Chapter 2 Mathematical Foundation

supplementary slides to

Machine Learning Fundamentals

©Hui Jiang 2020

published by Cambridge University Press

August 2020

Outline

Vectors and Matrices

- a vector: a list of numbers arranged in order
  - an abstract way to represent objects in math
  - vectors are written in a column, e.g. $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

- a matrix: a group of numbers arranged in a 2-d array, e.g. $\mathbf{A} \in \mathbb{R}^{m \times n}$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Matrix Multiplication (I)

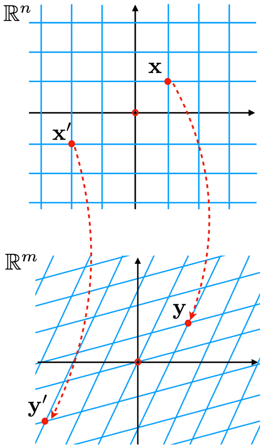
- a matrix: representing a particular way to move any point in one space to another
- matrix multiplication: the way to implement the above motion

$$\overbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_m \end{bmatrix}}^{\mathbf{y}} = \overbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i1} & \cdots & a_{ij} & \cdots & a_{in} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & \cdots & a_{mn} \end{bmatrix}}^{\mathbf{A}} \overbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_j \\ \vdots \\ x_n \end{bmatrix}}^{\mathbf{x}} \qquad y_i = \sum_{j=1}^{n} a_{ij} x_j \quad (\forall i = 1, 2, \cdots, m)$$

$\mathbf{y} = \mathbf{A}\mathbf{x}$ $(\mathbf{A} \in \mathbb{R}^{m \times n})$ represents a mapping from a point $\mathbf{x}$ in $\mathbb{R}^n$ to another point $\mathbf{y}$ in $\mathbb{R}^m$

Linear Transformation as Matrix Multiplication

- matrix multiplication: can only represent a **linear transformation**
- a mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$ is linear if and only if :
  - the origin in $\mathbb{R}^n$ is mapped to the origin in $\mathbb{R}^m$
  - *every* straight line in $\mathbb{R}^n$ is always mapped to a straight line (or a single point) in $\mathbb{R}^m$
- nonlinear transformations: require other methods rather than matrix multiplication



Matrix Multiplication (II)

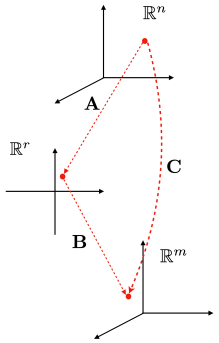
- matrix multiplication between two matrices $\mathbf{A} \in \mathbb{R}^{m\times r}$ and $\mathbf{B} \in \mathbb{R}^{r\times n}$

$$\mathbf{C} = \mathbf{AB} \quad (\text{with } \ \mathbf{C} \in \mathbb{R}^{m\times n})$$

$$\underbrace{\begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & c_{ij} & \vdots \\ c_{m1} & \cdots & c_{mn} \end{bmatrix}}_{\mathbf{C}} = \underbrace{\begin{array}{c} \\ \text{i-th row} \\ \\ \end{array}\begin{bmatrix} a_{11} & \cdots & a_{1r} \\ \vdots & a_{ik} & \vdots \\ a_{m1} & \cdots & a_{mr} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{array}{c} \text{j-th column} \\ \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & b_{kj} & \vdots \\ b_{r1} & \cdots & b_{rn} \end{bmatrix} \end{array}}_{\mathbf{B}} \qquad c_{ij} = \sum_{k=1}^{r} a_{ik} b_{kj}$$

$$\forall i = 1, 2, \cdots, m$$
$$\forall j = 1, 2, \cdots, n$$

- $\mathbf{C}$ represents a composition of two linear transformations: $\mathbf{A}$ and $\mathbf{B}$



Matrix Operations (I)

- **matrix transpose**: $\mathbf{A} \in \mathbb{R}^{m\times n} \rightarrow \mathbf{A}^\intercal \in \mathbb{R}^{n\times m}$
  - a square matrix $\mathbf{A}$ is symmetric *iff* $\mathbf{A} = \mathbf{A}^\intercal$
  - $(\mathbf{A}^\intercal)^\intercal = \mathbf{A} \quad (\mathbf{AB})^\intercal = \mathbf{B}^\intercal\mathbf{A}^\intercal \quad (\mathbf{A}\pm\mathbf{B})^\intercal = \mathbf{A}^\intercal \pm \mathbf{B}^\intercal$
- **determinant**: $|\mathbf{A}| \in \mathbb{R}$ for a square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$
- **inverse matrix** $\mathbf{A}^{-1}$: for an $n \times n$ square matrix $\mathbf{A}$ *iif* $\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$
- **inner-product**: $\mathbf{w}\cdot\mathbf{x}$ $(\in \mathbb{R})$ of two vectors $\mathbf{w},\mathbf{x} \in \mathbb{R}^n$
  - $\mathbf{w}\cdot\mathbf{x} \triangleq \sum_{i=1}^n w_i x_i = \mathbf{w}^\intercal\mathbf{x} = \mathbf{x}^\intercal\mathbf{w}$
  - **norm** $(L_2$ norm$)$ of vector $\mathbf{w}$: $\|\mathbf{w}\|^2 = \mathbf{w}\cdot\mathbf{w}$
- **trace** of an $n\times n$ square matrix: $\text{tr}(\mathbf{A}) = \sum_{i=1}^n a_{ii}$

Matrix Operations (II)

- **eigenvalues** and **eigenvectors** of an $n \times n$ square matrix $\mathbf{A}$:
  $\mathbf{A}\,\mathbf{u} = \lambda\,\mathbf{u}$ with $\lambda \in \mathbb{R}$ and non-zero $\mathbf{u} \in \mathbb{R}^n$
- A symmetric matrix is **positive definite** (or **semi-definite**) if all eigenvalues are positive (or non-negative).
- If a square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ has n orthogonal eigenvectors $\mathbf{u}_i$ $(i = 1, 2, \cdots, n)$: $\mathbf{A}\,\mathbf{u}_i = \lambda_i\,\mathbf{u}_i$ (normalized as $\|\mathbf{u}_i\| = 1$), then $\mathbf{A}$ can be factorized as $\mathbf{A} = \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{U}^\intercal$.

$$\mathbf{A} = \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \\ | & | & & | \end{bmatrix}}_{\mathbf{U}} \underbrace{\begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}}_{\mathbf{\Lambda}} \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \\ | & | & & | \end{bmatrix}^\intercal}_{\mathbf{U}^\intercal}$$

Matrix Calculus

If y is a function involving all elements of a vector $\mathbf{x}$ (or a matrix $\mathbf{A}$), $\frac{\partial y}{\partial \mathbf{x}}$ (or $\frac{\partial y}{\partial \mathbf{A}}$) is a vector (or a matrix) of the same size.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{and} \quad \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

$$\frac{\partial y}{\partial \mathbf{x}} \triangleq \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_n} \end{bmatrix} \quad \text{and} \quad \frac{\partial y}{\partial \mathbf{A}} \triangleq \begin{bmatrix} \frac{\partial y}{\partial a_{11}} & \frac{\partial y}{\partial a_{12}} & \cdots & \frac{\partial y}{\partial a_{1n}} \\ \frac{\partial y}{\partial a_{21}} & \frac{\partial y}{\partial a_{22}} & \cdots & \frac{\partial y}{\partial a_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y}{\partial a_{m1}} & \frac{\partial y}{\partial a_{m2}} & \cdots & \frac{\partial y}{\partial a_{mn}} \end{bmatrix}$$

Matrix Calculus Formula for Machine Learning (I)

$$\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{x}^\intercal \mathbf{x}\right) = 2\mathbf{x}$$

$$\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{x}^\intercal \mathbf{y}\right) = \mathbf{y}$$

$$\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{x}^\intercal \mathbf{A}\mathbf{x}\right) = \mathbf{A}\mathbf{x} + \mathbf{A}^\intercal \mathbf{x}$$

$$\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{x}^\intercal \mathbf{A}\mathbf{x}\right) = 2\mathbf{A}\mathbf{x} \quad (\text{symmetric } \mathbf{A})$$

$$\frac{\partial}{\partial \mathbf{A}}\left(\mathbf{x}^\intercal \mathbf{A}\mathbf{y}\right) = \mathbf{x}\mathbf{y}^\intercal$$

Matrix calculus formula for machine learning (II)

$$\frac{\partial}{\partial \mathbf{A}}\left(\mathbf{x}^\intercal \mathbf{A}^{-1}\mathbf{y}\right) = -(\mathbf{A}^\intercal)^{-1}\mathbf{x}\mathbf{y}^\intercal(\mathbf{A}^\intercal)^{-1} \quad \text{(square } \mathbf{A}\text{)}$$

$$\frac{\partial}{\partial \mathbf{A}}\Big(\ln|\mathbf{A}|\Big) = (\mathbf{A}^{-1})^\intercal = (\mathbf{A}^\intercal)^{-1} \quad \text{(square } \mathbf{A}\text{)}$$

$$\frac{\partial}{\partial \mathbf{A}}\Big(\mathrm{tr}(\mathbf{A})\Big) = \mathbf{I} \quad \text{(square } \mathbf{A}\text{)}$$

Random Variables

- probability $\Pr(A)$: a real number between 0 and 1, indicating how likely a random event A is to occur
- **random variables**: taking different values in different probabilities, e.g. $X, Y, Z, \cdots$
  - discrete random variables (RVs)
  - continuous random variables (RVs)
- a random variable is fully specified by two ingredients:
  - its domain: the set of all possible values
  - its probability distribution: how likely it takes each value
- a probability function is used to characterize how likely a random variable may take each value in the domain

Probability Functions

- *probability mass functions (p.m.f)* for discrete random variables
  - $p(x) \triangleq \Pr(X = x)$ in the domain $\forall x \in \{x_1, x_2, \cdots\}$
  - sum-to-one constraint: $\sum_x p(x) = 1$
- *probability density functions (p.d.f)* for continuous random variables
  - define $p(x)$ to ensure $\Pr(a \le X \le b) = \int_a^b p(x)\,dx$
  - sum-to-one constraint: $\int_{-\infty}^{+\infty} p(x)\,dx = 1$

x	x_1	x_2	x_3	x_4
$p(x)$	0.4	0.3	0.2	0.1

Table: an example of p.m.f

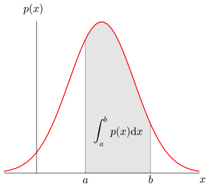


Figure: a p.d.f. in $(-\infty, \infty)$

Expectation: Mean, Variance and Moments

- **expectation** of any function of a random variable $f(X)$ is defined as

$$\mathbb{E}\big[f(X)\big] = \int_{-\infty}^{+\infty} f(x)\,p(x)\,dx \quad \text{or} \quad \mathbb{E}\big[f(X)\big] = \sum_{x} f(x)\,p(x)$$

- **mean** of a random variable X: $\mathbb{E}[X]$
  - the mean indicates the center of the distribution
- **variance** of X: $\mathrm{var}\big(X\big) = \mathbb{E}\Big[\big(X - \mathbb{E}[X]\big)^2\Big]$
  - the variance tells the average deviation from the center
- r-th **moment** of X: $\mathbb{E}[X^r]$ $(\forall r \in \mathbb{N})$
- show: $\mathrm{var}\big(X\big) = \mathbb{E}\big[X^2\big] - \big(\mathbb{E}[X]\big)^2$

Joint Distributions

joint distributions of multiple random variables are multivariate probability functions defined in the product space of their domains

- joint p.m.f. for two discrete RVs

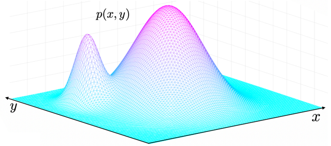
  $p(x,y) \overset{\Delta}{=} \Pr(X=x, Y=y)$
  - $\forall x \in \{x_1, \cdots\}, y \in \{y_1, \cdots\}$
  - $\sum_x \sum_y p(x,y) = 1$

$y \setminus x$	x_1	x_2	x_3
y_1	0.03	0.24	0.17
y_2	0.23	0.11	0.22

- joint p.d.f. for two continuous RVs

$$\Pr\big((x,y) \in \Omega\big) = \int\int_{\Omega} p(x,y)dxdy$$

  - $\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p(x,y)\,dxdy = 1$



Marginal and Conditional Distributions

- a joint distribution fully specifies all random variables
- **marginalization** (a.k.a. *the rule of sum* in probability)
  - joint distribution → marginal distribution

$$p(x) = \int_{-\infty}^{+\infty} p(x,y)dy \quad \text{or} \quad p(x) = \sum_y p(x,y)$$

- **conditional distribution**: $p(x\,|\,y) \triangleq \frac{p(x,y)}{p(y)}$
  - *the general product rule* in probability:

$$p(x_1, x_2, x_3, \cdots) = p(x_1)\,p(x_2|x_1)\,p(x_3|x_1,x_2)\,\cdots$$

  - conditional expectation: $\mathbb{E}_X\left[f(X)\,\middle|\,Y = y_0\right]$
  - conditional mean: $\mathbb{E}_X\left[X\,\middle|\,Y = y_0\right]$
- covariance: $\text{cov}(X,Y) = \mathbb{E}\big[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])\big]$
- independence $\iff p(x,y) = p(x)\,p(y) \quad (\forall x, y)$
- uncorrelatedness $\iff \text{cov}(X,Y) = 0$

Probability Distributions of Random Vectors

- random vector: whose elements are all random variables

$$p(\underbrace{x_1, x_2, x_3}_{\mathbf{x}}, \underbrace{y_1, y_2, y_3, y_4}_{\mathbf{y}}) = p(\mathbf{x}, \mathbf{y})$$

- conditional distribution: $p(\mathbf{x} \,|\, \mathbf{y}) \overset{\Delta}{=} \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{y})}$
- mean vector: $\mathbb{E}[\mathbf{x}] = \int \mathbf{x}\, p(\mathbf{x})\, d\mathbf{x} = \int\int \mathbf{x}\, p(\mathbf{x}, \mathbf{y})\, d\mathbf{x} d\mathbf{y}$ or $\mathbb{E}\big[\mathbf{x}\big] = \sum_{\mathbf{x}} \sum_{\mathbf{y}} \mathbf{x}\, p(\mathbf{x}, \mathbf{y})$
- covariance matrix: $\text{cov}(\mathbf{x}, \mathbf{y}) = \mathbb{E}\Big[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{y} - \mathbb{E}[\mathbf{y}])^\intercal\Big]$
- the rule of sum: $p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{y})\, d\mathbf{y}$ or $p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y})$
- the general product rule: $p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p(\mathbf{x})\, p(\mathbf{y}|\mathbf{x})\, p(\mathbf{z}|\mathbf{x}, \mathbf{y})$

Common Probability Distributions

- Binomial Distribution
- Multinomial Distribution
- Beta Distribution
- Dirichlet Distribution
- Univariate Gaussian (Normal) Distribution
- Multivariate Gaussian Distribution
- Poisson Distribution
- Uniform Distribution
- Gamma Distribution
- inverse-Wishart Distribution
- von Mises–Fisher Distribution

Binomial Distribution

$$\mathsf{B}(r \mid N, p) \triangleq \Pr(X = r) = \frac{N!}{r!\,(N-r)!}\, p^r (1-p)^{N-r}$$

- parameters: $N \in \mathbb{N}$ and $p \in [0, 1]$
- *support*: the domain of the random variable is $r \in \{0, 1, \cdots N\}$
- mean and variance:

$$\mathbb{E}[X] = Np$$

$$\mathsf{var}(X) = Np(1-p)$$

- sum-to-one constraint: $\sum_{r=0}^{N} \mathsf{B}(r \mid N, p) = 1$

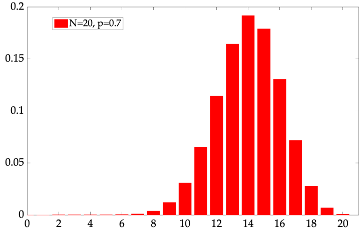


Figure: binomial distributions are **unimodal**

Multinomial Distribution

$$\text{Mult}\big(r_1, r_2, \cdots, r_m \,\big|\, N, p_1, p_2, \cdots, p_m\big) \triangleq \Pr(X_1 = r_1, \cdots, X_m = r_m)$$
$$= \frac{N!}{r_1!\, r_2! \,\cdots\, r_m!}\, p_1^{r_1}\, p_2^{r_2} \,\cdots\, p_m^{r_m}$$

- multivariate extension of binomial distribution
- parameters: $N \in \mathbb{N}$; $0 \leq p_i \leq 1$ $(\forall i)$ and $\sum_{i=1}^{m} p_i = 1$
- support (the domain of m random variables):
  $r_i \in \{0, 1, \cdots N\}$ $(\forall i = 1, \cdots, m)$ and $\sum_{i=1}^{m} r_i = N$
- means, variances and covariances:
  $\mathbb{E}\big[X_i\big] = N p_i$ and $\text{var}(X_i) = N p_i (1 - p_i) \quad (\forall i)$
  $\text{cov}(X_i, X_j) = -N p_i p_j \qquad (\forall i, j)$
- sum-to-one: $\sum_{r_1 \cdots r_m} \text{Mult}\big(r_1, \cdots, r_m \,\big|\, N, p_1, \cdots, p_m\big) = 1$

Beta Distribution

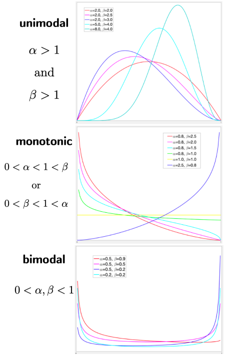
$$\text{Beta}\left(x \,\middle|\, \alpha, \beta\right) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\, x^{\alpha-1}\,(1-x)^{\beta-1}$$

- parameters: $\alpha > 0$ and $\beta > 0$
- support: $x \in \mathbb{R}$ and $0 \leq x \leq 1$
- mean and variance:

$$\mathbb{E}\left[X\right] = \frac{\alpha}{\alpha+\beta}$$

$$\text{var}(X) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

- sum-to-one: $\int_0^1 \text{beta}\left(x \,\middle|\, \alpha, \beta\right) dx = 1$



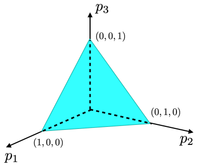
Dirichlet Distribution (I)

$$\text{Dir}\big(p_1, p_2, \cdots, p_m \,\big|\, r_1, r_2, \cdots, r_m\big) = \frac{\Gamma(r_1 + \cdots + r_m)}{\Gamma(r_1) \cdots \Gamma(r_m)}\, p_1^{r_1 - 1} p_2^{r_2 - 1} \cdots p_m^{r_m - 1}$$

- parameters: $r_i > 0 \ (\forall i = 1, \cdots, m)$
- support is an m-dimensional simplex:
  $0 < p_i < 1 \ \ (\forall i = 1, \cdots, m)$ and $\sum_{i=1}^{m} p_i = 1$
- means, variances and covariances:
  $\mathbb{E}\big[X_i\big] = \frac{r_i}{r_0}$ var$\big(X_i\big) = \frac{r_i(r_0 - r_i)}{r_0^2(r_0+1)}$
  cov$\big(X_i, X_j\big) = -\frac{r_i r_j}{r_0^2(r_0+1)}$, where $r_0 = \sum_{i=1}^{m} r_i$
- sum-to-one inside the simplex:
  $\int \cdots \int_{p_1 \cdots p_m} \text{Dir}\big(p_1, p_2, \cdots p_m \,\big|\, r_1, r_2, \cdots, r_m\big)\, dp_1 \cdots dp_m = 1$

Dirichlet Distribution (II)

- multivariate extension of beta distribution
- conjugate with multinomial distribution
- Dirichlet distributions ($m = 3$) in the 3-D simplex:



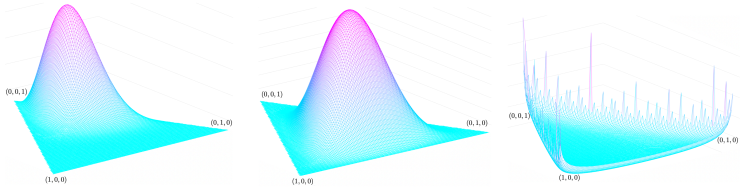


Figure: i) regular: $r_1 = 2.0, r_2 = 4.0, r_3 = 10.0$;
ii) symmetric: $r_1 = r_2 = r_3 = 4.0$;
iii) sparse: $r_1 = 0.7, r_2 = 0.8, r_3 = 0.9$;

Univariate Gaussian Distribution

$$\mathcal{N}(x \,|\, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \, e^{-\frac{(x-\mu)^2}{2\,\sigma^2}}$$

- parameters: $\mu \in \mathbb{R}$ and $\sigma^2 > 0$
- support: $x \in \mathbb{R}$
- mean and variance:

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

- sum-to-one: $\int_{-\infty}^{+\infty} \mathcal{N}(x \mid \mu, \sigma)\, dx = 1$
- bell-shaped **unimodal**

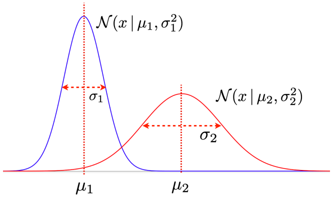


Figure: two univariate Gaussian distributions ($\sigma_2 > \sigma_1$).

Multivariate Gaussian Distribution

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} \, e^{-\frac{(\mathbf{x}-\boldsymbol{\mu})^\intercal \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}{2}}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- parameters: $\boldsymbol{\mu} \in \mathbb{R}^n$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n} \succ 0$
- support: $\mathbf{x} \in \mathbb{R}^n$
- mean vector and covariance matrix:

$$\mathbb{E}\big[\mathbf{x}\big] = \boldsymbol{\mu} \quad \text{and} \quad \text{cov}\big(\mathbf{x}, \mathbf{x}\big) = \boldsymbol{\Sigma}$$

- sum-to-one: $\int \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \, d\mathbf{x} = 1$
- any marginal distribution or conditional distribution is Gaussian.
- multivariate unimodal

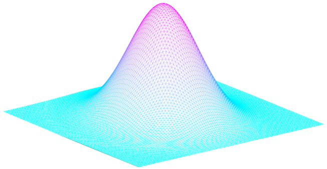


Figure: a multivariate Gaussian distribution in 2-dimensional space

Poisson Distribution

$$\text{Poisson}\big(n \,|\, \lambda\big) \triangleq \Pr(X = n) = \frac{e^{-\lambda} \cdot \lambda^n}{n!} \quad \forall n = 0, 1, 2 \cdots$$

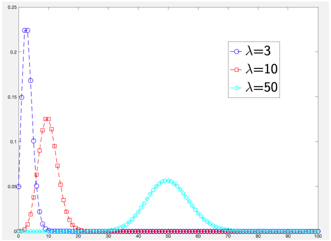
- parameter: $\lambda > 0$
- support: the domain of the random variable

$$n = 0, 1, 2, \cdots$$

- mean and variance:

$$\mathbb{E}[X] = \lambda \quad \text{and} \quad \text{var}(X) = \lambda$$

- sum-to-one: $\sum_{n=0}^{\infty} \text{Poisson}\big(n \,|\, \lambda\big) = 1$
- ideal distributions for count data



Transformation of Random Variables

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \xrightarrow{f} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

If the transformation f is differentiable and invertible, then

$$p(\mathbf{y}) = \left| \mathbf{J}(\mathbf{y}) \right| p(\mathbf{x}) = \left| \mathbf{J}(\mathbf{y}) \right| p\big(f^{-1}(\mathbf{y})\big)$$

where $\mathbf{J}(\mathbf{y})$ denotes the Jacobian matrix of $\mathbf{x} = f^{-1}(\mathbf{y})$:

$$\mathbf{J}(\mathbf{y}) = \left[\frac{\partial x_i}{\partial y_j} \right]_{n \times n} = \begin{bmatrix} \frac{\partial x_1}{\partial y_1} & \cdots & \frac{\partial x_1}{\partial y_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_n}{\partial y_1} & \cdots & \frac{\partial x_n}{\partial y_n} \end{bmatrix}$$

Information Theory

- Shannon's **information** of an event: $I(A) = -\log_2 \big(\Pr(A) \big)$
- **entropy** of a random variable: the total amount of uncertainty

$$H(X) = \mathbb{E}[-\log_2 \Pr(X = x)] = -\sum_x p(x) \log_2 p(x)$$

  - the amount of information to resolve the random variable
- **joint entropy** of two random variables:

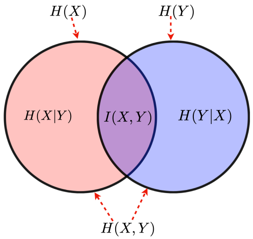
$$\begin{aligned} H(X, Y) &= \mathbb{E}_{X,Y}\big[-\log_2 \Pr(X = x, Y = y) \big] \\ &= -\sum_x \sum_y p(x, y) \log_2 p(x, y) \end{aligned}$$

- **conditional entropy**:
  $H(Y|X) = \mathbb{E}_{X,Y}[-\log_2 \Pr(Y = y|X = x)]$
  $= -\sum_x \sum_y p(x, y) \log_2 p(y|x)$

Mutual Information

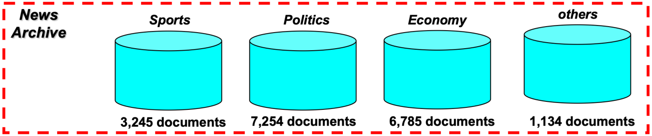
- **mutual information**:
  - entropy reduction of X after observing Y
  - how much Y can tell about X

$$
\begin{aligned}
I(X,Y) &= H(X) - H(X|Y) \\
&= \sum_x \sum_y p(x,y) \log_2 \left(\frac{p(x,y)}{p(x)p(y)} \right)
\end{aligned}
$$



- $I(X,Y) = H(Y) - H(Y|X)$
  $= H(X) + H(Y) - H(X,Y)$
- symmetrical: $I(X,Y) = I(Y,X)$
- non-negative: $I(X,Y) \geq 0$
- $I(X,Y) = 0$ *iff* X and Y are independent

Example: Mutual Information for Keyword Selection



- text categorization: classify text documents into various topics
- text documents contain a large number of distinct words
- goal: use mutual information as a criterion to automatically filter out non-informative words
- for any word (e.g. *score*) and a topic (e.g. *sports*), define two binary random variables X and Y:
  - $X \in \{0,1\}$: whether a document's topic is *sports* or not
  - $Y \in \{0,1\}$: whether a document contains *score* or not
  - $I(X,Y) \implies$ relationship between *score* and *sports*

Example: Mutual Information for Keyword Selection

- count all training documents to estimate $p(X,Y)$
- compute mutual information: $p(X,Y) \rightarrow I(X,Y)$

$$p(X{=}1,Y{=}1)=\frac{\#\text{ of docs with topic } \textit{sports} \text{ and containing } \textit{score}}{\text{total } \#\text{ of docs in the corpus}}$$

$$p(X{=}1,Y{=}0)=\frac{\#\text{ of docs with topic } \textit{sports} \text{ but not containing } \textit{score}}{\text{total } \#\text{ of docs in the corpus}}$$

$p(x,y)$	$y{=}0$	$y{=}1$	$p(x)$
$x{=}0$	0.80	0.02	0.82
$x{=}1$	0.11	0.07	0.18
$p(y)$	0.91	0.09	

$$\begin{aligned} I(X,Y) &= \sum_{x\in\{0,1\}} \sum_{y\in\{0,1\}} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} \\ &= 0.126 \end{aligned}$$

- for another pair, e.g. *what* vs. *sports*, $I(X,Y){=}0.00007$
- *sports* is a keyword for the topic *sports*, *what* is not
- repeat the above procedure for all pairs of words and topics to identify keywords

KL Divergence (I)

KL divergence is a criterion to measure the difference between two probability distributions that have the same support

$$\mathsf{KL}\Big(p(x) \,\big\|\, q(x)\Big) \triangleq \mathbb{E}_{x\sim p(x)}\left[\log\left(\frac{p(x)}{q(x)}\right)\right] = \sum_x p(x)\,\log\left(\frac{p(x)}{q(x)}\right)$$

Theorem 1

The KL divergence is always non-negative:

$$\mathsf{KL}\big(p(x) \,\|\, q(x)\big) \geq 0$$

And $\mathsf{KL}\big(p(x) \,\|\, q(x)\big) = 0$ *iff* $p(x) = q(x)$ *holds almost everywhere.*

proved as a corollary from Jensen's inequality

KL Divergence (II)

- $\mathsf{KL}\big(q(x)\,||\,p(x)\big)$ represents the amount of information lost when we replace one probability distribution $p(x)$ with another distribution $q(x)$
- when using a simple model $q(x)$ to approximate a complicated model $p(x)$, the best-fit $q^*(x)$ may be derived as:

$$q^*(x) = \arg\min_{q(x)} \mathsf{KL}\big(q(x)\,||\,p(x)\big)$$

- KL divergence vs. mutual information
  - $I(X,Y) = \mathsf{KL}\big(p(x,y)\,\|\,p(x)p(y)\big)$
  - mutual information can be viewed as the information gain from the assumption of independence

General Formulation for Optimization

the general formulation for all optimization problems:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \; f(\mathbf{x})$$

subject to

$$h_i(\mathbf{x}) = 0 \qquad (i = 1, 2, \cdots, m)$$

$$g_j(\mathbf{x}) \leq 0 \qquad (j = 1, 2, \cdots, n)$$

where m equality constraints and n inequality constraints jointly define a non-empty feasible set Ω for the free variables $\mathbf{x}$.

- two special cases:
    - linear programming
    - convex optimization

Optimality Conditions

- **optimality conditions**: the necessary and/or sufficient conditions for any $\mathbf{x}^*$ to be an optimal solution of an optimization problem
- optimality conditions may help to derive the analytic solution to some simple optimization problems
- three cases from easy to hard:
  - unconstrained optimization
  - optimization under equality constraints
  - general optimization subject to both inequality and equality constraints

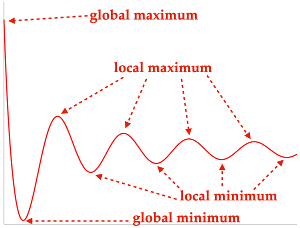
Optimality Conditions: Unconstrained Optimization (I)

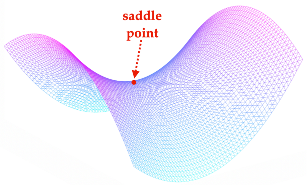
$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$$

- global minimum (maximum)
- local minimum (maximum)
- stationary point:

$$\nabla f(\hat{\mathbf{x}}) \triangleq \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}} = 0$$

- critical point: a stationary point or an undifferentiable point
- saddle point: a stationary point but not a local minimum (maximum)





Optimality Conditions: Unconstrained Optimization (II)

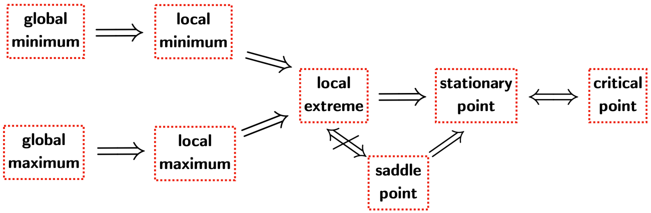


Figure: a diagram for any differentiable function

Theorem 2 (necessary condition for unconstrained optimization)

Assume $f(\mathbf{x})$ is differentiable everywhere. If $\mathbf{x}^*$ is a local minimum, then $\mathbf{x}^*$ must be a stationary point, i.e. the gradient vanishes at $\mathbf{x}^*$ as $\nabla f(\mathbf{x}^*) = 0$.

Optimality Conditions: Unconstrained Optimization (III)

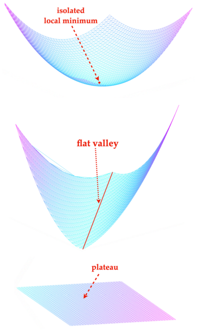
Hessian matrix: $\mathbf{H}(\mathbf{x}) = \left[\frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} \right]_{n \times n}$

Theorem 3 (second order necessary condition)

Assume $f(\mathbf{x})$ is twice differentiable. If $\mathbf{x}^*$ is a local minimum, then $\nabla f(\mathbf{x}^*) = 0$ and $\mathbf{H}(\mathbf{x}^*) \succeq 0$.

Theorem 4 (second order sufficient condition)

Assume $f(\mathbf{x})$ is twice differentiable. If a point $\mathbf{x}^*$ satisfies $\nabla f(\mathbf{x}^*) = 0$ and $\mathbf{H}(\mathbf{x}^*) \succ 0$, then $\mathbf{x}^*$ is an isolated local minimum.



i) isolated minimum: $\mathbf{H}(\mathbf{x}) \succ 0$

ii) flat valley: $\mathbf{H}(\mathbf{x}) \succeq 0$ and $\mathbf{H}(\mathbf{x}) \neq 0$

iii) plateau: $\mathbf{H}(\mathbf{x}) = 0$

Optimality Conditions: Equality Constraints (I)

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \ f(\mathbf{x})$$

subject to $h_i(\mathbf{x}) = 0 \quad (i = 1, 2, \cdots, m)$

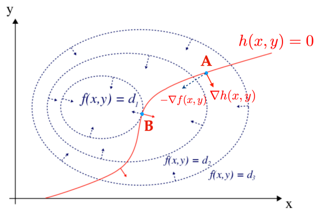
Theorem 5 (Lagrange necessary conditions)

Given $f(\mathbf{x})$ and $\{h_i(\mathbf{x})\}$ are differentiable. If a point $\mathbf{x}^*$ is a local optimum, then the gradients of these functions are linearly dependent at $\mathbf{x}^*$:

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) = 0$$

where $\lambda_i \in \mathbb{R}$ are the Lagrange multipliers.

stationary points of $f(\mathbf{x})$ may not satisfy the constraints.



$$\min_{x,y} \ f(x,y)$$

subject to

$$h(x,y) = 0$$

Optimality Conditions: Equality Constraints (II)

- Theorem 5 suggests **the method of Lagrange multipliers**
- introduce a Lagrange multiplier $\lambda_i \in \mathbb{R}$ for each equality constraint
- construct the *Lagrangian* function:

$$L\big(\mathbf{x}, \{\lambda_i\}\big) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \, h_i(\mathbf{x})$$

- minimize the *Lagrangian* function w.r.t. $\mathbf{x}$ and all Lagrange multipliers $\{\lambda_i\}$, converting a constrained optimization problem into an unconstrained one

Optimality Conditions: Inequality Constraints

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \; f(\mathbf{x})$$

subject to

$$h_i(\mathbf{x}) = 0 \qquad (i = 1, 2, \cdots, m)$$

$$g_j(\mathbf{x}) \leq 0 \qquad (j = 1, 2, \cdots, n)$$

- introduce a multiplier $\lambda_i \in \mathbb{R}$ for each equality constraint
- introduce a multiplier $\nu_i \geq 0$ for each inequality constraint
- construct a Lagrangian function:

$$L\big(\mathbf{x}, \{\lambda_i, \nu_j\}\big) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \, h_i(\mathbf{x}) + \sum_{j=1}^{n} \nu_j \, g_j(\mathbf{x})$$

- optimize $L\big(\mathbf{x}, \{\lambda_i, \nu_j\}\big)$ for the optimality conditions

Dual Problem and Strong Duality

- **Lagrange dual function**: defined as the lower-bound w.r.t. $\mathbf{x}$

$$L^*\Big(\{\lambda_i, \nu_j\}\Big) = \inf_{\mathbf{x}\in\Omega} \; L\Big(\mathbf{x}, \{\lambda_i, \nu_j\}\Big)$$

- generally $L^*\Big(\{\lambda_i, \nu_j\}\Big) \leq L\Big(\mathbf{x}, \{\lambda_i, \nu_j\}\Big) \leq f(\mathbf{x})$ for all $\mathbf{x} \in \Omega$
- **Lagrange dual problem**:

$$\{\lambda_i^*, \nu_j^*\} = \arg \max_{\{\lambda_i, \nu_j\}} \; L^*\Big(\{\lambda_i, \nu_j\}\Big)$$

  subject to $\nu_j \geq 0$ for all $j = 1, 2, \cdots, n$
- **strong duality** occurs if $L^*\big(\{\lambda_i^*, \nu_j^*\}\big) = f(\mathbf{x}^*)$, under which the dual problem is equivalent to the original problem

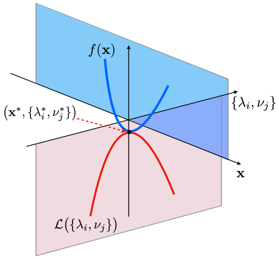
Optimality Conditions: KKT Conditions

Theorem 6 (KKT necessary conditions)

If $\mathbf{x}^*$ *and* $\{\lambda_i^*, \nu_j^*\}$ *is a saddle point of* $L(\mathbf{x}, \{\lambda_i, \nu_j\})$*, then* $\mathbf{x}^*$ *is a local minimum. The saddle point satisfies the following conditions:*

1. *stationariness:*
   $\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla h_i(\mathbf{x}^*) + \sum_{j=1}^{n} \nu_j^* \nabla g_j(\mathbf{x}^*) = 0$
2. *primal feasibility:*
   $h_i(\mathbf{x}^*) = 0 \;\; (i = 1, 2, \cdots, m)$
   $g_j(\mathbf{x}^*) \leq 0 \;\; (j = 1, 2, \cdots, n)$
3. *dual feasibility:* $\nu_j^* \geq 0 \quad (j = 1, 2, \cdots, n)$
4. *complementary slackness:*
   $\nu_j^* g_j(\mathbf{x}^*) = 0 \;\; (j = 1, 2, \cdots, n)$

strong duality $\Longrightarrow$ $\mathbf{x}^*$ and $\{\lambda_i^*, \nu_j^*\}$ is a saddle point


$f(\mathbf{x})$
$(\mathbf{x}^*, \{\lambda_i^*, \nu_j^*\})$
$\{\lambda_i, \nu_j\}$
$\mathbf{x}$
$\mathcal{L}(\{\lambda_i, \nu_j\})$

Numerical Optimization Methods

- optimality conditions do not always yield a useful closed-form solution
- many optimization problems in machine learning require iterative numerical methods
- take the unconstrained optimization problem as example

$$\arg\min_{\mathbf{x}\in\mathbb{R}^n} \; f(\mathbf{x})$$

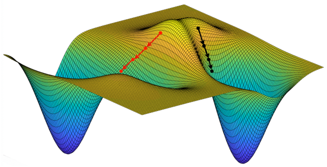
- numerical optimization methods
  - **zero-order methods**: using the function values alone, such as *grid search*
  - **first-order methods**: using the function values and gradients, such as *gradient descent method*
  - **second-order methods**: using the function values, gradients and Hessians, such as *Newton method*

First-order Methods: Gradient Descent Method

- the gradient points to a direction of the fastest increase of the function value
- *gradient descent*: repeatedly move a small step along the direction of the negative gradient until converged

Gradient Descent Method

```
randomly choose $\mathbf{x}^{(0)}$, and set $\eta_0$
set $n = 0$
**while** not converged **do**
    update:   $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \eta_n \nabla f(\mathbf{x}^{(n)})$
    adjust:   $\eta_n \to \eta_{n+1}$
    $n = n + 1$
**end while**
```



First-order Methods: Stochastic Gradient Descent (I)

- the objective function $f(\mathbf{x})$ in machine learning can often be decomposed as a sum of homogeneous components:

$$f(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} f_i(\mathbf{x})$$

  e.g. $f_i(\mathbf{x})$ indicates the loss measure on each training sample
- when N is large, it is too expensive to compute

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

- **stochastic gradient descent**: estimate $\nabla f(\mathbf{x})$ using a random sample or a small subset (mini-batch) of random samples at each time

First-order Methods: Stochastic Gradient Descent (II)

Stochastic Gradient Descent (SGD)

```
randomly choose $\mathbf{x}^{(0)}$, and set $\eta_0$
set $n = 0$
**while** not converged **do**
    randomly choose a sample $k$
    update:    $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \eta_n \nabla f_k(\mathbf{x}^{(n)})$
    adjust:    $\eta_n \to \eta_{n+1}$
    $n = n + 1$
**end while**
```

First-order Methods: Stochastic Gradient Descent (III)

Mini-batch SGD

```
randomly choose $\mathbf{x}^{(0)}$, and set $\eta_0$
set $n = 0$
**while** not converged **do**
    randomly shuffle all training samples into mini-batches
    **for** each mini-batch $B$ **do**
        update:   $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \frac{\eta_n}{|B|} \sum_{k \in B} \nabla f_k(\mathbf{x})$
        adjust   $\eta_n \to \eta_{n+1}$
        $n = n + 1$
    **end for**
**end while**
```

Second-order Methods: Newton method

- expand $f(\mathbf{x})$ at any fixed $\mathbf{x}_0$ according to the Taylor's theorem

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + (\mathbf{x}-\mathbf{x}_0)^\intercal \nabla f(\mathbf{x}_0) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_0)^\intercal \mathbf{H}(\mathbf{x}_0)(\mathbf{x}-\mathbf{x}_0)$$

- $\nabla f(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = 0 \implies \mathbf{x}^* = \mathbf{x}_0 - \mathbf{H}^{-1}(\mathbf{x}_0)\,\nabla f(\mathbf{x}_0)$
- **Newton method** uses the updating rule at each iteration:

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \mathbf{H}^{-1}(\mathbf{x}^{(n)})\,\nabla f(\mathbf{x}^{(n)})$$

- Newton method is not feasible in machine learning since it is too costly to invert a Hessian matrix
- **quasi-Newton methods** aim to approximate the Hessian matrix, e.g. DFP, BFGS, Quickprop, Hessian-free, etc.