MLF

Weekwise Summary

Anant Kumar
(Course TA)

Contents

Week 3

The Four Fundamental Subspaces

1. Any $m \times n$ matrix A can be viewed as representing, in some bases, a *linear transformation* that maps vectors in $\mathbb{R}^n$ to vectors in $\mathbb{R}^m$.

2. The four fundamental subspaces of the matrix A, with rank r, are:

- The ***column space***, denoted as $C(A)$, is the space spanned by the columns of the matrix A taken as vectors:

  $$C(A) = \{Av \mid v \in \mathbb{R}^n\}$$

  Its dimension is r.

- The ***nullspace***, denoted as $N(A)$, is the set of vectors that are mapped to the zero vector by the matrix:

  $$N(A) = \{v \in \mathbb{R}^n \mid Av = 0\}$$

  It's dimension is $n - r$.

- The ***row space***, denoted as $C(A^T)$ is the space spanned by the rows of A. It is the same as the column space of A^T. It's dimension is also r.

- The ***left nullspace***. denoted as $N(A^T)$, is the nullspace of A^T. It's dimension is $m - r$.

3. Two vector spaces U and V are said to be *orthogonal* and denoted $U \perp V$, if for every pair of vectors $u \in U$ and $v \in V$, $u^T v = 0$.

4. The row space and the nullspace are subspaces of the domain set $\mathbb{R}^n$ of A and are *orthogonal*: $N(A) \perp C(A^T)$.

5. The column space and the left nullspace are the subspaces of the co-domain set $\mathbb{R}^m$ of the matrix A and are orthogonal: $C(A) \perp N(A^T)$.

6. Row reduce A to an *echelon form* U. Then:

- The *non-zero* rows of U are a basis for the *row space* of A.

- The *columns of* A that correspond to the *pivot columns of* U form a basis for the *column space* of A.
- The variables corresponding to the non-pivot columns are the free variables. The basis for the nullspace of A is obtained by vectors which are obtained by, in turn, assigning each free variable the value 1 and the other free variables the value 0.

Projections

1. The *projection* of vector b along the vector a is given by $\left(\frac{a^T b}{a^T a}\right) a$.

2. The *projection matrix* $\mathbb{P}$ associated with vector a is given by

$$\mathbb{P} = \frac{aa^T}{a^T a}$$

3. The projection of b along a can then be found as $\mathbb{P}b$.

4. The projection matrix associated with a matrix A is

$$\mathbb{P} = A(A^T A)^{-1} A^T$$

5. In case, $A^T A$ is not invertible, we need to replace $(A^T A)^{-1}$ with the ***pseudo inverse*** $(A^T A)^{\dagger}$.

6. For a matrix A, its projection matrix $\mathbb{P}$ projects any vector b into the *column space of* A, i.e. $\mathbb{P}b \in C(A)$ and can be thought of as the best approximation of b in the column space of A.

7. The projection matrix $\mathbb{P}$

- is ***symmetric***, that is, $\mathbb{P}^T = \mathbb{P}$,
- is ***idempotent***, that is, $\mathbb{P}^2 = \mathbb{P}$.

Conversely, any matrix satisfying the above two properties can be taken to be a projection matrix.

8. For any projection matrix $\mathbb{P}$, the matrix $(I - \mathbb{P})$ is an *orthogonal projection* matrix. If $\mathbb{P}$ projects a vector b onto the $C(A)$, then $(I - \mathbb{P})$ projects onto the subspace orthogonal to column space of A, that is, the left nullspace $N(A^T)$.

Ax = b: Least Square Solution

1. The matrix equation $Ax = b$ is solvable if and only if $b \in C(A)$.

2. In case $b \notin C(A)$, we can still find an approximate solution $\hat{x}$ that minimizes the squared norm $\|Ax - b\|^2$.

3. This *least square solution* can be obtained by solving the ***normal equations***:

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

and is given by

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

4. Finding the least square solution is equivalent to solving the equation

$$A\hat{x} = \mathbb{P}b$$

where $\mathbb{P}$ is the projection matrix of A.

Week 4

Linear & Polynomial Regression

1. Given a dataset

$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_n, y_n)\}$$

of data points $x_i \in \mathbb{R}^d$ and the corresponding labels $y_i \in \mathbb{R}$, a ***linear regression*** is a model that predicts a label $\hat{y}_i$ for each input vector x_i by a relation:

$$\hat{y}_i = x_i^T w + b, \quad i = 1, 2, \ldots, n$$

where $w = [w_1\ w_2\ \ldots\ w_d]^T$ is a d dimensional *weight* vector.

2. Introduce a constant feature 1 to each of the data point and construct a *data matrix* A of size $n \times (d+1)$, where each row is one data-point (the last column being all ones):

$$A = \begin{bmatrix} — & x_1^T & — & 1 \\ — & x_2^T & — & 1 \\ & \vdots & & 1 \\ — & x_n^T & — & 1 \end{bmatrix}$$

3. Similarly, redefine the weight vector to a $(d+1)$-dimensional vector:

$$w = [w_1\ w_2\ \ldots\ w_d\ b]^T$$

4. With these modifications, the predictions for all the n data-points can be written in a combined form as Aw.

5. If $y = [y_1\ y_2\ \ldots\ y_n]^T$ be the $n \times 1$ column vector of the actual labels, the problem is to find the best (in the least square sense) solution w of the equation

$$Aw = y$$

6. The optimum solution is obtained by solving the *normal equation*:

$$A^T A w = A^T y$$

7. In case, A^TA is invertible, the optimum solution

$$\hat{w} = (A^TA)^{-1}A^Ty$$

8. In case, A^TA is not invertible, we need to replace $(A^TA)^{-1}$ with the ***pseudo inverse*** $(A^TA)^{\dagger}$.
9. **Polynomial regression**: Transform the features of the data-points via a feature map ϕ which takes as input the features and produces all possible powers and combination of powers of features upto degree p and then perform linear regression in the transformed feature space. The result is same as that of the linear regression except for the fact that the data matrix A now contains as its rows $\phi(x_i)^T$, x_i being the i-th data point.
10. **Linear regression with regularization**: The loss function for the *regularized linear regression*, also known as the ***ridge regression*** is given by

$$L(w) = \frac{1}{2}\left(\|Ax - y\|^2 + \lambda\|w\|^2\right)$$

Minimizing the loss function for the optimum weight $\hat{w}$ leads to solving the equation

$$(A^TA + \lambda I)\hat{w} = A^Ty$$

which gives

$$\hat{w} = (A^TA + \lambda I)^{-1}A^Ty$$

- Too small value of λ leads to ***overfitting***. (It is almost no regularization)
- Too large value of λ leads to ***underfitting***.

Eigenvalues & Eigenvectors

1. Given a *square* matrix A, an ***eigenvalue*** λ is a scalar so that for some vector x, the equation

$$Ax = \lambda x$$

is satisfied. The corresponding vector x is known as an ***eigenvector***.

2. The eigenvalues can be obtained by solving the ***characteristic equation***

$$\det(A - \lambda I) = 0$$

For a matrix A of size $n \times n$, this is a polynomial equation in λ of degree n. Its n roots are the eigenvalues of A.

3. The eigenvectors are the solutions of the equation

$$(A - \lambda I)x = 0$$

That is to say that the eigenvectors lie in the null space of the matrix $(A - \lambda I)$, for each λ.

4. Geometrically, eigenvectors are vectors whose direction doesn't change when they are multiplied by A. They are only *stretched* by a stretching factor λ.
5. For matrices that are diagonal, upper triangular or lower triangular, the eigenvalues are the entries on the main diagonal.
6. Eigenvalues of ***similar*** matrices are exactly the same.
7. If λ is an eigenvalue of A with an eigenvector v, then λ^k is an eigenvalue of A^k with the same eigenvector v. In addition, if A is invertible, then λ^{-1} (that is $1/\lambda$) is an eigenvalue of A^{-1} with the same eigenvector.
8. Eigenvalues of A^T are same as those of A. However, the eigenvectors are ***not*** the same.
9. If $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the n eigenvalues of A, then

$$\begin{aligned} \lambda_1 + \lambda_2 + \ldots + \lambda_n &= \text{trace}(A) \\ \lambda_1 \lambda_2 \cdots \lambda_n &= \det(A) \end{aligned}$$

10. A ***symmetric*** matrix has all its eigenvalues ***real***.
11. Eigenvectors associated with ***distinct*** eigenvalues are ***independent***.

Diagonalization

1. A square matrix A is said to be ***diagonalizable*** if there exists an *invertible* matrix P such

$$P^{-1}AP = \Lambda,$$

a diagonal matrix.

2. For a square matrix A of size $n \times n$, if the n *eigenpairs* are (λ_1, v_1), (λ_2, v_2), ..., (λ_n, v_n), form a matrix P be formed by taking as its columns the vectors v_1, v_2, ..., v_n, respectively:

$$P = \begin{bmatrix} | & | & \cdots & | \\ v_1 & v_2 & \cdots & v_n \\ | & | & \cdots & | \end{bmatrix}$$

This matrix P ***diagonalizes*** the matrix A, that is, the product $P^{-1}AP$ is a diagonal matrix whose diagonal entries are the corresponding eigenvalues:

$$P^{-1}AP = \Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$$

3. Conversely, given matrix P and the diagonal matrix Λ, we can construct the matrix A as

$$A = P\Lambda P^{-1}$$

4. The diagonalizing matrix P is ***not unique***.

5. Not all matrices possess n linearly independent eigenvectors, so ***not all matrices are diagonalizable***.

6. If A is diagonalized by P, so that $A = P\Lambda P^{-1}$, then for any *integer* k:

$$A^k = P\Lambda^k P^{-1}$$

7. A real matrix Q is said to be ***orthogonal*** if and only if

$$Q^T Q = I, \text{ which means that } Q^{-1} = Q^T.$$

8. The columns of an orthogonal matrix are ***orthonormal***.

9. If A is *real and symmetric matrix* ($A^T = A$), then:

- Eigenvalues of A are real.

- Eigenvectors associated with *distinct* eigenvalues are ***orthogonal***.
- A is ***orthogonally diagonalizable***, that is, $\exists$ an orthogonal matrix Q such that $A = Q\Lambda Q^T$ for a diagonal matrix Λ. The matrix Q contains, as its columns, the normalized eigenvectors of A.

Week 5

Complex Numbers & Complex Vectors

1. A complex number is of the form $z = a + ib$, where $a, b \in \mathbb{R}$ and $i = \sqrt{-1}$ is the imaginary unit. The set of complex numbers is $\mathbb{C}$.

2. For every complex number z, it's modulus $|z| = \sqrt{a^2 + b^2}$ and $\arg(z) = \tan^{-1} \dfrac{b}{a}$.

3. For $z = a + ib$, the complex conjugate is obtained by replacing i with $-i$, that is $\bar{z} = a - ib$.

4. Complex vector: An n-dimensional complex vector is an $n \times 1$ matrix whose entries are complex numbers:

$$x = [x_1 \; x_2 \; \ldots \; x_n]^T \quad \text{for } x_i \in \mathbb{C}$$

This complex vector $x \in \mathbb{C}^n$.

5. Inner product: For complex vectors $x, y \in \mathbb{C}^n$, the inner product of x and y is defined as $\langle x, y \rangle = \bar{x}^T y$. Notice, that it is not commutative. Infact, $\langle y, x \rangle = \overline{\langle x, y \rangle}$.

6. Norm of a complex vector: For the vector

$$x = [x_1 \; x_2 \; \ldots \; x_n]^T \in \mathbb{C}^n,$$

the norm is defined as

$$\|x\| = \sqrt{\langle x, x \rangle}$$

which gives the working rule as

$$\|x\|^2 = |x_1|^2 + |x_2|^2 + \ldots + |x_n|^2$$

7. For a square complex matrix A, its ***conjugate transpose***, denoted by A^* is obtained by taking the complex conjugate of every element and then transposing the matrix.

Hermitian & Unitary Matrices

1. A square complex matrix A is said to be ***Hermitian***, if and only if $A^* = A$. (The corresponding real analogue would be *symmetric* matrix.)
2. In a Hermitian matrix, the diagonal entries are purely real and positions that are symmetrical with respect to the main diagonal contain entries that are complex conjugate of each other.
3. All eigenvalues of a Hermitian matrix are real.
4. Eigenvectors corresponding to distinct eigenvalues are *orthogonal*.
5. A square complex matrix U is said to be ***unitary***, if and only if $U^*U = I$ or $U^* = U^{-1}$ (The corresponding real analogue would be *orthogonal* matrix.)
6. The columns of a unitary matrix are ***orthonormal***.
7. ***Unitary matrices preserve lengths***: For any vector x and a unitary matrix U:
$$\|Ux\| = \|x\|$$
8. Eigenvalues of a unitary matrix have absolute value 1, that is if λ be an eigenvalue of a unitary matrix, then $|\lambda| = 1$.
9. Eigenvectors of a unitary matrix, corresponding to distinct eigenvalues are *orthogonal*.
10. ***Schur's theorem:*** Any $n \times n$ matrix A is similar to an upper triangular matrix, i.e. $\exists$ an *upper triangular* matrix T and a unitary matrix U so that $A = UTU^*$.
11. **Spectral theorem:** A Hermitian matrix is ***unitarily diagonalizable***. If A is Hermitian, $\exists$ a unitary matrix U such that $A = U\Lambda U^*$ for a diagonal matrix Λ. The matrix U contains, as its columns, the normalized eigenvectors of U, and Λ contains, along the main diagonal, the corresponding eigenvalues.

Week 6

Singular Value Decomposition

1. For any real $m \times n$ matrix A, the matrices A^TA and AA^T have the ***same non-zero*** eigenvalues.

2. The eigenvalues of A^TA (or those of AA^T) are ***non-negative***. In particular, the non-zero eigenvalues must be ***strictly positive***.

3. Let these positive eigenvalues be λ_1, λ_2, ..., λ_r (where r is the rank of matrix A). Then, the ***singular values*** of the matrix A are the numbers

$$\begin{aligned} \sigma_1 &= \sqrt{\lambda_1} \\ \sigma_2 &= \sqrt{\lambda_2} \\ &\vdots \\ \sigma_r &= \sqrt{\lambda_r} \end{aligned}$$

4. Work out the eigenvectors of A^TA associated with the eigenvalues λ_1, λ_2, ..., λ_r. Let them be v_1, v_2, ..., v_r. These are referred to as the ***right singular vectors*** of the matrix A. Now extend this set, using Gram-Schmidt to an orthonormal set

$$\{v_1, v_2, \ldots, v_r, v_{r+1}, \ldots, v_n\}$$

Construct the matrix V with these vectors as columns:

$$V = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \ldots & v_n \\ | & | & & | \end{bmatrix}$$

5. Work out the eigenvectors of AA^T associated with the eigenvalues λ_1, λ_2, ..., λ_r. Let them be u_1, u_2, ..., u_r. These are referred to as the ***left singular vectors*** of the matrix A. Now extend this set, using Gram-Schimdt to an orthonormal set

$$\{u_1, u_2, \ldots, u_r, u_{r+1}, \ldots, u_m\}$$

Construct the matrix U with these vectors as columns:

$$U = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \dots & u_m \\ | & | & & | \end{bmatrix}$$

6. The SVD of A is then

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

where Σ is an $m \times n$ matrix having the structure:

$$\Sigma = \begin{bmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \ddots & & \\ & & & \sigma_r & \\ & & & & 0 \end{bmatrix}_{m \times n}$$

7. In practice, however, we needn't workout the eigenvectors for both A^TA and AA^T. It is useful to do it only for one them. If $m < n$, then the size of AA^T is smaller. Work out the left singular vectors u_i's and then obtain r of the right singular vectors v_i's using

$$v_i = \frac{1}{\sigma_i} A^T u_i$$

Then extend the set $\{v_1, v_2, \dots, v_r\}$ to have n orthonormal vectors using Gram Schmidt orthogonalization.

8. If $n < m$, then the size of A^TA is smaller. Work out the right singular vectors v_i's and then obtain r of the left singular vectors u_i's using

$$u_i = \frac{1}{\sigma_i} A u_i$$

Then extend the set $\{u_1, u_2, \dots, u_r\}$ to have m orthonormal vectors using Gram Schmidt orthogonalization.

9. It is instructive to note that the columns of U and V give the orthonormal bases for all four fundamental subspaces of an $m \times n$ matrix A:

first	r	columns of U :	**column space** of A
last	$m-r$	columns of U :	**left nullspace** of A
first	r	columns of V :	**row space** of A
last	$n-r$	columns of V :	**nullspace** of A

10. We can write A as the sum of rank 1 matrices in the following way:

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_r u_r v_r^T$$

11. Restricting the above sum to k terms ($k \leq r$) gives the rank-k approximation of the matrix A:

$$A_k = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_k u_k v_k^T, \quad (k \leq r)$$

Specifically, the rank-1 approximation of matrix A is

$$A_1 = \sigma_1 u_1 v_1^T$$

12. A geometrical aspect of SVD: Treat the rows of an $m \times n$ matrix A as m points in n-dimensional space. Then for any column vector v such that $\|v\| = 1$, the quantity $\|Av\|^2$ can be interpreted as the ***sum of the squared lengths of the projections of the rows of*** A ***along*** v. And hence, the ***best fit line through the origin*** is the one ***maximizing*** $\|Av\|^2$.

- The ***first singular vector***, v_1, of A, is the direction of the best fit line through the origin for the m points in the n-dimensional space that are the rows of the matrix A. Thus,

$$v_1 = \underset{\|v\|=1}{\arg\max} \|Av\|$$

- The value $\sigma_1 = \|Av_1\|$ is the ***first singular value*** of A. Notice that σ_1^2 is the sum of the projections of the points on the line determined by v_1.
- Similarly, the second singular vector

$$v_2 = \underset{v \perp v_1, \|v\|=1}{\arg\max} \|Av\|$$

and so on.

13. If the SVD of $A = U\Sigma V^T$, then the ***pseudoinverse*** of A is

$$A^\dagger = V\Sigma^\dagger U^T$$

where

$$\Sigma^\dagger = \begin{bmatrix} 1/\sigma_1 & & & & \\ & 1/\sigma_2 & & & \\ & & \ddots & & \\ & & & 1/\sigma_r & \\ & & & & 0 \end{bmatrix}_{n\times m}$$

Positive Definite Matrices

1. For any $n \times n$ real, symmetric matrix A, the product $x^T Ax$, where $x \in \mathbb{R}^n$, is a ***pure quadratic form***.

2. The real and symmetric matrix A is said to be ***positive definite*** if for every *nonzero* vector $x \in \mathbb{R}^n$, $x^T Ax > 0$.

3. Each of the following tests is a *necessary and sufficient condition* for the real symmetric matrix A to be ***positive definite***:

(I) $x^T Ax > 0$ for all non-zero vectors x.

(II) All the eigenvalues of A are positive: $\lambda_i > 0$.

(III) All the upper left submatrices A_k have positive determinants: $\det(A_k) > 0$.

(IV) In the echelon form (without row exchanges) of the matrix A, all the pivots entries are positive.

4. Each of the following tests is a *necessary and sufficient condition* for the real symmetric matrix A to be ***positive semidefinite***:

(I) $x^T Ax \geq 0$ for all vectors x.

(II) All the eigenvalues of A are nonnegative: $\lambda_i \geq 0$.

(III) All the upper left submatrices A_k have nonnegative determinants: $\det(A_k) \geq 0$.

(IV) In the echelon form (without row exchanges) of the matrix A, all the pivots entries are nonnegative.

5. For checking for negative definiteness (or negative semidefiniteness) of real symmetric matrix A, we can check for *positive* definiteness (or *positive* semidefiniteness) of $-A$.

6. In case, the real symmetric matrix A is neither positive nor negative definite (or semidefinite), then it is an ***indefinite*** matrix.

7. If A is a positive definite matrix, then so are A^2, A^3, A^4, ..., and A^{-1}.

8. If A and B are positive definite, then so is $A + B$.

Week 7

Principal Component Analysis

1. The ***Principal Component Analysis*** (PCA) tries to find out the directions along which the *projected* data has ***maximum variance***, which are the same directions along which the ***minimum reconstruction error*** occurs.

2. In order to perform PCA on a dataset $\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$ where each of the data point is a d-dimensional vector, the following steps can be done:

I. Find the mean vector $\bar{x}$ of the dataset:

$$\bar{x} = \frac{x_1 + x_2 + \ldots + x_n}{n}$$

II. Form the data matrix:

$$X = \begin{bmatrix} | & | & & | \\ x_1 - \bar{x} & x_2 - \bar{x} & \ldots & x_n - \bar{x} \\ | & | & & | \end{bmatrix}$$

III. Obtain the ***covariance matrix***:

$$C = \frac{1}{n} X X^T = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T$$

IV. Find the eigenvalues of C and arrange them in the descending order:

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$$

and the associated *unit* eigenvectors u_1, u_2, …, u_d.

V. The eigenvector u_1 associated with the greatest eigenvalue λ_1 is referred to as the ***first principal component***, the vector u_2 associated with the eigenvalue λ_2 is called the ***second principal component*** and so on.

3. The principal components are ***orthogonal*** to each other.

4. The projection z_i of the data-point x_i, onto an m-dimensional subspace, where $m < d$, is obtained as follows:

$$z_i = \sum_{j=1}^{m} (x_i^T u_j) u_j + \sum_{j=m+1}^{d} (\bar{x}^T u_j) u_j$$

for each $i = 1, 2, \ldots n$.

5. The projection z_i of the datapoint x_i, onto the *first principal component* u_1, is obtained as follows:

$$z_i = (x_i^T u_1) u_1 + \sum_{j=2}^{d} (\bar{x}^T u_j) u_j$$

6. The ***reconstruction error*** is

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

7. The ***variance*** of the *projected* data along the i-th principal component is equal to the corresponding eigenvalue λ_i.

8. If the dataset is very ***high dimensional***, so that $d \gg n$, PCA can still be efficiently implementing by considering

$$C = \frac{1}{n} X^T X$$

which is of size $n \times n$, instead of the $d \times d$ matrix $C = \frac{1}{n} X X^T$.

Week 8

Introduction to Optimization

1. Three pillars on which Machine Learning stands on: **Linear Algebra**, **Optimization**, and **Probability & Statistics**.

2. An optimization problem can be either

(i) an ***unconstrained optimization***, or

(ii) a ***constrained optimization*** problem.

3. The general form of a constrained optimization problem is to *minimize* an ***objective function*** $f(x)$ subject to some ***inequality constraints*** $g_i(x) \leq 0$ (for $i = 1, 2, \ldots, k$) and some ***equality constraints*** $h_j(x) = 0$ (for $j = 1, 2, \ldots, \ell$):

$$\begin{aligned} & \min_x \ f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad i = 1, 2 \ldots, k \\ \text{and} \quad & h_j(x) = 0, \quad j = 1, 2 \ldots, \ell \end{aligned}$$

Here, x is a d-dimensional vector whose components are real numbers: $x \in \mathbb{R}^d$ and f is a real valued function so that $f(x) \in \mathbb{R}$.

4. The general form of an ***unconstrained optimization*** is to *minimize* an objective function where x is free to vary over the entire $\mathbb{R}^d$.

5. Maximizing $f(x)$ is equivalent to minimizing $-f(x)$.

6. A point/vector x^* is said to be a ***local minimizer*** of $f(x)$ if $f(x^*) \leq f(x)$ for all x is a ***neighbourhood*** of x^*.

7. A point x^* is said to be a ***global minimizer*** of $f(x)$ if $f(x^*) \leq f(x)$ for all $x \in \text{dom}\ (f)$.

8. Given a real-valued function f, the notation $\arg\min f(x)$ denotes the argument (a point in the domain of f) that minimizes the function f, assuming such a point is unique.

Unconstrained Optimization

1. If for a point x^* in the domain of f, it is found that

(i) $\nabla f(x^*) = 0$, and

(ii) the $\det H(x^*) > 0$, where $H(x)$ is the ***Hessian*** matrix evaluated at x,

then x^* is a local minimizer of f.

2. The direction of $\nabla f(x)$ is the direction of the ***steepest ascent***, while the direction of $-\nabla f(x)$ is the direction of the ***steepest descent*** at some point x in the domain of f.

3. Gradient Descent: For unconstrained optimization, an important method to find a local minimizer is the gradient descent method (also referred as the *steepest descent method*) which, starting from an initial point x_0, iteratively moves through a sequence of points x_0, x_1, x_2, ... in such a way that finally a local minimizer is found as the limit of the sequence $\{x_0, x_1, x_2, \ldots\}$.

4. The point x_k obtained after the k-th iteration is obtained as

$$x_k = x_{k-1} - \eta_k \nabla f(x_{k-1})$$

where η_k is the ***step size*** chosen during the k-th iteration.

5. A suitable choice of η_k is important for the method of gradient descent to converge. If η_k is too large, the algorithm might end up oscillating about the local minimizer; if η_k is too small, it process might become too slow.

6. A suitable choice for the step size during the k-th step can be found as

$$\eta_k = \arg\min_{\eta \geq 0} f\left(x_{k-1} - \eta \nabla f(x_{k-1})\right)$$

7. In practice, we can work with a *constant* step size chosen suitably.

Week 9

Constrained Optimization: One Inequality

1. The problem is to find

$$\begin{aligned} & \min_x && f(x) \\ \text{subject to} \quad & && g(x) \leq 0 \end{aligned}$$

Note that $x \in \mathbb{R}^d$.

2. A direction characterized by a (unit) vector d is said to be a ***descent direction*** at a point x if

$$d^T \nabla f(x) < 0$$

3. The value of the objective function will necessarily ***decrease*** if one moves along a descent direction. As such, a descent direction is a property of the objective function $f(x)$.

4. At a point x in the constraint region $g(x) \leq 0$, a direction characterized by a unit vector d is said to be a ***feasible direction*** if there exists a $\eta_0 > 0$ such that for all $\eta \in (0, \eta_0]$, the point $x + \eta d$ is still in the constraint region, i.e. $g(x + \eta d) \leq 0$. In simple terms, a feasible direction at a point is a direction along which we can move by some step size so that we still remain in the constraint region.

5. The feasible direction is a property of the constraint function $g(x)$.

6. **Necessary conditions:** If x^* is an *optimal* point that minimizes the objective function f, then at the point x^*:

(i) ***no descent direction should be a feasible direction***,

(ii) $\nabla f(x^*) = -\lambda \nabla g(x^*)$ for some $\lambda \geq 0$, and

(iii) $\lambda g(x^*) = 0$ for the same λ as above.

Constrained Optimization: One Equality

1. The problem is to find

$$\begin{aligned} & \min_x && f(x) \\ \text{subject to} \quad & && g(x) = 0 \end{aligned}$$

Note that $x \in \mathbb{R}^d$.

2. Necessary conditions: If x^* is an *optimal* point that minimizes the objective function f, then at the point x^*:

(i) $g(x^*) = 0$

(ii) $d^T \nabla g(x^*) = 0$ for any feasible direction d,

(iii) $\nabla f(x^*) = -\lambda \nabla g(x^*)$ for some arbitrary $\lambda \in \mathbb{R}$.

Notice that in this case λ is allowed to take either positive or negative values. λ is referred to as the ***Lagrange multiplier*** and equation (iii) above is the *Lagrange equation.*

3. In general, it may ***not*** be feasible to solve the system of equations that satisfy the Lagrange equations.

4. In case the constraint region $\Omega = \{x \mid g(x) \leq 0\}$ is ***convex***, the ***projected gradient descent*** algorithm can be applied and the optimum x^* can be found as the limit of sequence $\{x_0, x_1, x_2, \ldots\}$, where

$$x_k = \prod(x_{k-1} - \eta_k \nabla f(x_{k-1})), \quad k = 1, 2, \ldots$$

5. The projection operator $\prod$ projects any vector onto Ω:

$$\prod(v) = \min_{x \in \Omega} \|v - x\|^2$$

Convex Sets

1. A set $S \subset \mathbb{R}^d$ is a *convex set* if *for every* pair of points x_1 and x_2 which are in S, the line segment joining these points is also entirely in S, i.e.

$$\forall\ x_1, x_2 \in S \Rightarrow \lambda x_1 + (1 - \lambda) x_2 \in S,\ \lambda \in [0, 1]$$

2. Examples of convex sets include:

- the empty set
- a set consisting of a single point
- a line or a line segment
- a subspace
- a hyperplane
- a linear variety (a translation of a subspace)
- a half-space
- $\mathbb{R}^d$

3. Convex sets in $\mathbb{R}^d$ have the following properties:

(a) If S is a convex set and β is any real number, then the set

$$\beta S = \{x \mid x = \beta v,\ v \in S\}$$

is also convex.

(b) If S_1 and S_2 are convex sets, then the set

$$S_1 + S_2 = \{x \mid x = v_1 + v_2,\ v_1 \in S_1, v_2 \in S_2\}$$

is also convex.

(c) The intersection of any collection of convex sets is also convex.

4. Convex Combination: Let $S = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^d$. Then a *convex combination* of the elements of S is a linear combination of the elements of S so that the multiplying coefficients are all *non-negative* and sum up to 1. That is to say, any $z \in \mathbb{R}^d$ is a convex combination of the elements of S if $\exists\ \lambda_1, \lambda_2, \ldots, \lambda_n$ so that

$$\lambda_i \geq 0, \quad i = 1, 2, \ldots, n$$

$$\sum_{i=1}^{n} \lambda_i = 1, \text{ and}$$

$$z = \lambda_1 x_1 + \lambda_2 x_2 + \ldots + \lambda_n x_n$$

5. **Convex Hull:** Let $S = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^d$. The *convex hull* of the set S, denoted $\mathrm{CH}(S)$ is the set of all possible convex combinations of the elements of S:

$$CH(S) = \{z \mid z = \sum_{i=1}^{n} \lambda_i x_i, \text{ for } \lambda_1, \lambda_2 \ldots, \lambda_n \geq 0, \sum_{i=1}^{n} \lambda_i = 1\}$$

6. The convex hull of the set S can also be defined as the intersection of all convex sets that contain the set S.

Convex Functions

1. **Epigraph of a function:** Let $X \subset \mathbb{R}^d$ and $Y \subset \mathbb{R}$ and consider a function $f : X \to Y$. It's ***epigraph*** is then the set

$$\mathrm{epi}(f) = \left\{ \begin{bmatrix} x \\ z \end{bmatrix} \subset X \times Y \mid z \geq f(x) \right\}$$

   That is, the epigraph of f consists of all points "on or above" the graph of f.

2. A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be a ***convex function*** if *any* of the following holds:

   I. the $\mathrm{epi}(f) \subset \mathbb{R}^{d+1}$ is a convex set.

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

$$f\left(\lambda x_1 + (1-\lambda)x_2\right) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$

   III. for any point y in a neighbourhood of x,

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

   assuming f to be differentiable.

   IV. the ***Hessian*** of f is a ***positive semi-definite*** matrix.

Properties of Convex Functions

1. If f is a convex function, then all *local minima* of f are also *global minima*.

2. The set of all global minima of a convex function is a convex set.

3. If $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable, convex function, then for a point $x^* \in \mathbb{R}^d$ to be a *global minimum* of f, the ***necessary and sufficient*** is $\nabla f(x^*) = 0$.

4. If $f, g : \mathbb{R}^d \to \mathbb{R}$ be convex functions, then their ***sum*** $f(x) + g(x)$ is also a convex function.

5. If $f : \mathbb{R} \to \mathbb{R}$ be a ***convex*** and ***non-decreasing*** function, and $g : \mathbb{R}^d \to \mathbb{R}$ be a ***convex*** function, then their ***composition*** $f(g(x))$ is a convex function.

6. If $f : \mathbb{R} \to \mathbb{R}$ be a ***convex*** and ***non-increasing*** function, and $g : \mathbb{R}^d \to \mathbb{R}$ be a ***concave*** function, then their ***composition*** $f(g(x))$ is a convex function.

7. The composition of two convex functions ***need not*** be convex.

8. A function f is concave if and only if $-f$ is convex.

Week 10

Optimization: An Application in ML

1. Given a dataset

$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_n, y_n)\}$$

of data points $x_i \in \mathbb{R}^d$ and the corresponding labels $y_i \in \mathbb{R}$, a common Machine Learning problem is to find a model function $h : \mathbb{R}^d \to \mathbb{R}$ that can predict the label of a test data point as

$$\hat{y} = h(x_{\text{test}})$$

2. In ***linear regression***, we look for a linear function

$$h_w(x) = w^T x$$

for some $w \in \mathbb{R}^d$.

3. The weight w corresponding to the "best" model is determined from the given data $\mathcal{D}$.

4. A suitable *performance measure* is the ***sum of squares error***:

$$f(w) = \frac{1}{2} \sum_{i=1}^{n} \left(w^T x_i - y_i\right)^2$$

5. The specific goal of linear regression is to select a w so that $f(w)$ is minimized:

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

6. The loss function $f(w)$ is a convex function since it is the sum of convex functions. Hence, the global minimum is obtained by setting $\nabla_w f(w) = 0$

7. Define the matrices:

$$X = \begin{bmatrix} — & x_1^T & — \\ — & x_2^T & — \\ & \vdots & \\ — & x_n^T & — \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

The loss function can then be written in the matrix form as

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

which can further be transformed as

$$f(w) = \frac{1}{2}w^T(X^TX)w - w^T(X^Ty) + \frac{1}{2}y^Ty$$

8. The gradient of the loss function equated to zero:

$$\nabla f(w) = (X^TX)w - X^Ty = 0$$

9. The optimal weight w^* is the solution of the ***normal*** equation:

$$(X^TX)w^* = X^Ty$$

which gives

$$w^* = (X^TX)^\dagger X^Ty$$

10. For $x_i \in \mathbb{R}^d$, the "inverse" computation requires $O(d^3)$ number of operations, which may be computationally expensive for large d.

11. The gradient descent method can be used as it doesn't involve inverse calculation: Starting from an initial guess w_0, iteratively compute the weight in the k-th iteration, using the step size η_k, as

$$w_k = w_{k-1} - \eta_k \nabla f(w_{k-1})$$

which, upon plugging the gradient, becomes

$$w_k = w_{k-1} - \eta_k \left((X^TX)w_{k-1} - X^Ty\right)$$

12. In case, the number of points n is also very large, we can use the ***stochastic gradient descent*** (SGD) in which, at each iteration, the exact gradient ∇f is replaced by a cheap, *unbiased estimator* of the gradient.
13. In a particular instantiation of the SGD, called the *mini-batch* SGD, at each iteration, the exact gradient is replaced by the *average* gradient of a *uniformly sampled subset* of the training data.
14. The SGD has a *slower convergence rate* compared to the actual gradient descent.
15. Both the gradient descent and the stochastic gradient descent can be also applied efficiently to non-convex optimization problems as well.

Revisiting Optimization

Unconstrained Optimization

If f is *convex*, the ***global minimum*** can be obtained by solving $\nabla f(x) = 0$.

Constrained Optimization

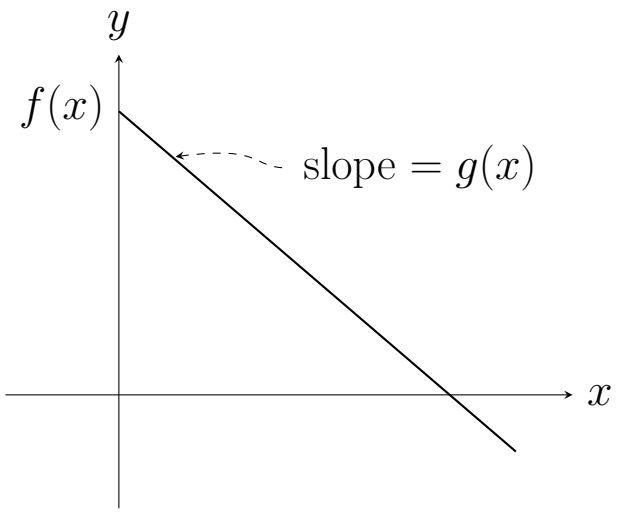
1. The problem:

$$\begin{aligned} & \min_{x} && f(x) \\ \text{subject to} \quad & && g(x) \leq 0 \end{aligned}$$

   where $f, g : \mathbb{R}^d \to \mathbb{R}$.
2. Define the ***Lagrangian***:

$$\mathcal{L}(x, \lambda) = f(x) + \lambda g(x)$$

   where $x \in \mathbb{R}^d$ and $\lambda \in \mathbb{R}$.
3. Viewed as a function of λ, this is a *linear* function. Since, $g(x) \leq 0$, the gradient of the line is negative, so the graph is a downward slanting line:



4. As a result:

$$f(x) = \max_{\lambda \geq 0} \mathcal{L}(x, \lambda)$$

5. Hence, the original problem is equivalent to finding the *minimum* value of $\mathcal{L}(x, \lambda)$ for $\lambda \geq 0$:

$$\begin{array}{rl} \min\limits_{x} & f(x) \\ \text{subject to} & g(x) \leq 0 \end{array} \equiv \min_{x} \left(\max_{\lambda \geq 0} \mathcal{L}(x, \lambda) \right)$$

6. $\min\limits_{x} \left(\max\limits_{\lambda \geq 0} \mathcal{L}(x, \lambda) \right)$ is known as the ***primal form*** of the optimization problem.

7. This min max problem is usually hard to solve.

8. The corresponding ***dual form*** of the optimization problem is the max min form:

$$\max_{\lambda \geq 0} \left(\min_{x} \mathcal{L}(x, \lambda) \right)$$

9. Thus

$$\boxed{\min_{x} \left(\max_{\lambda \geq 0} \mathcal{L}(x, \lambda) \right)} \longleftrightarrow \boxed{\max_{\lambda \geq 0} \left(\min_{x} \mathcal{L}(x, \lambda) \right)}$$

PRIMAL DUAL

10. The ***primal optimum*** x^* is the point at which the primal form attains the *minimum* $f(x^*)$.

11. Let $\ell(\lambda) = \min_x \mathcal{L}(x, \lambda)$. The ***dual optimum*** λ^* is the value at which the dual form attains its *maximum* $\ell(\lambda^*)$.

12. In general, the value at the primal optimum may not be equal to the value at the dual optimum.

13. The principle of ***weak duality*** says that the value at the dual optimum is less than or equal to the value at the primal optimum:

$$\ell(\lambda^*) \leq f(x^*)$$

14. In case, the objective function f and the inequality constraints are *convex* functions, the value at the dual optimum is equal to the value at the primal optimum. This is referred to as ***strong duality***.

15. Let the objective function f and the inequality constraint function g be *convex* functions. Then strong duality holds with some *regularizing conditions*. Let the primal optimum be x^* and the dual optimum be λ^*. Then the following conditions hold true:

(a) **Stationarity condition:**

$$\nabla f(x^*) + \lambda^* \nabla g(x^*) = 0$$

(b) **Complimentary slackness condition:**

$$\lambda^* \; g(x^*) = 0$$

(c) **Primal feasibility condition:**

$$g(x^*) \leq 0$$

(d) **Dual feasibility condition:**

$$\lambda^* \geq 0$$

These are referred to as the **KKT** conditions.

Karush-Kuhn-Tucker (KKT) Conditions

1. The generalized form of the constraint optimization problem:

$$\min_x \ f(x)$$

$$\text{subject to} \quad g_i(x) \leq 0, \quad i = 1, 2, \ldots, n$$

$$h_j(x) = 0, \ \ j = 1, 2, \ldots, m$$

where f, g_i, $h_j : \mathbb{R}^d \to \mathbb{R}$ for $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$.

2. The generalized *Lagrangian*:

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \sum_{i=1}^{n} \lambda_i g_i(x) + \sum_{j=1}^{m} \mu_j h_j(x)$$

where $x \in \mathbb{R}^d$, $\lambda = \begin{bmatrix} \lambda_1 & \lambda_2 & \ldots & \lambda_n \end{bmatrix}^T$, $\mu = \begin{bmatrix} \mu_1 & \mu_2 & \ldots & \mu_m \end{bmatrix}^T$.

3. The scalars λ_is are called ***KKT multipliers*** while the scalars μ_js are called the ***Lagrange multipliers***.

4. If x^* be a *local* minimizer of f, there exist λ_i^* and μ_j^* satisfying the following conditions referred to as the KKT conditions:

(a) **Stationarity condition:**

$$\nabla f(x^*) + \sum_{i=1}^{n} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{m} \mu_j^* \nabla h_j(x^*) = 0$$

(b) **Complimentary slackness condition:**

$$\lambda_i^* \ g_i(x^*) = 0, \quad i = 1, 2, \ldots, n$$

(c) **Primal feasibility condition:**

$$g_i(x^*) \leq 0 \quad i = 1, 2, \ldots, n$$

$$h_j(x^*) = 0 \ \ j = 1, 2, \ldots, m$$

(d) **Dual feasibility condition:**

$$\lambda_i^* \geq 0, \quad i = 1, 2, \ldots, n$$

5. If f and the inequality constraint functions g_i are *convex*, then the local minimizer x^* is also a *global* minimizer.

Support Vector Machines (SVM)

1. An important ML task is to find a ***linear classifier*** that has a large *geometric margin*, i.e., whose *decision boundary* is well separated from all the training data points.

2. Such a classifier is known as the ***Support Vector Machine*** or SVM for short.

3. Given a dataset

$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_n, y_n)\}$$

of data points $x_i \in \mathbb{R}^d$ and the corresponding labels $y_i \in \{-1, +1\}$.

4. The SVM is a linear model that predicts a label for a input x as

$$h_w(x) = \text{sign}(w^T x)$$

with as large ***geometric margin*** (distance between the decision boundary and the nearest data point) as possible.

5. The problem of finding the optimum weight w reduces to a ***quadratic optimization*** problem:

$$\min_w \frac{1}{2}\|w\|^2$$

$$\text{subject to} \quad 1 - y_i w^T x_i \leq 0, \;\; i = 1, 2, \ldots, n$$

6. Since the objective function is quadratic, it is a convex function. In addition, the inequality constraint function is linear and hence also convex. As a result, any local optimum will be global minimum.

Week 11

Continuous Random Variables

1. **Random variable:** A random variable X is a function that maps the ***sample space*** Ω to the set of real numbers $\mathbb{R}$. It assigns to each element $\omega \in \Omega$ one and only one value $X(\omega) = x$.
2. Let X be a random variable. Then its ***cumulative distribution function (cdf)*** is defined by

$$F_X(x) = P(X \leq x)$$

3. A random variable X is a ***continuous random variable*** if its cumulative distribution function $F_X(x)$ is a continuous function $\forall\ x \in \mathbb{R}$.
4. The ***probability density function (pdf)*** of a random variable X is a *non-negative* function $f_X(x)$ such that

$$F_X(x) = \int_{-\infty}^{x} f_X(x)\ \mathrm{d}x, \quad \forall\ x \in \mathbb{R}$$

5. A cdf $F_X(x)$ always has the following properties:
   (a) $\lim\limits_{x \to -\infty} F_X(x) = 0$
   (b) $\lim\limits_{x \to \infty} F_X(x) = 1$
   (c) $F_X(x)$ is a ***non-decreasing*** function.
6. If $f_X(x)$ is the pdf of a random variable X, then at the points where $f_X(x)$ is continuous,

$$\frac{\mathrm{d}F_X(x)}{\mathrm{d}x} = f_X(x)$$

7. A pdf $f_X(x)$ always has the following properties:
   (a) $f_X(x) \geq 0\ \forall\ x \in \mathbb{R}$
   (b) $\int_{\mathbb{R}} f_X(x)\ \mathrm{d}x = 1$, where $\int_A$ represents integration over the set A.

(c) If $A \subset \mathbb{R}$, then $P(A) = \int_A f_X(x)\ \mathrm{d}x$. In other words,

$$P(a \leq X \leq b) = \int_a^b f_X(x)\ \mathrm{d}x$$

8. The ***support*** of a random variable X is

$$\text{support } X = \{x \mid f_X(x) > 0\}$$

9. For a random variable X, its ***expectation*** or ***expected value*** is

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

often denoted as μ_X.

10. If X be a random variable and $g(X)$ be some function of the random variable, then

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f_X(x)\ \mathrm{d}x$$

11. ***Expectation is a linear operation:***

(a) $E[aX + b] = aE[x] + b$ for $a, b \in \mathbb{R}$

(b) $E[X_1 + X_2 + \ldots + X_n] = E[X_1] + E[X_2] + \ldots + E[X_n]$

12. For a random variable X, its ***variance***, denoted as σ_X^2, is

$$\sigma_X^2 = \mathrm{Var}(X) = E[(X - \mu_X)^2] = E[X^2] - (E[X])^2$$

so that

$$\mathrm{Var}(X) = \int_{-\infty}^{\infty} (x - \mu_X)^2 f_X(x)\ \mathrm{d}x$$

13. For a random variable X and $a, b \in \mathbb{R}$:

$$\mathrm{Var}(aX + b) = a^2 \mathrm{Var}(X)$$

14. If X is a continuous random variable, and A is the event that $a < X < b$ (where possibly $b \to \infty$ or $a \to -\infty$), then the ***conditional*** pdf of X given the event A is

$$f_{X|A}(x) = \begin{cases} \dfrac{f_X(x)}{P(A)} & a \leq x < b \\ 0 & \text{otherwise} \end{cases}$$

15. The ***conditional expectation*** of X given A is

$$E[X|A] = \int_{-\infty}^{\infty} x f_{X|A}(x)\, \mathrm{d}x$$

16. The ***conditional expectation*** of $g(X)$ given A is

$$E[g(X)|A] = \int_{-\infty}^{\infty} g(x) f_{X|A}(x)\, \mathrm{d}x$$

17. The ***conditional variance*** of X given A is

$$\mathrm{Var}(X|A) = E[X^2|A] - (E[X|A])^2$$

18. For a random variable X and any event A, the ***total expectation*** of X is

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

19. If $Y = g(X)$, where $g : \mathbb{R} \to \mathbb{R}$ is a ***strictly monotonous differentiable*** function so that g has a ***unique inverse*** and let $h = g^{-1}$. Then, the pdf of the random variable Y is given by

$$f_Y(y) = f_X(h(y))|h'(y)|$$

20. If $Y = g(X)$, where $g : \mathbb{R} \to \mathbb{R}$ is differentiable but g is ***not*** strictly monotonous. However, there is a partition of $\mathbb{R}$ into *disjoint* intervals, say I_1, I_2, ... in each of which g is strictly monotonous and let $g_k(x) = g(x)$ for $x \in I_k$. Then, the pdf of the random variable Y is given by

$$F_Y(y) = \sum_k f_X\left(g_k^{-1}(y)\right) \left| \frac{\mathrm{d}}{\mathrm{d}y} g_k^{-1}(y) \right|$$

Two Random Variables

1. Two random variables X and Y are ***jointly continuous*** if there exists a *non-negative* function $f_{X,Y} : \mathbb{R}^2 \to \mathbb{R}$, such that, for any set $A \subset \mathbb{R}^2$:

$$P\left((X,Y) \in A\right) = \iint_A f_{X,Y}(x,y)\, \mathrm{d}x\, \mathrm{d}y$$

The function $f_{X,Y}$ is called the ***joint probability density function*** of X and Y.

2. The joint pdf of X and Y must satisfy

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{X,Y}(x,y)\,\mathrm{d}x\,\mathrm{d}y = 1$$

3. The ***marginal*** pdf of X can be obtained from the joint pdf by integrating over all values of y:

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x,y)\,\mathrm{d}y, \quad \forall\, x$$

4. The ***marginal*** pdf of Y can be obtained from the joint pdf by integrating over all values of x:

$$f_Y(x) = \int_{-\infty}^{\infty} f_{X,Y}(x,y)\,\mathrm{d}x, \quad \forall\, y$$

5. If X and Y are jointly continuous, the ***conditional pdf*** of X given Y is

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

6. For two jointly continuous random variables X and Y:

(a) The expected value of X given $Y = y$ is

$$E[X|Y=y] = \int_{-\infty}^{\infty} x f_{X|Y}(x \mid y)\,\mathrm{d}x$$

(b) For a function $g(X)$ of X

$$E[g(X)|Y=y] = \int_{-\infty}^{\infty} g(x) f_{X|Y}(x \mid y)\,\mathrm{d}x$$

(c) The ***conditional variance*** of X given A is

$$\mathrm{Var}(X|Y=y) = E[X^2|Y=y] - (E[X|Y=y])^2$$

7. If two jointly continuous random variables are independent if and only if

$$f_{X|Y}(x|y) = f_X(x)$$

Equivalently, two jointly continuous random variables are independent if and only if

$$f_{X,Y}(x,y) = f_X(x)f_Y(y)$$

8. For independent random variables X and Y

$$E[XY] = E[X]E[Y]$$

9. If X and Y be independent random variables, then $\forall$ functions g and h

$$E[g(X)h(Y)] = E[g(X)]E[h(X)]$$

10. Special distributions of *independent* random variables: Let X and Y be two independent random variables, with pdf $f_X(x)$, $f_Y(x)$ and cdf $F_X(x)$, $F_Y(x)$, respectively.

(a) **Sum:** The pdf of $Z = X+Y$ is obtained using the ***convolution integral***:

$$f_Z(z) = \int_{-\infty}^{\infty} f_X(x)f_Y(z-x)\,\mathrm{d}x$$

(b) **Maximum:** The cdf of the function $Z = \max\{X, Y\}$ is

$$F_Z(z) = F_X(z)F_Y(z)$$

(c) **Minimum:** The cdf of the function $Z = \min\{X, Y\}$ is

$$F_Z(z) = 1 - (1 - F_X(z))\,(1 - F_Y(z))$$

11. If X and Y are random variables with expectations μ_X and μ_Y respectively, then their ***covariance*** is defined to be

$$\mathrm{Cov}(X,Y) = E[(X-\mu_X)(Y-\mu_Y)] = E[XY] - \mu_X\mu_Y$$

12. The ***correlation coefficient*** of random variables X and Y is defined to be

$$\rho = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}}$$

13. If the random variables X and Y are independent, then

$$\mathrm{Cov}(X, Y) = 0$$

(Note: *the converse need not be true.*)

14. Transformed random variables: Suppose the random variables X_1, X_2 are *transformed* to random variables Y_1, Y_2 by some ***one to one*** mapping:

$$\begin{aligned} Y_1 &= u_1(X_1, X_2), \\ Y_2 &= u_2(X_1, X_2) \end{aligned}$$

which can be inverted to

$$\begin{aligned} X_1 &= w_1(Y_1, Y_2), \\ X_2 &= w_2(Y_1, Y_2) \end{aligned}$$

Let J be the ***Jacobian*** of this transformation:

$$J = \det \begin{bmatrix} \dfrac{\partial w_1}{\partial y_1} & \dfrac{\partial w_1}{\partial y_2} \\ \dfrac{\partial w_2}{\partial y_1} & \dfrac{\partial w_2}{\partial y_2} \end{bmatrix}$$

Then the joint pdf of Y_1, Y_2 is

$$f_{Y_1,Y_2}(y_1, y_2) = f_{X_1,X_2}\left(w_1(y_1, y_2), w_2(y_1, y_2)\right) |J|$$

where $|\cdot|$ represents the absolute value.

Week 12

Random Vectors

1. A ***random vector*** X is a vector, each of whose element is a random variable:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}$$

2. The ***mean vector*** of the random vector X is

$$E[X] = \begin{bmatrix} E[X_1] \\ E[X_2] \\ \vdots \\ E[X_n] \end{bmatrix}$$

3. A ***random matrix*** M is a matrix, each of whose element is a random variable:

$$X = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1n} \\ X_{21} & X_{22} & \dots & X_{2n} \\ \vdots & \cdots & \vdots & \vdots \\ X_{m1} & X_{m2} & \dots & X_{mn} \end{bmatrix} = [X_{ij}]_{m \times n}$$

4. The ***mean matrix*** of the random matrix M is

$$E[M] = \begin{bmatrix} E[X_{11}] & E[X_{12}] & \dots & E[X_{1n}] \\ E[X_{21}] & E[X_{22}] & \dots & E[X_{2n}] \\ \vdots & \cdots & \vdots & \vdots \\ E[X_{m1}] & E[X_{m2}] & \dots & E[X_{mn}] \end{bmatrix} = [E[X_{ij}]]_{m \times n}$$

5. The ***covariance matrix*** of a random vector X with mean vector μ is defined as

$$\text{Cov}(X) = E[(X - \mu)(X - \mu)^T]$$

which can be simplified to

$$\mathrm{Cov}(X) = \begin{bmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \dots & \mathrm{Cov}(X_1, X_n) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \dots & \mathrm{Cov}(X_2, X_n) \\ \vdots & \vdots & \dots & \vdots \\ \mathrm{Cov}(X_n, X_1) & \mathrm{Cov}(X_n, X_2) & \dots & \mathrm{Var}(X_n) \end{bmatrix}$$

6. Let X be an n-dimensional random vector and the random vector Y be defined as

$$Y = AX + b,$$

where A is a constant $m \times n$ matrix and b is a fixed m-dimensional vector. Then:

$$E[Y] = AE[X] + b$$

and

$$\mathrm{Cov}(Y) = A\mathrm{Cov}(X)A^T$$

Bivariate and Multivariate Normal

1. Two random variable X and Y are said to be ***bivariate normal*** or ***jointly normal*** if $aX+bY$ has a normal distribution for all $a, b \in \mathbb{R}$.

2. If X and Y are jointly normal, then X and Y must be individually normal.

3. If $X \sim \mathcal{N}(\mu_X, \sigma_X^2)$ and $Y \sim \mathcal{N}(\mu_Y, \sigma_Y^2)$ are *independent*, then they are jointly normal.

4. If $X \sim \mathcal{N}(\mu_X, \sigma_X^2)$ and $Y \sim \mathcal{N}(\mu_Y, \sigma_Y^2)$ are jointly normal then

$$X + Y \sim \mathcal{N}\left(\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2 + 2\rho\sigma_X\sigma_Y\right)$$

where ρ is the *correlation coefficient* of X and Y.

5. If X and Y are bivariate normal and uncorrelated, then they are independent.

6. If for $i = 1, 2, \dots n$, the random variables Z_i's are i.i.d. and ***standard normal***:

$$Z_i \sim \mathcal{N}(0, 1), \quad i = 1, 2, \dots, n$$

then the vector

$$Z = \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_n \end{bmatrix}$$

is called the ***standard normal vector***.

7. Taking $z = [z_1\ z_2\ \ldots\ z_n]^T$, the pdf of the standard normal vector Z is

$$f_Z(z) = \frac{1}{(2\pi)^{n/2}} e^{-\frac{1}{2}z^T z} = \frac{1}{(2\pi)^{n/2}} e^{-\frac{1}{2}\|z\|^2}$$

8. $E[Z] = 0$ and $\mathrm{Cov}(Z) = I$, the unit matrix.

9. Let A be a (non-singular) $n \times n$ matrix and μ be an n-dimensional vector and let

$$X = AZ + \mu$$

Introduce the notation $\Sigma = AA^T$. Then

$$E[X] = AE[Z] + \mu = \mu$$

and

$$\mathrm{Cov}(X) = AA^T = \Sigma$$

10. Transformation:

$$X = AZ + \mu$$

gives

$$Z = A^{-1}(X - \mu)$$

Then Jacobian is $\det(A^{-1}) = \dfrac{1}{\sqrt{\det(\Sigma)}}$

11. The pdf of X is

$$f_X(x) = \frac{1}{(2\pi)^{n/2}\sqrt{\Sigma}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

Hence, $X \sim \mathcal{N}(\mu, \Sigma)$

12. Let $X \sim \mathcal{N}(\mu_X, \Sigma_X)$ be an n-dimensional normal vector. Further, suppose B be a full rank $m \times n$ constant matrix and b is an m-dimensional constant vector. Then the random vector

$$Y = BX + b$$

is also distributed normally with mean vector μ_Y and covariance matrix Σ_Y, given by

$$\mu_Y = B\mu_X + b$$

$$\text{and } \Sigma_Y = B\Sigma_X B^T$$

That is

$$BX + b \sim \mathcal{N}\left(B\mu_X + b, B\Sigma_X B^T\right)$$

13. If $X \sim \mathcal{N}(\mu, \Sigma)$, then its components X_i and X_j are *independent* ***if and only if*** $\Sigma_{ij} = 0$.

Maximum Likelihood Estimate

1. Let X_1, X_2, X_3, ..., X_n be a random i.i.d. sample drawn from a distribution with a parameter θ, which can be a vector $\theta = [\theta_1\, \theta_2\, \ldots, \theta_k]^T$. Suppose that x_1, x_2, x_3, ..., x_n are the observed values of X_1, X_2, X_3, ..., X_n. The ***likelihood*** function is then defined as

$$\begin{aligned} L(x_1, x_2, \ldots, x_n; \theta) &= f_{X_1, X_2, \ldots, X_n}(x_1, x_2, \ldots, x_n; \theta) \\ &= \prod_{i=1}^{n} f_{X_i}(x_i; \theta) \end{aligned}$$

2. A ***maximum likelihood estimate (MLE)*** of θ, represented as $\hat{\theta}_{ML}$ is a value of θ that ***maximizes*** the likelihood function.

3. Often, it is easier to maximize the ***log likelihood*** function

$$\ln L(x_1, x_2, \ldots, x_n; \theta) = \sum_{i=1}^{n} \ln f_{X_i}(x_i; \theta)$$

4. MLE of some common distributions on the basis of the observed sample values $X_1 = x_1$, $X_2 = x_2$, ..., $X_n = x_n$, with

$$\bar{x} = \frac{x_1 + x_2 + \ldots + x_n}{n}$$

and

$$\text{S.D.} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Distribution	pmf/pdf	Parameter (θ)	MLE ($\hat{\theta}_{ML}$)
Bernoulli	$p^x(1-p)^{1-x}$, $x \in \{0, 1\}$	p	$\hat{p} = \bar{x}$
Binomial	${}^nC_x\, p^x(1-p)^{n-x}$, $x = 0, 1, \ldots, n$	p	$\hat{p} = \frac{\text{no. of successes}}{n}$
Poisson	$\frac{\lambda^x e^{-\lambda}}{x!}$, $x = 0, 1, 2, \ldots$	λ	$\hat{\lambda} = \bar{x}$
Uniform	$\frac{1}{b-a}, x \in [a, b]$ 0 otherwise	a, b	$\hat{a} = \min\{x_i\}$ $\hat{b} = \max\{x_i\}$
Exponential	$\lambda e^{-\lambda x}$, $x \geq 0$	λ	$\hat{\lambda} = \frac{1}{\bar{x}}$
Normal	$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$, $x \in \mathbb{R}$	μ, σ	$\hat{\mu} = \bar{x}$ $\hat{\sigma} = \text{S.D.}$

Linear Regression with Gaussian Noise

1. Given the data set

$$\mathcal{D} = \{(x_i, y_i) \mid i = 1, 2, \ldots, n\}$$

where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, a ***linear regression*** model tries to find the "best" linear function that fits the data set.

2. Defining the vector $X = [x_1\, x_2\, \ldots\, x_n]^T$ and $Y = [y_1\, y_2\, \ldots\, y_n]^T$, a model is

$$Y = w^T X + \epsilon$$

for a weight vector $w \in \mathbb{R}^d$ and a ***zero mean Normal vector*** whose each component are i.i.d. and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. Then

$$Y \mid X \sim \mathcal{N}(w^T X, \sigma^2)$$

Note that X and w are constant vectors but Y is a random vector.

3. Considering the unknown weight w as a parameter, maximum likelihood estimation can be used to obtain the optimum weight. This turns out to be the same problem as ***minimize***

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

Gaussian Mixture Model

1. An important ***unsupervised*** task in Machine Learning is to group an ***unlabeled*** data

$$\mathcal{D} = \{x_1, x_2, \ldots, x_n\}, \quad x_i \in \mathbb{R}^d$$

into some kind of homogeneous ***clusters***.

2. An approach is to assume that the data was produced by a generative model and then adjust the model parameters to maximize the probability that the model would produce exactly the data that is observed.

3. Let the clusters be assigned labels as 1, 2, ..., K, and let Z be a random variable that takes on values from these cluster labels with certain probabilities:

$$P(Z = k) = \pi_k, \quad k = 1, 2, \ldots, K$$

4. As a result, $\pi_k \geq 0$ for $k = 1, 2, \ldots, K$ and

$$\sum_{k=1}^{K} \pi_k = 1$$

5. Assume that the observed data vector $x = [x_1\, x_2\, \ldots\, x_n]^T$ is the value attained by the random vector

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}$$

6. A ***mixture model*** assumes a probability density of the form

$$P(X = x) = \sum_{k=1}^{K} P(Z = k) P(X = x | Z = k)$$

7. It is assumed that given $Z = k$, the random vector X is distributed *normally* with mean vector μ_k and variance matrix Σ_k:

$$X | Z = k \sim \mathcal{N}(\mu_k, \Sigma_k)$$

8. The ***Gaussian mixture model (GMM)*** represents the distribution

$$P(X = x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)$$

with $\pi_k \geq 0$ for $k = 1, 2, \ldots, K$ and $\sum_{k=1}^{K} \pi_k = 1$.

9. GMM is a density estimator.

10. The complete GMM is parametrized by the mean vectors, covariance matrices and the mixture weights from all component densities, collectively represented as

$$\lambda = \{\pi_k, \mu_k, \Sigma_k\}, \quad k = 1, 2, \ldots, K$$

11. The maximum likelihood estimate of the parameters λ is obtained iteratively using a variant of ***expectation-maximization (EM)*** algorithm.

12. In EM algorithm, start from an initial parameter set λ_0 and iterate through the data points one by one and perform the following two steps till either the algorithm converges or an upper limit of iteration is reached.

13. For the data point x_ℓ:

- **E step:** Evaluate the ***responsibilities*** using the current parameter:

$$\gamma(z_{\ell k}) = \frac{\pi_k \mathcal{N}(x_\ell|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_\ell|\mu_j, \Sigma_j)}$$

  The *responsibility* for a data point is the probability that it belongs to a cluster:

$$\gamma(z_{\ell k}) = P(Z = k|X = x_\ell)$$

- **M step:** Re-estimate the parameters using the current responsibilities:

$$\mu_k^{\text{new}} = \frac{1}{N_k} \sum_{\ell=1}^{n} \gamma(z_{\ell k}) x_\ell$$

$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{\ell=1}^{n} \gamma(z_{\ell k})(x_\ell - \mu_k^{\text{new}})(x_\ell - \mu_k^{\text{new}})^T$$

$$\pi_k^{\text{new}} = \frac{N_k}{n}$$

  where

$$N_k = \sum_{\ell=1}^{n} \gamma(z_{\ell k})$$

Some Inequalities and CLT

1. Markov's Inequality: Let X be a non-negative random variable with a finite mean μ. Then

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

2. **Chebyshev's Inequality:** Let X be a random variable with

$$E[X] = \mu \quad \text{and} \quad \text{Var}(X) = \sigma^2$$

Then

$$P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$$

3. **Hoeffding Inequality:** Suppose $X_1, X_2, \ldots, X_n$ be i.i.d.'s so that $E[X_i] = \mu$ and $a \leq X_i \leq b$. Define

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

Then

$$P(|\bar{X} - \mu| \geq \delta) \leq 2\exp\left(-\frac{2n\delta^2}{(b-a)^2}\right)$$

4. **Weak law of large numbers:** Suppose $X_1, X_2, \ldots, X_n \sim$ i.i.d. X so that $E[X] = \mu$ and $\text{Var}(X) = \sigma^2$. Define the sample mean

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

Then

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

5. **Central Limit Theorem (CLT):** Suppose $X_1, X_2, \ldots, X_n \sim$ i.i.d. X so that $E[X] = \mu$ and $\text{Var}(X) = \sigma^2$. Define

$$Y = X_1 + X_2 + \ldots + X_n$$

Then

$$\frac{Y - n\mu}{\sigma\sqrt{n}} \approx \mathcal{N}(0, 1)$$