Introduction to Algorithms for Data Mining and Machine Learning

Chapter 2: Mathematical Foundations

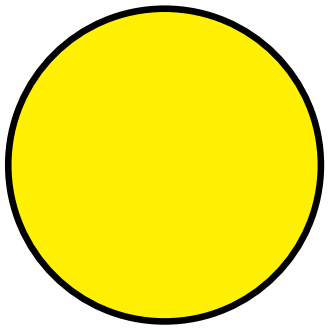
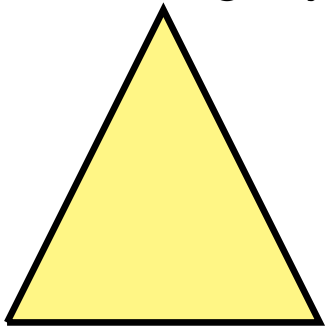
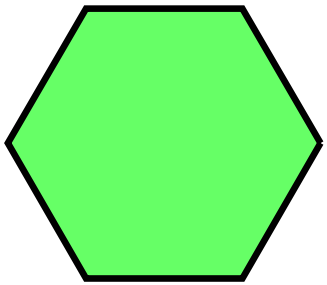
Xin-She Yang

For details, please read the book:

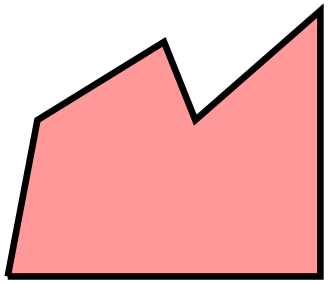
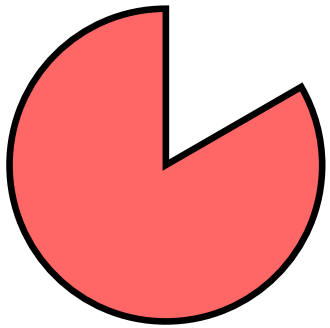
Xin-She Yang, **Introduction to Algorithms for Data Mining and Machine Learning**, Academic Press/Elsevier, (2019).

Convexity

Convex domain: a line linking any two points in the domain should remain in the domain.

Convex domains/shapes

concave domains

Convex Set

A set S in an n-dimensional space is called a convex set if, for any two points (x and y) in S, we have

$$\theta x + (1-\theta)y \in S, \quad \forall x, y \in S, \quad \theta \in [0,1].$$

Convex Function

A function $f(x)$ defined on a convex set Ω is called a convex function if

$$f(\alpha x + \beta y) \le \alpha f(x) + \beta f(y), \quad \forall x, y \in \Omega, \quad \alpha \ge 0, \ \beta \ge 0, \quad \alpha + \beta = 1.$$

To show that $f(x) = x^2 - 1$ is convex, we can show that the inequality always holds for any x and y. That is

$$(\alpha x + \beta y)^2 - 1 \le \alpha(x^2 - 1) + \beta(y^2 - 1), \quad \forall x, y \in \mathbb{R}, \quad \alpha, \beta \ge 0, \ \ \alpha + \beta = 1.$$

Assuming it is true and re-arranging the inequality, we have (using $\alpha + \beta = 1$)

$$\alpha x^2 + \beta y^2 - (\alpha x + \beta y)^2 \ge 0.$$

This is equivalent to

$$\alpha x^2 + \beta y^2 - \alpha^2 x^2 - 2\alpha\beta xy - \beta^2 y^2 = \alpha(1 - \alpha)(x - y)^2 = \alpha\beta(x - y)^2 \ge 0,$$

which is always true. So, the inequality is true and $f(x)$ is indeed convex.

To show that $f(x) = x^2 - 1$ is convex, we can show that the inequality always holds for any x and y. That is

$$(\alpha x + \beta y)^2 - 1 \le \alpha(x^2 - 1) + \beta(y^2 - 1), \quad \forall x, y \in \mathbb{R}, \quad \alpha, \beta \ge 0, \ \ \alpha + \beta = 1.$$

Assuming it is true and re-arranging the inequality, we have (using $\alpha + \beta = 1$)

$$\alpha x^2 + \beta y^2 - (\alpha x + \beta y)^2 \ge 0.$$

This is equivalent to

$$\alpha x^2 + \beta y^2 - \alpha^2 x^2 - 2\alpha\beta xy - \beta^2 y^2 = \alpha(1 - \alpha)(x - y)^2 = \alpha\beta(x - y)^2 \ge 0,$$

which is always true. So, the inequality is true and $f(x)$ is indeed convex.

Property of convex functions

- If f_1 and f_2 are convex, $\alpha f_1 + \beta f_2$ is also convex for any $\alpha, \beta \ge 0$.
- If f_1, f_2, ..., f_n are convex, then $\max\{f_1, f_2, ..., f_n\}$ is also convex.
- If $f(x)$ and $g(x)$ are convex, then $f(g(x))$ is convex under non-decreasing conditions. For example, $\exp[f(x)]$ is convex if $f(x)$ is convex.

Computational Complexity

Order Notation

The main notation in complexity theory is the order notation O. For a given problem size n, $O(n^2)$ means that the calculations take the order of n^2 algebraic operations. So both $10n^2 + 100$ and $5n^2$ are the same order of $O(n^2)$ if n is (usually) large.

Example

The multiplication of two $n \times n$ square matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ has a complexity of $O(n^3)$.

The product $\boldsymbol{C} = \boldsymbol{AB}$ has $n \times n$ entries, and each entry is the sum of the product of a row of $\boldsymbol{A}$ and a column of $\boldsymbol{B}$, which requires n multiplications and $n - 1$ sums. Thus, the complexity of obtaining one entry of $\boldsymbol{C}$ is $O(n)$. So, the total number of algebraic operations is $O(n) \times (n \times n) = O(n^3)$.

Computational Complexity

Order Notation

The main notation in complexity theory is the order notation O. For a given problem size n, $O(n^2)$ means that the calculations take the order of n^2 algebraic operations. So both $10n^2 + 100$ and $5n^2$ are the same order of $O(n^2)$ if n is (usually) large.

Example

The multiplication of two $n \times n$ square matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ has a complexity of $O(n^3)$.

The product $\boldsymbol{C} = \boldsymbol{AB}$ has $n \times n$ entries, and each entry is the sum of the product of a row of $\boldsymbol{A}$ and a column of $\boldsymbol{B}$, which requires n multiplications and $n - 1$ sums. Thus, the complexity of obtaining one entry of $\boldsymbol{C}$ is $O(n)$. So, the total number of algebraic operations is $O(n) \times (n \times n) = O(n^3)$.

Notes

The order notation for complexity mainly concerns the approximate number of calculations. It does not represent the actual computational time that may largely depend on the speed of the computer, implementation details (e.g., vectorization), the programming language used, architecture (e.g., parallelization, cloud), and other factors (such as the underlying operating system and other running programs).

NP-Hard Problems

Class P

If a problem can be solved by a Turing machine (computer) in polynomial time, we say its complexity is $O(n^k)$ where n is the problem size and k is the order. Class P denotes all the problems that can be solved in a polynomial time. In most cases, these problem are considered 'easy' (in computer science), but it does not mean that they can be solved quickly in practice.

For example, the inverse of an $n \times n$ matrix can be $O(n^3)$. If $n = 10^6$, the complexity is $O(n^3) = O(10^{18})$, which can take a very long time. For a computer (with Intel Core-i7), it has about 150×10^9 flops (or 150 gigaflops), which means that it requires $O(10^{18}/(150 \times 10^9)) = O(10^7/1.5)$ seconds or about 77 days.

Class NP

A non-deterministic polynomial-time (NP) hard problem is really difficult to solve because the time tends to increase exponentially with problem size. For example, in the well-known travelling salesman problem (TSP) of visiting each of n cities exactly once, the number of possible combinations is $n!$. For 100 cities, this means $100! = 9.3 \times 10^{157}$ (it takes much longer than the age of the universe by all computers in the world to try all combinations). Many problems in data mining and machine can belong to this class.

Norms

A solution to an n-dimensional optimization problem is represented by a column vector

$$\boldsymbol{x} = (x_1, x_2, ..., x_n)^T = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad x_i \in \mathbb{R},$$

where T is the transpose to convert a row vector into a column vector.

L_p-norm or p-norm

The p-norm of a vector is

$$\|\boldsymbol{x}\|_p = \Big(|x_1|^p + |x_2|^p + ... + |x_n|^p\Big)^{1/p} = \Big(\sum_{i=1}^{n} |x_i|^p\Big)^{1/p}, \quad p > 0.$$

Commonly used norms are $p = 2$ (Cartesian/Euclidean), $p = 1$ and $p = \infty$.

- $p = 2$: The length of $\boldsymbol{x}$ is $\|\boldsymbol{x}\|_2 = \sqrt{x_1^2 + x_2^2 + ... + x_n^2}$.
- $p = 1$: $\|\boldsymbol{x}\| = |x_1| + |x_2| + ... + |x_n|$. [Manhattan norm]
- $p = \infty$: $\|x\|_\infty = \max\{|x_1|, |x_2|, ..., |x_n|\}$. [Maximum norm]

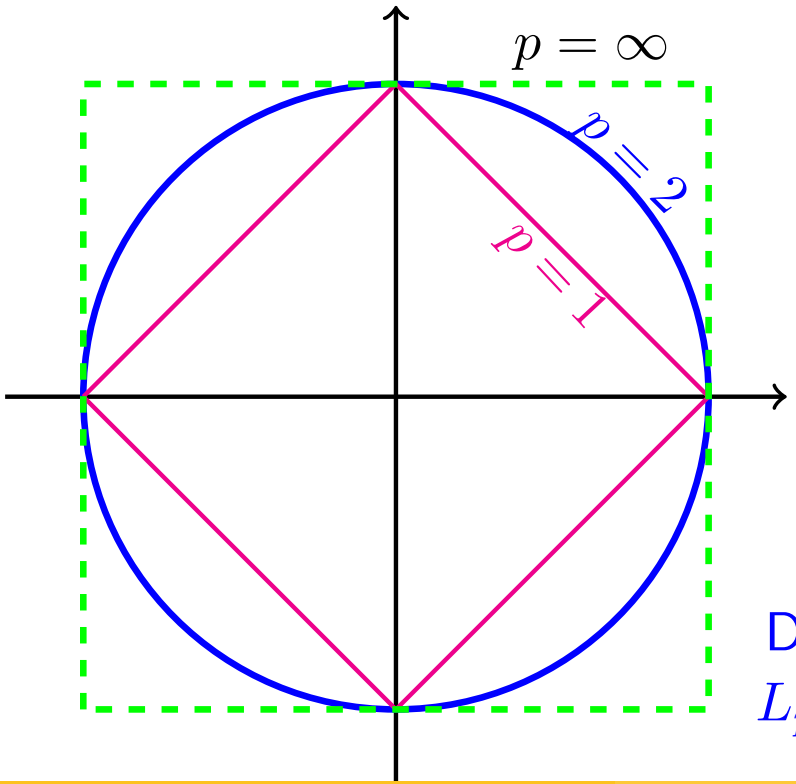
For the vector $\boldsymbol{x} = [2, 3, -1, 7, 9]$, its norms are

$$\|\boldsymbol{x}\|_2 = \sqrt{|2|^2 + |3|^2 + |-1|^2 + |7|^2 + |9|^2} = \sqrt{144} = 12.$$

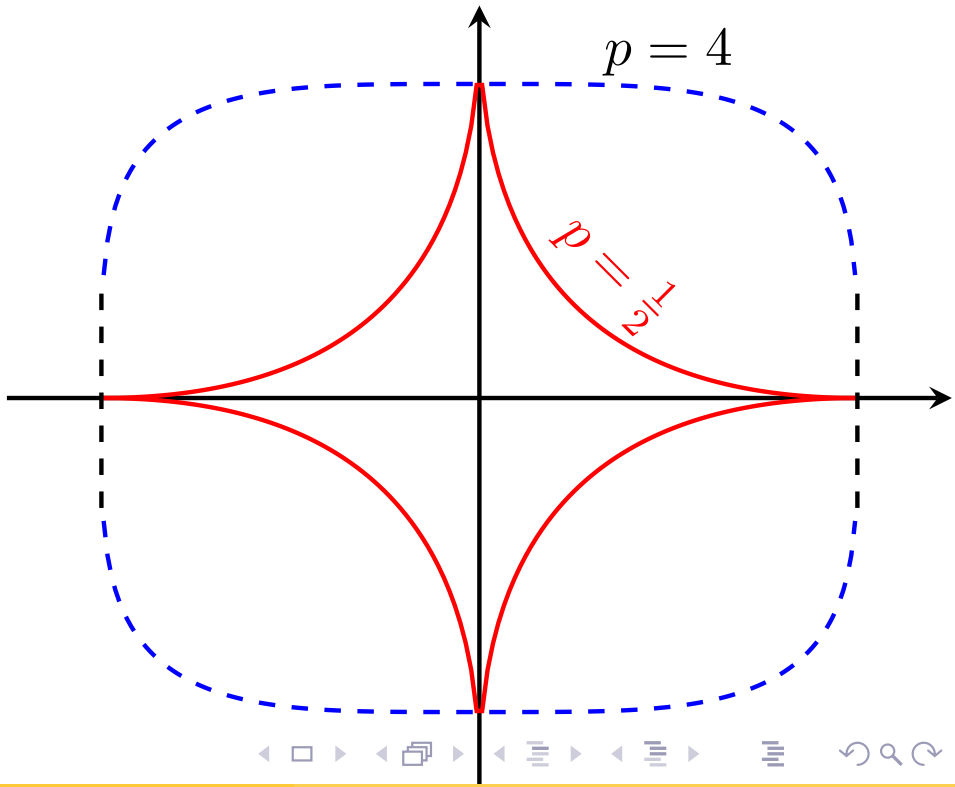
$$\|\boldsymbol{x}\|_1 = |2| + |3| + |-1| + |7| + |9| = 22.$$

$$\|\boldsymbol{x}\|_\infty = \max\{|2|, |3|, |-1|, |7|, |9|\} = 9.$$

Exercise: `Use R to calculate different norms of a vector.`



Different L_p-norms



Eigenvalues and Eigenvectors

For a square real matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$

$$\boldsymbol{A} \equiv [a_{ij}] = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \dots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix},$$

its eigenvalue λ and its corresponding eigenvector $\boldsymbol{u}$ are given by

$$\boldsymbol{A}\boldsymbol{u} = \lambda \boldsymbol{u},$$

which can be calculated by the determinant

$$\det(\boldsymbol{A} - \lambda \boldsymbol{I}) = 0.$$

Since $\boldsymbol{A}$ has a size of $n \times n$, there are usually n eigenvalues λ_i $(i = 1, 2, ..., n)$ and n corresponding eigenvectors $\boldsymbol{u}_i$, though eigenvalues may not be always distinct.

Eigenvalues (Cont'd)

Properties of eigenvalues

If the eigenvalues of $\boldsymbol{A}$ are λ_i $(i = 1, 2, ..., n)$, the eigenvalues of $\boldsymbol{A}^k$ are λ_i^k where k is a positive integer.

Definiteness

- If all the eigenvalues λ_i of a square symmetric $\boldsymbol{A}$ are all positive (i.e., $\lambda_i > 0$), the matrix is called positive definite.
- If $\lambda_i < 0$ for $\forall i$, then $\boldsymbol{A}$ is negative definite.
- In general, if all $\lambda_i \geq 0$, $\boldsymbol{A}$ is positive semi-definite. Conversely, all $\lambda_i \leq 0$ means negative semi-definite.

Example

$\boldsymbol{A} = \begin{pmatrix} 7 & 3 \\ 3 & 7 \end{pmatrix}$ has the eigenvalues of 4 and 10. So, it is positive definite.

$\boldsymbol{B} = \begin{pmatrix} 3 & 7 \\ 7 & 3 \end{pmatrix}$ has the eigenvalues of -4 and 10. Thus, it is neither positive definite nor negative definite.

Exercise: Find the eigenvalues and eigenvectors of a 3×3 matrix using R.

Random Variable

The noise level on a street and the number of calls at a call center are random variables. Noise levels can be considered as continuous, thus the random variable is also continuous. The random variable associated with the number of calls is called a discrete random variable.

- The uppercase X is used to denote the random variable, whereas the lowercase x represents its values of outcomes [e.g., the coin-flipping event has two outcomes: 0 (tail) and 1 (head). That is, $x_i \in \{0, 1\}$].
- A probability $p(x_i)$ is a function that assigns the probability (likeliness) to all the values x_i of a random variable X.
- All probabilities of a random variable X must be summed to 1. That is

$$\sum_i p(x_i) = 1, \quad (i = 1, 2, ...),$$

  where $p(x_i)$ is called the probability mass function (PMF).
- For a continuous variable X, $p(x)$ is called the probability density function (PDF), and the preceding sum becomes an integral

$$\int_\Omega p(x)dx = 1, \quad \Omega = \text{all possible outcomes/event space.}$$

Mean and Variance

- The mean or expectation of a discrete random variable X is the weighted sum

$$\mu = \mathbb{E}[X] = \sum_i x_i p(x_i).$$

- The variance σ^2 is the expectation of the deviation squared. That is

$$\sigma^2 \equiv \text{var}[X] \equiv \mathbb{E}[(X-\mu)^2] = \sum_i (x_i - \mu)^2 p(x_i).$$

The standard deviation σ is simply the square root of the variance.

For a continuous variables, the preceding sums become integrals and we have

$$\mu = \mathbb{E}[X] = \int x p(x) dx, \quad \sigma^2 = \int (x-\mu)^2 p(x) dx.$$

Example

x_i	0	1	2	2.5
$p(x_i)$	0.15	0.3	0.3	0.25

$$\mu = \sum_{i=1}^{4} x_i p(x_i) = 0 \times 0.15 + 1 \times 0.3 + 2 \times 0.3 + 2.5 \times 0.25 = 1.525.$$

$$\sigma^2 = \sum_{i=1}^{4} (x_i - \mu)^2 p(x_i) = 0.736875.$$

Moment

Moment

In general, the kth moment of X is defined by

$$\mu_k = \mathbb{E}[X^k] = \int x^k p(x) dx, \quad (k = 1, 2, ...).$$

The kth central moment of X is defined by

$$\nu_k = \mathbb{E}[(X - \mu)^k] = \int (x - \mu)^2 p(x) dx, \quad (k = 1, 2, ...).$$

Clearly, the mean or expectation is the first moment, whereas the variance is the second central moment.

Exercise: `Write some R codes to calculate the first 3 moments and central moments of` X `in the preceding example.`

Exercise: `Explore the functions/subroutes in R and generate 1000 random samples that obey a uniform distribution and plot out their histogram.`

Probability Distributions

Uniform Distribution

A simple uniform distribution has a probability density function

$$p(x) = \frac{1}{b-a}, \quad a \le x \le b, \;\; b > a > 0,$$

which means that $p(x)$ is simply a constant.

It is easy to show that its mean is $(a+b)/2$ and its variance is $\frac{(b-a)^2}{12}$.

Exponential Distribution

The exponential distribution has the following probability density function

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & (x \ge 0), \quad \lambda > 0, \\ 0, & (x < 0). \end{cases}$$

Its mean and variance are

$$\mu = 1/\lambda, \quad \sigma^2 = 1/\lambda^2.$$

Bernoulli Distribution

For a binary random variable with outcomes of either 1 (success, yes) or 0 (failure or no), the probability of taking $m = 1$ is p (thus the probability of taking $m = 0$ is $q = 1 - p$). The probability mass function can be written as

$$B(m, p) = \begin{cases} p & \text{if } m = 1, \\ 1 - p & \text{if } m = 0. \end{cases}$$

This is equivalent to

$$B(m, p) = p^m q^{1-m} = p^m (1 - p)^{1-m}, \quad m \in \{0, 1\}.$$

It is easy to show that its mean is $\mathbb{E}[X] = p$ and variance is $\text{var}[X] = pq = p(1 - p)$.

Binomial Distribution

Bernoulli distribution concerns a single experiment or trial. For multiple independent trials (n), the probability distribution of exactly m successes is

$$B_n(m, n, p) = \begin{pmatrix} n \\ m \end{pmatrix} p^m (1 - p)^{n-m}, \quad \begin{pmatrix} n \\ m \end{pmatrix} = \frac{n!}{m!(n - m)!},$$

where $n! = n(n - 1)(n - 2)...3 \times 2 \times 1$ is the factorial.

It is straightforward to show that $\mathbb{E}[X] = np$ and $\text{var}[X] = np(1-p)$ for the binomial distribution.

Example: Coin-Flipping

A fair coin is flipped 10 times, what is the probability of showing exactly 7 heads?

As the coin is fair, the probability of showing a head (1) in a single flip is $p = 0.5$ [$q = 0.5$ for tail (0)]. For $n = 10$ flips, the probability of showing $m = 7$ heads is

$$B_{10}(7, 10, 0.5) = \begin{pmatrix} 10 \\ 7 \end{pmatrix} 0.5^7 (1 - 0.5)^{10-7}$$

$$= \frac{10!}{7!(10-7)!} 0.5^{10} = 120 \times 0.5^{10} \approx 0.117.$$

The expected number of heads (mean) and variance are

$$\mathbb{E}[X] = np = 10 \times 0.5 = 5.$$

$$\text{var}[X] = np(1-p) = 10 \times 0.5 \times (1 - 0.5) = 2.5.$$

Exercise: `Use R to generate 100 samples that obey a binomial distribution, and then calculate their mean and variance.`

Poisson Distribution

Poisson distribution can be considered the limit/special case of the binomial distribution for rare, discrete events when p is small. That is $\lambda = np$ is a constant when $p \ll 1$ and $n \gg 1$.

Poisson distribution

The probability density function of the Poisson distribution is

$$p(x) = \frac{\lambda^x e^{-\lambda}}{x!}, \quad \lambda > 0, \quad (x = 0, 1, 2, ...).$$

By convention, $0! = 1$. Both its mean and variance are λ. That is

$$\mathbb{E}[X] = \mathrm{var}[X] = \lambda.$$

Applications

Many processes obey the Poisson distribution, including the number of calls in a call centre per hour, number of cars on a road in an hour, number of visitors to a web page per day, number of goals in a world cup by a team, number of typos on a page and many others.

Example: Emails

Suppose you receive 20 emails per day (on average). What is the probability of getting exactly 4 emails during a 2-hour class?

You have $\frac{20}{24} = \frac{5}{6}$ emails per hour. So for a 2-hour class, $\lambda = \frac{5}{6} \times 2 = \frac{10}{6}$. Thus, the probability of getting (exactly) $x = 4$ emails is

$$p(x = 4) = \frac{\lambda^x e^{-\lambda}}{x!} = \frac{(10/6)^4 e^{-4}}{4!} \approx 0.0607.$$

The probability of no email at all is $p(x = 0) = \frac{(10/6)^0 e^{-10/6}}{0!} = e^{-10/6} = 0.189$.

Example: Emails

Suppose you receive 20 emails per day (on average). What is the probability of getting exactly 4 emails during a 2-hour class?

You have $\frac{20}{24} = \frac{5}{6}$ emails per hour. So for a 2-hour class, $\lambda = \frac{5}{6} \times 2 = \frac{10}{6}$. Thus, the probability of getting (exactly) $x = 4$ emails is

$$p(x = 4) = \frac{\lambda^x e^{-\lambda}}{x!} = \frac{(10/6)^4 e^{-4}}{4!} \approx 0.0607.$$

The probability of no email at all is $p(x = 0) = \frac{(10/6)^0 e^{-10/6}}{0!} = e^{-10/6} = 0.189.$

Example: Defects

A product has a defect rate of (on average) 0.5 out of 100. What is the probability of at least one defect product in a batch of 500?

Based on the description, we know $\lambda = \frac{0.5}{100} \times 500 = 2.5$. The probability of no defect at all $(x = 0)$ is

$$p(x = 0) = \frac{2.5^0 e^{-2.5}}{0!} = \frac{1 \times e^{-2.5}}{1} \approx 0.0821.$$

Thus, the probability of at least one defect is

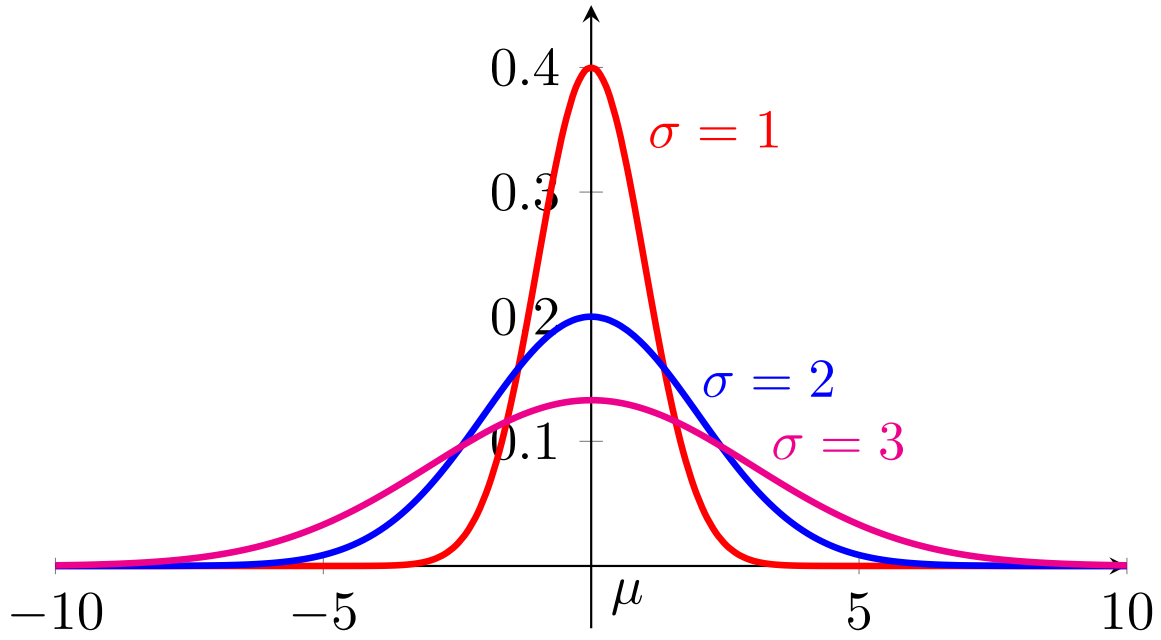
$$p(x \geq 1) = 1 - p(x = 0) = 1 - 0.0821 \approx 0.9179.$$

Gaussian Distribution

The Gaussian distribution or normal distribution is the most widely used distribution. Its probability density function is

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

where σ is the standard deviation and μ is its mean (i.e., $\mu = \mathbb{E}[X]$).



Exercise: `Generate 1000 samples (using R) that obey a normal distribution, and then estimate their sample mean and variance.`

Bayesian Rule

- Two events A and B are independent if they have no influence on each other. The probability $P(A \cap B)$ of both events occurring (their joint probability) is simply the product of the probabilities of each individual event. That is

$$P(A \cap B) = P(A)P(B).$$

- If A and B are not independent, their joint probability is

$$P(A \cap B) = P(B|A)P(A) = P(B \cap A) = P(A|B)P(B),$$

where $P(B|A)$ denotes the conditional probability of event B occurs, given that event A has occurred. Here, "|" means "given that".

Bayes' Theorem or Bayesian Rule

By re-arranging the preceding equation, we have the well-known Bayes' theorem

$$P(B|A) = \frac{P(B \cap A)}{P(A)} = \frac{P(A|B)P(B)}{P(A)}.$$

As the whole event space Ω can be decomposed into B and $\bar{B}$ (not B) (i.e., $\Omega = B \cup \bar{B}$), we have $P(\Omega) = P(B) + P(\bar{B}) = 1$ or $P(\bar{B}) = 1 - P(B)$. We can calculate $P(A)$ by

$$P(A) = P(B)P(A|B) + P(\bar{B})P(A|\bar{B}).$$

Thus we have

$$P(B|A) = \frac{P(B)P(A|B)}{P(A)} = \frac{P(B)P(A|B)}{P(B)P(A|B) + P(\bar{B})P(A|\bar{B})}$$

Terminology

Both $P(A)$ and $P(B)$ are called marginal probabilities. In case of estimating $P(B|A)$ after event A has occurred, $P(B)$ is called a *prior* distribution, which reflects the existing belief in B. Usually, $P(B)$ can be taken as a uniform distribution if there is no prior knowledge.

$P(B|A)$ is called the *posterior* distribution of B by incorporating new knowledge/data about A. The condition probability $P(A|B)$ is called the likelihood of A occurring given that B is true.

Loosely speaking, the Bayesian theorem means that

$$\text{Posterior distribution} \propto \text{likelihood} \times \text{prior distribution}.$$

Example: Test Accuracy

Assume a particular method of (hypothetical) drug testing can have an accuracy of 99% if athletes are taking drugs. For athletes not taking drugs, the positive test is only 0.5%.

In a particular sport event, suppose only 1 in 1000 athletes may take this kind of drug. Now suppose an athlete is selected at random and the test shows positive for the drug, what is the probability of the athlete is actually taking the drug?

Example: Test Accuracy

Assume a particular method of (hypothetical) drug testing can have an accuracy of 99% if athletes are taking drugs. For athletes not taking drugs, the positive test is only 0.5%.

In a particular sport event, suppose only 1 in 1000 athletes may take this kind of drug. Now suppose an athlete is selected at random and the test shows positive for the drug, what is the probability of the athlete is actually taking the drug?

If B denotes an athlete taking the drug, and A denotes the event that the individual test is positive. We have

$$P(B) = 1/1000 = 0.001, \quad P(A|B) = 0.99, \quad P(A|\bar{B}) = 0.005.$$

Thus, the probability that the athlete is actually taking the drug is

$$P(B|A) = \frac{P(B)P(A|B)}{P(A)} = \frac{P(B)P(A|B)}{P(B)P(A|B) + P(\bar{B})P(A|\bar{B})}$$

$$= \frac{0.001 \times 0.99}{0.001 \times 0.99 + 0.999 \times 0.005} \approx 0.165,$$

which is surprisingly a low probability.

Statistical methods based on the Bayesian rule is called Bayesian statistics with many applications. Bayesian inferencing is a powerful technique for machine learning.

Monte Carlo Sampling

A Markov chain is sequence of random variables that its next state will only depend on the current state and the transition probability, independent of its past history. [For example, tomorrow's weather (sunny, cloudy, etc.) will depend on the weather today (current state) and how it may change (transition), not the weather yesterday/last week.]

Mathematically, the state s of random variable X at next step $k+1$ only depends on its state at the current step k. That is

$$P(X_{k+1}|X_k = s_k, X_{k-1} = s_{k-1}, ..., X_1 = s_1) = P(X_{k+1} = s|X_k = s_k).$$

All the past states $(s_{k-1}, ..., s_2, s_1)$ are irrelevant and do not appear in the equation.

Property of Markov Chains

A useful property of a Markov chain is that its long-term state as $k \to \infty$ will 'forget' their initial states, leading to a stationary distribution $\pi(\theta)$ for parameter θ.
To estimate (a random parameter) θ, we need two things: the current value θ (current state) and the transition probability (called transition kernel) from θ to a new estimate (state) θ'. That is

$$T(\theta, \theta') = p_{\theta \to \theta'}(\theta'|\theta) = p(\theta'|\theta).$$

The stationary distribution $\pi(\theta)$ satisfies the following (symmetric) balance condition:

$$T(\theta, \theta')\pi(\theta) = T(\theta', \theta)\pi(\theta').$$

Monte Carlo Markov Chain (MCMC)

Monte carlo is a random sampling method that is widely used. A powerful Monte Carlo sampling method is the Monte Carlo Markov Chain (MCMC), using the property of the long-term stationary distribution $\pi(\theta)$.

To estimate the distribution of θ properly, we need

- A proposal distribution $q(\theta, \theta')$, such as a uniform or Gaussian distribution, (acting as a jump distribution or transition), to propose a new θ' from θ
- A criterion to accept or reject the move from θ' to θ.

Metropolis-Hastings (MH) Algorithm

To draw random numbers/samples that obey distribution $p(\theta)$, the MH algorithm starts with $\theta = \theta_0$ (a random guess) at $k = 1$, and draws a candidate θ_* from the proposal distribution $q(\theta_{k-1}, .)$. Then, calculate the ratio

$$r = \min\left\{\frac{p(\theta_*)q(\theta_*, \theta_{k-1})}{p(\theta_{k-1})q(\theta_{k-1}, \theta_*)}, 1\right\}.$$

The move from θ_{k-1} to the candidate θ_* is accepted with a probability r such that $\theta_k \leftarrow \theta_*$; otherwise, the move is discarded and no change is made (i.e., $\theta_k \leftarrow \theta_{k-1}$).

When running the MN algorithm, θ is typically a vector of multiple dimensions.

Algorithm 1: Metropolis-Hastings Algorithm

Data: Probability distribution $p(\theta)$ to be estimated

Initial guess θ_0 at $k = 1$;
Choose a proposal distribution $q(.,.)$;
for *(a given number of samples)* **do**
    Propose a candidate θ_* by drawing from $q(\theta_{k-1}, .)$;
    Calculate the ratio $r = \min\left\{\frac{p(\theta_*)q(\theta_*,\theta_{k-1})}{p(\theta_{k-1})q(\theta_{k-1},\theta_*)}, 1\right\}$;
    General a random number u uniformly distributed in [0,1];
    if $u < r$ **then**
        Accept the move $\theta_k \leftarrow \theta_*$;
    else
        Reject the move and set $\theta_k \leftarrow \theta_{k-1}$;
    end
    Update counter $k \leftarrow k + 1$;
end

If the proposal distribution $q(a, b)$ is symmetric such that $q(a, b) = q(b, a)$, then r becomes

$$r = \min\left\{\frac{p(\theta_*)}{p(\theta_{k-1})}, 1\right\}.$$

The MH algorithm becomes the classic Metropolis algorithm. For example, we can use the normal distribution N(0,1) as the proposal distribution $q(a, b) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(b-a)^2}$.
If $r = 1$ is used, we have the well-known Gibbs sampler (with many software packages).

Notes

MCMC

When using MCMC simulations such as the MH algorithm, the desired samples are reached as $k \to \infty$. Thus, most simulations will discard some initial samples for a certain period (called, burn-in period). However, the choice of such burn-in period is mostly empirical, and thus care should be taken in the final samples. Some statistical tests may be useful to see if the samples can indeed obey the desired probability distribution.

Notes on Software

Many software packages (e.g., R, Python and Matlab/Octave) have implemented pseudo-random number generators that can generate most of the commonly used probability distributions such as uniform, Gaussian, exponential and many others.

Exercise: `Explore R functionalities and try to draw 1000 points/samples that obey a Poisson distribution with` $\lambda = 4$. `Visualize your samples using histograms and test if they indeed obey the desired distribution.`

Entropy

Entropy

For a given probability distribution $p(x)$, the Shannon entropy H is defined by

$$H(p) = -\sum_i p(x_i) \log[p(x_i)], \quad \text{(discriete)},$$

$$H(p) = -\int p(x) \log[p(x)] dx, \quad \text{(continuous)}.$$

The log is usually in base 2. Since $0 < p \leq 1$, $H(p) \geq 0$ measures the (average) amount of information in the distribution.

Cross Entropy

For two distributions $p(x)$ and $q(x)$, the cross entropy $H(p, q)$

$$H(p, q) = -\int p(x) \log[q(x)] dx$$

measures the distance or similarity/dissimilarity between $p(x)$ and $q(x)$.

KL Divergence

The Kullback-Leibler (KL) divergence measures the difference between $p(x)$ given $q(x)$. That is

$$D_{KL}(p,q) \equiv D_{KL}(p||q) = \int p(x) \log \left[\frac{p(x)}{q(x)}\right] dx,$$

which can also be written as

$$D_{KL}(p,q) = \int p(x) \log[p(x)] dx - \int p(x) \log[q(x)] dx = H(p,q) - H(p).$$

It is the difference between the cross entropy and Shannon entropy. Clearly, if $p(x) = q(x)$, the KL divergence becomes zero.

Exercise: For two distributions $p = [0.05, 0.15, 0.2, 0.2, 0.2, 0.15, 0.05]$ and $q(x) = [0.1, 0.1, 0.1, 0.4, 0.1, 0.1, 0.1]$, calculate $H(p)$, $H(q)$, $H(p,q)$ and $D_{KL}(p,q)$.

References

References

- Xin-She Yang, **Introduction to Algorithms for Data Mining and Machine Learning**, Academic Press/Elsevier, (2019).
- Xin-She Yang, **Optimization Techniques and Applications with Examples**, John Wiley & Sons, (2018).

Notes on Software

There are many different software packages for optimization, data mining and machine learning, including R, Pathon and Matlab implementations.

- For symbolic computation and mathematics, free software packages are Axiom, Maxima, SymPy and others (list).
- Data Mining and Machine Learning Software
- For the Gibbs sampler, please see WinBUGs

Any questions? **Thank you :)**