- Unfortunately, the above algorithm oscillates between x_t and x_{t+1}. This can be solved by introducing a scalar factor (step-size η) for d.
- Step size of η = (½)^n fails to work if x_0 (starting value) is very far away from the actual value (in fact, it can never cross a value of 2). Hence, it's a better idea to work with 1/(t + 1) which will eventually *converge* to the actual value.
- Considering the above aspects, an acceptable algorithm is as follows:



ALGORITHM – GRADIENT DESCENT ALGORITHM.

```
Initialize at $x_0 \in \mathbb{R}$
    for $t = 1, 2, \ldots$
        $x_{t+1} = x_t - \eta_t f'(x_t)$     where $\boxed{\eta_t = \frac{1}{t+1}}$
    end.
```

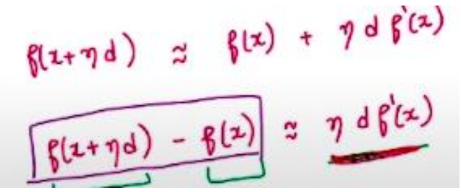
- The afore mentioned *Gradient Descent* algorithm converges to a local minimum of the given function. Even though, no general purpose algo exists that'll converge to a global minimum, converging to a local minimum serves well in most machine learning exercises. Such functions are called convex functions.
- Taylor's series can be used to understand why derivative appears in the update rule.



$$f(x+\eta d) = f(x) + \eta d f'(x) + \frac{\eta^2 d^2}{2} f''(x) + \cdots$$

NOTE: This implies that the function value at any x_hat, can be calculated if the local information (at x) is known.

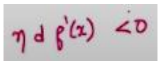
For small-enough η, ignore the higher-order terms (from the 3rd term onwards) and f at the new x can be approximated as follows:



$$f(x+\eta d) \approx f(x) + \eta d f'(x)$$

$$\boxed{f(x+\eta d) - f(x)} \approx \eta d f'(x)$$

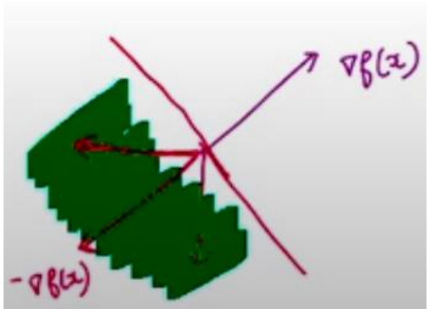
Since the LHS is negative quantity (due to decreasing function value), it's required that



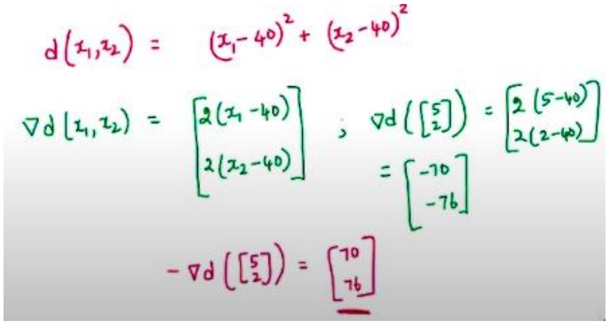
$\eta d f'(x) < 0$

Since η is a positive quantity (step-size), $d f'(x) < 0$. If we choose d to be -f`(x), then it satisfies this condition, since negative of the square of derivative of the function is always less than 0. Thus proved.

When dealing with higher dimensions of the input data, this can be rewritten as $d^T \nabla f(x) < 0$. Pictorially, this constitutes the region to the left of the perpendicular to the gradient vector of the function.



- In the case of the cow-grass example above, in order to move towards the grass from (x1, x2), say (5, 2), following calculations should be done.



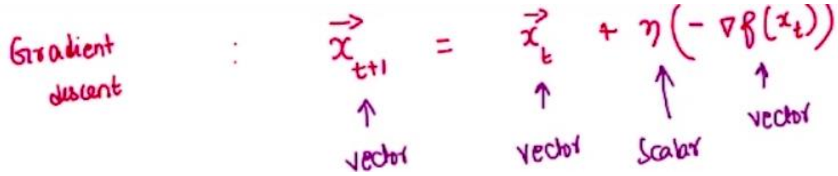
$$d(x_1,x_2) = (x_1-40)^2 + (x_2-40)^2$$

$$\nabla d(x_1,x_2) = \begin{bmatrix} 2(x_1-40) \\ 2(x_2-40) \end{bmatrix} \; ; \; \nabla d\left(\begin{bmatrix} 5 \\ 2 \end{bmatrix}\right) = \begin{bmatrix} 2(5-40) \\ 2(2-40) \end{bmatrix} = \begin{bmatrix} -70 \\ -76 \end{bmatrix}$$

$$-\nabla d\left(\begin{bmatrix} 5 \\ 2 \end{bmatrix}\right) = \underline{\begin{bmatrix} 70 \\ 76 \end{bmatrix}}$$

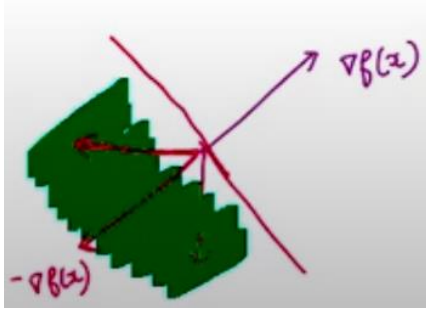
Now, this vector gives the direction that takes the cow (presently at (5, 2)) towards the grass at (40, 40). In order to compute the vector, choose an appropriate η to scale this direction vector before adding to (5, 2).

Generalizing this algo for an d-dimensional space, we get



$$\text{Gradient descent} \quad : \quad \underset{\overset{\uparrow}{\text{vector}}}{\vec{x}_{t+1}} = \underset{\overset{\uparrow}{\text{vector}}}{\vec{x}_t} + \underset{\overset{\uparrow}{\text{Scalar}}}{\eta}\left(-\underset{\overset{\uparrow}{\text{vector}}}{\nabla f(x_t)}\right)$$

As mentioned before, this is not the *only* direction to move so that f reduces. It can take any of the infinite directions to the left side of the perpendicular to the gradient vector of the function.



- Newton method is an alternative (to gradient descent) algorithm to calculate the minimum value of a function in iterations. Following is the update rule in this case.

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

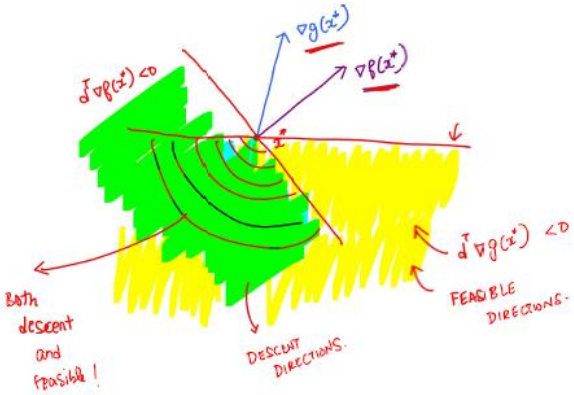
This method might seem to have more precision than the gradient descent method, but in the case of higher dimensions, it requires computing the second order derivative of the function (Hessian matrices) that can get tough to compute.

- Taylor's series can be used to derive the linear approximation formula from week-2 as follows. Assuming x = a + Δ, Taylor's series can be written as
  f(x) = f(a) + Δ.f`(a) + Δ^2.f``(a) + …

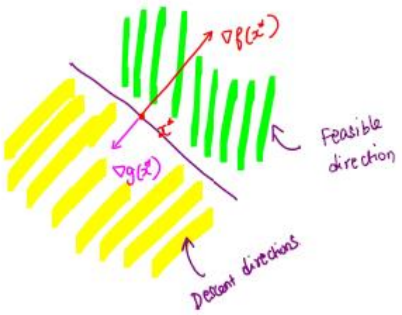
  The above can be rewritten as
  f(x) = f(a) + (x – a).f`(a) + $(x - a)^2$.f``(a)
  This equation is the linear approximation formula learnt during week-2.

Week8

- Given a problem that requires to minimize objective function f(x) such that g(x) satisfies the inequality g(x) <= 0, point x^* will solve it *optimally*, if and only if there is no *further* descent direction that's also feasible. In the above context, descent direction is dependent on f(x) and is given by direction d such $d^T\nabla f(x^*) < 0$, and feasible direction is given by the same direction d such that $d^T\nabla g(x^*) <= 0$. In other words, the problem is solved only if there is no direction d available such that both $d^T\nabla f(x^*) < 0$ and $d^T\nabla g(x^*) <= 0$.
- The feasible descent directions are depicted below.

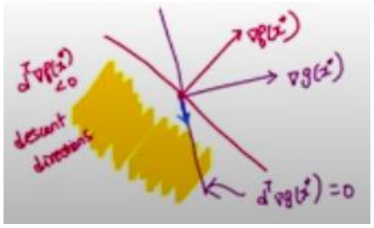


  From the above picture, it's clear that the intersection of green and yellow regions comprises of all feasible descent directions.
- Now, x* is considered an optimal solution only when $d^T\nabla f(x^*)$ and $d^T\nabla g(x^*)$ are anti-parallel as depicted below.



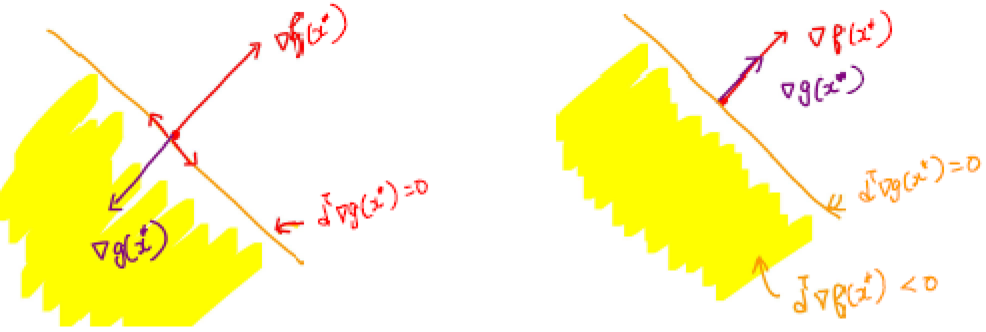
This is mathematically represented in an equation as $\nabla f(x^*) = -\lambda \nabla g(x^*)$, where λ is a positive scalar value.

- If the inequality in the above problem is changed to an equality condition (g(x) = 0), x^* is optimal only when $d^T\nabla g(x^*) = 0$, where d denotes the direction (vector) of movement. This is pictorially shown as follows.



Thus, all feasible directions lie on a line (with the blue arrow)

- Extending this argument, the best possible (optimal) solutions for the constrained (by equality) optimization problem defined above occur when f and g move parallel or anti-parallel as shown in the picture below.



and mathematically represented as $\nabla f(x^*) = -\lambda \nabla g(x^*)$, where λ (called *Lagrange multiplier*) is any non-zero scalar value.

- For example, consider a problem defined by the following bi-variate objective function f and constraint g.

$$f(x_1, x_2) = x_1^2 + 2x_2 + 4x_2^2$$

$$g(x_1, x_2) = x_1^2 + x_2^2 - 1$$

The gradient of these functions are as follows:

$$\nabla f\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}\right) = \begin{bmatrix}2x_1\\2+8x_2\end{bmatrix} \quad ; \quad \nabla g\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}\right) = \begin{bmatrix}2x_1\\2x_2\end{bmatrix}$$

Solution per Lagrange method is as follows:

$$\underbrace{\begin{bmatrix}2x_1\\2+8x_2\end{bmatrix}}_{\nabla f\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}\right)} = -\lambda \underbrace{\begin{bmatrix}2x_1\\2x_2\end{bmatrix}}_{\nabla g\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}\right)} \Rightarrow \quad \begin{aligned} 2x_1 &= -\lambda 2x_1 \quad -① \\ 2+8x_2 &= -\lambda(2x_2) \quad -②. \end{aligned}$$

$$① \Rightarrow \quad 2x_1 + \lambda 2x_1 = 0 \Rightarrow 2x_1(1+\lambda) = 0$$

$$\Downarrow$$

$$\text{either} \quad x_1 = 0 \quad (\text{or}) \quad \lambda = -1$$

Substituting $x_1 = 0$ into the second equation,

$\underline{\text{Case 1}}$: $\boxed{\lambda = -1}$

$$2+8x_2 = -(-1)\,2x_2 \Rightarrow 2+8x_2 = 2x_2 \Rightarrow -6x_2 = 2$$

$$\Rightarrow \boxed{x_2 = -1/3}$$

Now, substituting this value of x_2 into the equation representing constraint g, we have

$$x_1^2 + x_2^2 = 1 \Rightarrow x_1^2 = 1 - 1/9 = x_1^2 = 8/9 \Rightarrow x_1 = \left\{+\frac{\sqrt{8}}{3}, -\frac{\sqrt{8}}{3}\right\}$$

thus, yielding points are $\left\{\begin{bmatrix}\sqrt{8}/3\\-1/3\end{bmatrix}, \begin{bmatrix}-\sqrt{8}/3\\-1/3\end{bmatrix}\right\}$

$\underline{\text{Case 2}}$: $x_1 = 0$

$$x_1^2 + x_2^2 = 1 \Rightarrow x_2^2 = 1 \Rightarrow x_2 = 1 \text{ or } -1$$

thus, yielding points $\left\{\begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}0\\-1\end{bmatrix}\right\}$

Now, the solution to the problem is worked out from these points, as follows:

To find $\boxed{\min_{x} f(x) \quad g(x)=0}$, Substitute each potential solution into f

$$f(x) = x_1^2 + 2x_2 + 4x_2^2$$

$$f\left(\begin{bmatrix}0\\1\end{bmatrix}\right) = 6 ✓$$

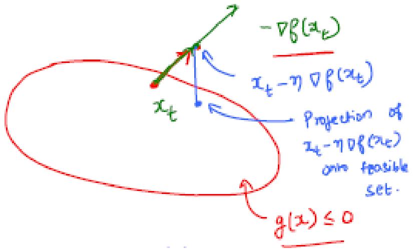
$$f\left(\begin{bmatrix}0\\-1\end{bmatrix}\right) = 2 ✓$$

$$f\left(\begin{bmatrix}\sqrt{8}/3\\-1/3\end{bmatrix}\right) = \frac{8}{9} - \frac{2}{3} + \frac{4}{9} = \frac{4}{3} - \frac{2}{3} = 2/3 ✓$$

$$f\left(\begin{bmatrix}-\sqrt{8}/3\\-1/3\end{bmatrix}\right) = 2/3 ✓$$

Thus, the point (0, 1) is called the maximizer and the points $\left\{\begin{bmatrix}\sqrt{8}/3\\-1/3\end{bmatrix}, \begin{bmatrix}-\sqrt{8}/3\\-1/3\end{bmatrix}\right\}$ are called minimizers.

- In general, it may not be always possible to solve such constrained optimization problems using Lagrange method, in which case it can be solved through the *Projected* Lagrange method.
- As far the objective function is a convex function, the gradient descent can be projected onto the feasible set to solve the problem. This is pictorially represented as

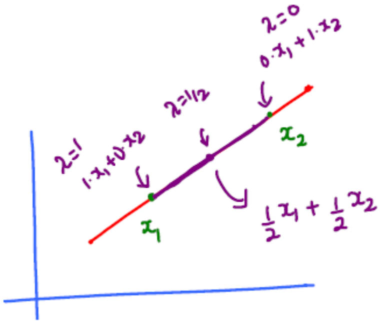


- The corresponding algorithm is called Projected Gradient Descent and rewritten as follows:

```
PROJECTED GRADIENT DESCENT
    $x_0 \leftarrow$ initialization
    for $t = 1, \dots, T$
        $x_{t+1} = \underbrace{\Pi\big(\underbrace{x_t - \eta \nabla f(x_t)}_{\text{gradient step}}\big)}_{\text{Projection step.}}$
    end.
```

Convexity

- A set of points containing x_1 and x_2 is called a convex set when $\lambda x_1 + (1-\lambda) x_2$ is also in the set, where λ is in [0, 1]
- A pictorial representation of this concept is as follows:



Note that the line joining the two points x_1 and x_2 will contain all points in the convex set.

- A hyperplane is convex, since a line joining any two points in the hyperplane produces a convex set.
- If two sets S1 and S2 are convex, S1⋂S2 is also convex.
- Set defined as {x ∈ R^d: Ax = b} given A ∈ R^{m*d} and b ∈ R^{d*1} is convex.
- Convex combinations is defined as follows:

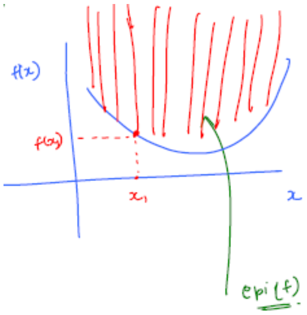
Convex Combinations

Let $\boxed{S=\{x_1,\dots,x_n\}} \subseteq \mathbb{R}^d$. Then we will say $z \in \mathbb{R}^d$ to be a $\underbrace{\text{convex combination}}$ of boints in S if $\exists\, \lambda_1,\dots,\lambda_n$ s.t $\boxed{\lambda_i \geq 0}, \boxed{\sum_{i=1}^{n}\lambda_i = 1}$ s.t $\boxed{z = \lambda_1 x_1 + \lambda_2 x_2 + \dots + \lambda_n x_n.}$

- Set consisting of all such convex combinations is called a *convex hull* and lies inside the area bounded by points x_1, x_2…x_n. This is mathematically represented as follows:

$$\underbrace{\text{Convex-Hull}(S)}_{\{x_1,\dots,x_n\}} = \left\{ z : z = \sum_{i=1}^{n} \lambda_i x_i \text{ for some } \lambda_1,\dots,\lambda_n \geq 0, \ \sum \lambda_i = 1 \right\}$$

- Alternatively, convex hull is defined to be the intersection of all sets containing the points x_1, x_2…x_n.
- A convex hull is also a convex set.
- epigraph epi(f) is defined as [x, z], where z > f(x). Note that epigraph is R^{d+1}. An example in R1 is shown below.



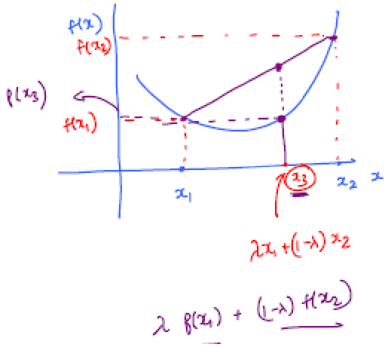
- Function is a convex function, when its domain and its epigraph is a convex set. If the function's domain is not a convex set, the function is not convex.

- Here's another definition of convex function.

A function $f: \mathbb{R}^d \to \mathbb{R}$ is convex iff

$\forall x_1, x_2 \in \mathbb{R}^d$ and all $\lambda \in [0,1]$

$$\boxed{f(\lambda x_1 + (1-\lambda) x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)}$$

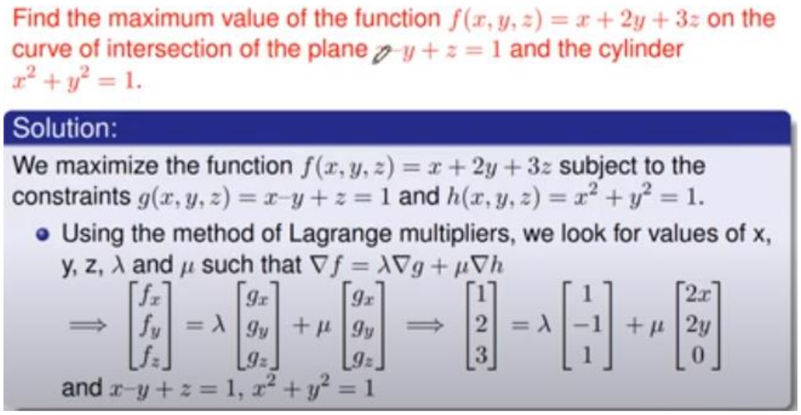


- In yet another definition, the function is convex if the value at any point on its domain is greater than the linear approximation at that point. It's stated mathematically as

$$\boxed{f(y) \geq f(x) + (y-x)^T \nabla f(x)}$$

NOTE: For this definition, it's assumed that the function is differentiable.

- A differentiable function f: Rd->R is convex, if
  - The Hessian matrix H is positive definite or positive semi-definite matrix, det(H) > 0
  - Eigenvalues of the Hessian matrix H are non-negative, Eigenvalues(H) >= 0.
- If f is a convex function, all its local minima are also global minima. This implies that optimization logic that minimizes to local minima also minimizes to global minima, in the case of convex functions.
- f is a convex function, if the determinant of the Hessian matrix formed by f_{xx}, f_{xy} and f_{yy} is positive. If the determinant is negative, the function is not convex.
- In order to find the interval over which the function is convex, calculate its double-derivative f`` and solve f`` > 0. If f`` < 0, the function is concave.
- Function *f*: Rd -> R, f(x) = x^TAx is convex, if matrix A is positive definite or positive semi-definite.
- To find the maximum/minimum of an objective function f given one or more constraints (g and h), start with the Lagrange multiplier method $\nabla f = \lambda \nabla g + \mu \nabla h$. Solve for variables, satisfying the above equation and all the given constraints. Example follows.



Find the maximum value of the function $f(x,y,z) = x + 2y + 3z$ on the curve of intersection of the plane $x - y + z = 1$ and the cylinder $x^2 + y^2 = 1$.

Solution:

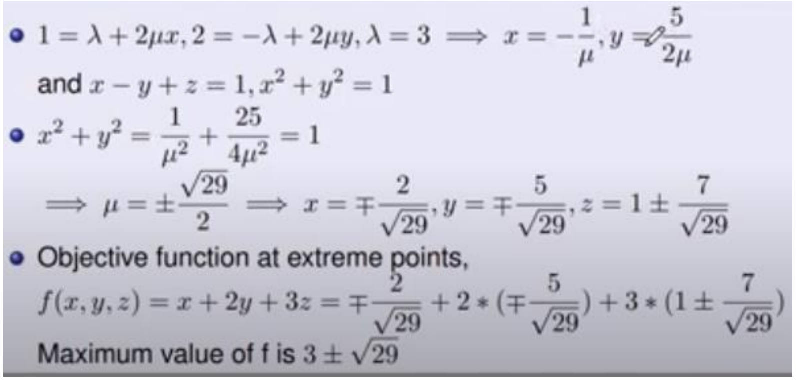
We maximize the function $f(x,y,z) = x + 2y + 3z$ subject to the constraints $g(x,y,z) = x - y + z = 1$ and $h(x,y,z) = x^2 + y^2 = 1$.

- Using the method of Lagrange multipliers, we look for values of x, y, z, λ and μ such that $\nabla f = \lambda \nabla g + \mu \nabla h$

$$\implies \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \lambda \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} + \mu \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} \implies \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \lambda \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} + \mu \begin{bmatrix} 2x \\ 2y \\ 0 \end{bmatrix}$$

and $x - y + z = 1$, $x^2 + y^2 = 1$

From the above equation, we get



- $1 = \lambda + 2\mu x, 2 = -\lambda + 2\mu y, \lambda = 3 \implies x = -\frac{1}{\mu}, y = \frac{5}{2\mu}$ and $x - y + z = 1, x^2 + y^2 = 1$
- $x^2 + y^2 = \frac{1}{\mu^2} + \frac{25}{4\mu^2} = 1$ $\implies \mu = \pm\frac{\sqrt{29}}{2} \implies x = \mp\frac{2}{\sqrt{29}}, y = \mp\frac{5}{\sqrt{29}}, z = 1 \pm \frac{7}{\sqrt{29}}$
- Objective function at extreme points, $f(x,y,z) = x + 2y + 3z = \mp\frac{2}{\sqrt{29}} + 2 * (\mp\frac{5}{\sqrt{29}}) + 3 * (1 \pm \frac{7}{\sqrt{29}})$ Maximum value of f is $3 \pm \sqrt{29}$

Week9

- For a convex function, every local minima is a global minima too. It means that there could be multiple local minima with the same value.
- Set of all global minima of a convex function is a convex set.
- For any function *f* (R^d->R) that is both differentiable and convex, if $x^* \in R^d$ is a global minimum of *f*, then $\nabla f(x^*) = 0$. Note that this is true, whether the function is convex or not. This is called the first-order optimality condition.
- Converse of the above theorem is also true; thus, if $\nabla f(x^*) = 0$, then x* is a global minimum. However, this is true only when the function is convex.
- Properties of convex functions:
  - If *f* and *g* (both R^d->R) are convex functions, then h(x) = f(x) + g(x) is also a convex function.
  - If *f* and *g* (both R^d->R) are convex functions, then h(x) = f(x) * g(x) is also a convex function.
  - If *f* (R->R) and *g* (R^d->R) are convex functions, where *f* is non-decreasing, then h(x) = f(g(x)) is also a convex function.
  - If *f* (R->R) and *g* (R^d->R) are convex functions, where *g* is linear, then h(x) = f(g(x)) is also a convex function.
  - It follows from the above that if *f* is not non-decreasing, or *g* is linear, f(g(x)) isn't necessarily convex.
- In a linear regression problem of machine learning, where the regression line is represented by a linear equation $h(x) = w^T x$ ($w \in R^d$), sum of squares error is represented as $\sum_{i=1}^{n}(w^T x_i - y_i)^2$ where n is the number of items in the data set.
- Specific goal of a linear regression is to minimize the sum of squares error above. This can be mathematically represented as

  $$\min_{w \in R^d} \frac{1}{2}\sum_{i=1}^{n}(w^T x_i - y_i)^2$$

  NOTE: ½ is a scaling factor, applied merely to render the calculations simpler.
- Above equation of sum of squares error is a convex function, since it's a composition of a convex function $f(z) = z^2$ and $g(w) = w^T x - y_i$. Since g(w) is a linear function, the composition f(g) is a convex function.

- In order to find the minimum w, equate the gradient to 0 to get $w^* = (x^Tx)^\dagger (x^Ty)$ where $(x^Tx)^\dagger$ is called a pseudo-inverse. This is the analytical solution.
- To find the global minimum of the composition, take its gradient, which yields

  $$\nabla f(w) = (x^Tx)w - x^Ty$$

- Computation of the inverse is O(d^3), where d is the number of dimensions and is highly expensive. Hence, it's advisable to use the iterative algorithm which computes gradient descent on each step.

  $$W^{t+1} = W^{t} - \eta_t \underbrace{\nabla f(W^t)}$$

- Note that the gradient calculation $(x^Tx)w - x^Ty$ doesn't involve an inverse computation and hence much more efficient computationally, and can be further simplified by approximating it using a technique called *stochastic* gradient.
- The stochastic gradient technique essentially samples a smaller subset of datapoints (uniformly, at random) and computes gradient (over several iterations) on this subset, instead of using all datapoints. When averaged over all iterations, the resultant w should be equal to w*
- Finding the optimum value of *f* that's constrained by say, h(x) <= 0 is same as minimizing the maximum of a Lagrangian L(x, λ). This is called ***primal problem*** and is represented as follows:

  $$\boxed{\begin{array}{l}\min_x \; f(x) \\ \text{s.t } h(x) \le 0\end{array}} \quad \equiv \quad \boxed{\min_x \left[\max_{\lambda \ge 0} \; L(x,\lambda)\right]}$$

  NOTE: $L(x,\lambda) = f(x) + \lambda h(x)$

  NOTE2: λ is called the lagrangian multiplier.
- The min-max (primal) problem can be converted to a max-min (dual) problem as follows:

  $$\max_{\lambda \ge 0} \left[\min_x \; L(x,\lambda)\right].$$

  Note that the min (inner function) problem is a concave function.
- Solution for the min-max problem is as follows.

  $$x^* = \arg\min_x \left[\max_{\lambda \ge 0} \; f(x) + \lambda h(x)\right]$$

- Solution for max-min problem (dual) is as follows

  $$\lambda^* = \arg\max_{\lambda \ge 0} \left[\underbrace{\min_x \; f(x) + \underline{\lambda} h(x)}\right]$$

- When the function value at the dual optimum is less than or equal to function value at the primal optimum, it's called weak duality.
- If *f* and *h* are convex functions, then we get strong duality. At this point, f(x*) = h(x*). In this case, we can solve either primal problem *or* the dual problem to arrive at the optimal solution x* for *f*.

- Thus, for the objective function *f* and inequality constraint *h*, the (local) optimal solution (x*, λ*) is given by the Karush-Kuhn-Tucker (KKT) conditions, which are enlisted as follows:

(a) $\nabla f(x^*) + \lambda^* \nabla h(x^*) = 0$ [Stationarity condition]. ✓

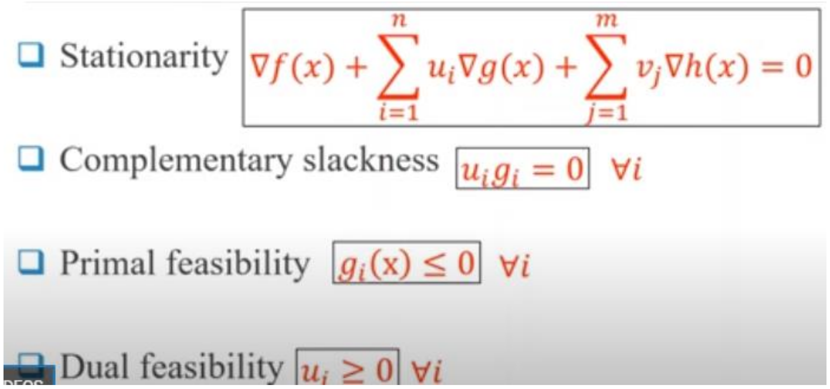
(b) $\lambda^* h(x^*) = 0$ [Complementary slackness condition]

(c) $h(x^*) \leq 0$ [Primal feasibility]. ✓

(d) $\lambda^* \geq 0$ [Dual feasibility] ✓

NOTE: If the functions *f* and *h* are convex, these conditions ensure optimal solution, that're not just local.

- If the list of constraints includes equality constraints (represented as l) in addition to the inequality conditions (represented as h), the KKT conditions can be re-written as



□ Stationarity $\boxed{\nabla f(x) + \sum_{i=1}^{n} u_i \nabla g(x) + \sum_{j=1}^{m} v_j \nabla h(x) = 0}$

□ Complementary slackness $\boxed{u_i g_i = 0}$ $\forall i$

□ Primal feasibility $\boxed{g_i(\mathrm{x}) \leq 0}$ $\forall i$

□ Dual feasibility $\boxed{u_i \geq 0}$ $\forall i$

NOTE: Vectors *u* and *v* represent the lagrangian multipliers for the inequality and the equality constraints respectively.

NOTE2: If the (primal) inequality constraints use > or >=, the dual feasibility should use <=

- Dual of dual is primal
- If either the primal or dual problem has an infeasible solution, then the value of the objective function of the other is unbounded.
- If either the primal or dual problem has a solution then the other also has a solution and their optimum values are equal.
- If one of the variables in the primal has unrestricted sign, the corresponding constraint in the dual is satisfied with equality.

Week11

- Experiment in a sample space is represented as (Ω, F, P), where Ω is the sample space, F is the set of experiments and P is the probability
- Axioms of probability with continuous random variables
  - P(A) >= 0
  - P(Ω) = 1

  - P (A$_1$ U A$_2$ …U A$_n$) = ∑P(A$_i$) for I = 1 to n
- In the case of continuous variables, domain and range of sample space Ω is uncountably finite (set all real numbers)
- When the continuous random variable X takes an exact value x, the probability is 0 by definition.
- PDF and CDF of continuous random variable is defined as follows:

$$f_X(x) = \frac{P(X \in [x, x+dx])}{dx} \qquad \text{PDF}$$

$$F_X(x) = P(X \leq x) \qquad \text{CDF}$$

- Following are the properties of PDF and CDF

i) $f_X(x) \geq 0$

ii) $\int_{-\infty}^{\infty} f_X(x)dx = 1$

(ii) $F_X(-\infty) = 0$

(iii) $F_X(\infty) = 1$

(iv) F_X is increasing

- Also, F$_x$(b) – F$_x$(a) = P (a < X <= b)
- For a continuous random variable X with PDF f$_x$, an event A is a subset of a real line and its probability is computed as P(A) = ∫$_A$ f(x)dx
- Conditional probability is given by the following formula

$$f_{X|A}(x) = \frac{P(X \in [x, x+dx] | A)}{dx}$$

- When PDF is integrated, you get the CDF. Likewise, when CDF is differentiated, you get PDF.
- Expectation of a continuous random variable is given by

$$E[X] = \int_{-\infty}^{\infty} x \cdot f_X(x)dx$$

- Properties of expectation are

i) $E[X+Y] = EX + EY$

ii) $Y = g(X)$

$$E[Y] = \int_{-\infty}^{\infty} g(x) \cdot f_X(x)dx$$

- Variance is given by $EX^2 - (EX)^2$
- Properties of variance are

i) $Var[X+Y] \neq var[X] + var[Y]$

ii) $Var[aX] = a^2 \, var[X]$

iii) $Var[x] \geq 0$

- Standard deviation is the square root of the variance
- In the case of uniform distribution in the interval [a, b], expectation is $(b - a)^2 / 12$
- Total expectation law is as follows:

$$E[X] = E[X|A] \cdot P(A) + E[X|A^c] \, P(A^c)$$

- Joint distribution

$$f_{XY}(x,y) = \frac{P(X \in [x, x+dx], Y \in [y, y+dy])}{dx \cdot dy}$$

- Properties of joint distribution

i) $f_{XY}(x,y) \geq 0$

ii) $\int_x \int_y f_{XY}(x,y) \, dx \, dy = 1$

- Cumulative distribution

$$F_{XY}(x,y) = P(X \leq x, Y \leq y)$$

- Properties of cumulative distribution

(i) $F_{XY}(-\infty, -\infty) = 0$

(ii) $F_{XY}(\infty, \infty) = 1$

$F_{XY}(x,y)$ is non-decreasing. Cumulative probability increasing when either x or y increases.

- Marginal densities are given by

$$f_X(x) = \int_{-\infty}^{\infty} f_{XY}(x,y) \, dy$$

$$f_Y(y) = \int_{-\infty}^{\infty} f_{XY}(x,y) \, dx$$

- Conditional density is given by

$$f_{X|Y}(x|y) = \frac{f_{XY}(x,y)}{f_Y(y)}$$

- X and Y are independent if

$$f_{XY}(x,y) = f_X(x) \cdot f_Y(y)$$

- Covariance of random variables X and Y is defined as

$$cov[X,Y] = E[(X-EX)(Y-EY)]$$
$$= E[XY] - (EX)(EY)$$

- Correlation coefficient is given by

$$\rho[X,Y] = \frac{cov[X,Y]}{\sqrt{Var[X] \cdot Var[Y]}}$$

- If X and Y are independent random variables, covariance is 0, implying that the correlation coefficient is 0 and hence they are uncorrelated. However, the reverse is not true. Uncorrelated random variables need not be independent.
- To get the joint distribution of derived random variables from original random variables, use

$f_{YZ}(y,z) = f_{WX}(q(y,z), r(y,z))|J|$, where J represents the Jacobian $J = \begin{bmatrix} \frac{\partial q}{\partial y} & \frac{\partial q}{\partial z} \\ \frac{\partial r}{\partial y} & \frac{\partial r}{\partial z} \end{bmatrix}$

- Memoryless property of exponential distribution

$$P(X \geq a | X \geq b) = P(X \geq a-b)$$

- Expectation of exponential distribution exp(λ) is 1/λ and variance is $(1/\lambda)^2$

$$X \sim exp(\lambda)$$

- When $Y \sim exp(\tau)$ and $Z = min(X,Y)$, then $Z \sim exp(\lambda+\tau)$

Week12

- Covariance matrix is always a square matrix. For a system with two variables, it's a 2 x 2 matrix.
- $Cov[X_1,X_2] = E[X_1X_2] - E[X_1]E[X_2]$
- Variances are in the position of diagonal elements in a covariance matrix.
- If Z = CY, where Y is a normal distribution and C is the coefficient matrix, then

$$E(Z) = E(CY) = CE(Y)$$
$$Var(Z) = Var(CY) = CVar(Y)\bar{C}$$

- Cov[X_1, X_2] = Cov[X_2, X_1]
- Maximum log likelihood is given by

$$R(\theta) = -\sum_{i=1}^{n} \log(P_\theta(x_i))$$

- In the case of multi-variate normal distribution, this method of estimation computes the mean and variance as follows:

$$\mathcal{P} : \{ N(\mu, \Sigma) : \mu \in \mathbb{R}^d, \Sigma \in S_d^+ \}$$

Data : $\{x_1, x_2, \cdots, x_N\}$ where $x_i \in \mathbb{R}^d$

ML estimates:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$$

$$\Sigma = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T$$

- Markov's inequality is given by the formula:

$$P(X \geq t) \leq \frac{\mu}{t}$$

- Chebyshev's inequality is given by the formula:

$$P(|X - \mu| \geq t) \leq \frac{\sigma^2}{t^2}$$

- WLLN is given by the formula:

$$P(\underline{|\bar{X}_n - \mu| \geq \epsilon}) \leq \frac{\sigma^2}{n\epsilon^2}$$