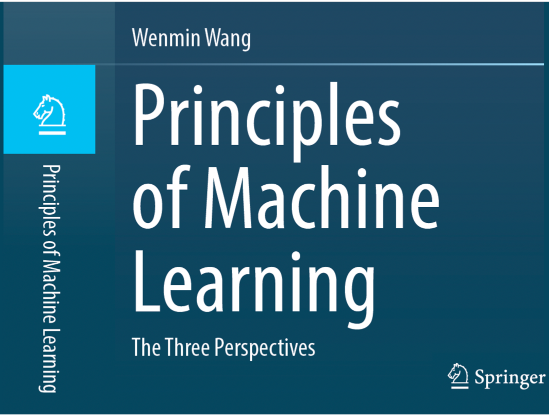


Wenmin Wang

Principles of Machine Learning

The Three Perspectives

Springer

Principles of Machine Learning

Part II Frameworks

3 Probabilistic Framework

4 Statistical Framework

5 Connectionist Framework

6 Symbolic Framework

7 Behavioral Framework

3 Probabilistic Framework

Contents

3.1 Overview

3.2 Basics of Probability Theory

3.3 Computational Learning Theory

3.4 Bayesian Models

3.5 Markov Models

3.6 Probability Models

3.7 Probabilistic Graphical Models

3.8 Monte Carlo Methods

About the Probabilistic Framework

- The framework comes in handy when we need to quantify uncertainty or stochasticity in observed data or environment of machine learning.
- The theoretical foundation of the framework is probability theory.
- The core of the framework is probability learning theory, especially computational learning theory.

Probability Theory

- What is probability theory:
  - a branch of mathematics about uncertain problems and random phenomena.
- Where is it used in machine learning:
  - used to study the inherent laws in uncertainty and stochasticity,
  - used to analyze the possibility of various outcomes based on the theory.
- The two parties in probability theory:
  - Frequentist:
    the probability is the limit of the relative frequency of an event after many trials.
  - Bayesian:
    the probability is simply a measure of a degree of belief in an event.

Probability Learning Theory

- Why does it need probability learning theory:
  - Because machine learning must always deal with uncertain quantities, and sometimes may also need to deal with stochastic (non-deterministic) quantities. Uncertainty and stochasticity can arise from many sources.
- The contents of probability learning theory:
  - Probably approximately correct (PAC) learning.
  - PAC-Bayesian learning.
  - Bayesian Occam’s learning.
  - Probabilistic programming for machine learning.

Probability Space

Definition: (Probability Space)

A probability space is defined as a 3-tuple $\langle \Omega, \mathcal{F}, P \rangle$, Where, Ω denotes sample space, $\mathcal{F}$ denotes event space, and P is probability function.

- Ω is a sample space, a set of all possible outcomes in an experiment.
- $\mathcal{F}$ is an event space, $\mathcal{F} \subseteq 2^{\Omega}$, the collection of all subsets of Ω.
- P is a probability function on $\langle \Omega, \mathcal{F} \rangle$, $P : \mathcal{F} \to [0, 1]$. $\forall A, B \in \mathcal{F}$:

  $P(A) \geq 0$. $P(\Omega) = 1$. If $A \cap B = \emptyset$, then $P(A \cup B) = P(A) + P(B)$.
- On the basis of above three axioms, following results can be obtained,

  $P(\emptyset) = 0$. $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.

Random Variables

Definition: (Random Variables)

A random variable X on the probability space is a measurable function, $X : \Omega \rightarrow \Sigma$, where Σ be a measurable space.

- Σ: a real-valued space in many cases, i.e., $\Sigma = \mathbb{R}$.
- $P(X = a)$: the probability of a random variable X taking on the value of a.
- $X = a$: the random variable X taking on the value of a, where $a \in \Sigma$.
- $\mathrm{Val}\left(X\right)$: the range of the random variable X, e.g., $a \in \mathrm{Val}\left(X\right)$.

Examples

Examples for Probability Space:

Throwing a dice, then

- $\Omega = \{1, 2, 3, 4, 5, 6\}$, with six sides that are numbered 1 to 6.
- $\mathcal{F} = \{\emptyset, \{1, 3, 5\}, \{2, 4, 6\}, \Omega\}$, for the events of odd or even.

Examples for Random Variable:

Throwing a dice, if

- $\mathrm{Val}\left(X\right) = \{1, 2, 3, 4, 5, 6\}$ refers to the range of X is all number, then $X = 3$ means the number on dice is 3.
- $\mathrm{Val}\left(X\right) = \{1, 0\}$ refers to the range of X is where odd or even, then $X = 3$ means the number on dice is odd.

Discrete vs Continuous

Discrete

- Discrete random variable: if the range of a random variable X is countable.
- Discrete distribution: for a discrete random variable X, the probability mass function:

$$P_X(x) = P(X = x)\,, \text{where } x \in \mathrm{Val}(X)\,, \text{and } \sum_x P_X(x) = 1.$$

Continuous

- Continuous random variable: if the range of a random variable X is uncountable.
- Continuous distribution: for a continuous random variable X, the probability density function:

$$P(a \leq X \leq b) = \int_a^b p(x)\,dx, \text{and } \int_{\mathrm{Val}(X)} p(x)\,dx = 1.$$

Probability Relationship

Conditional Probability

- Let $\forall X, Y \in \mathcal{F}$, and $P(Y) \neq 0$, then

$$P(X|Y) = \frac{P(X \cap Y)}{P(Y)}.$$

Joint Probability

- For two random variables $P(X, Y) = P(X \cap Y)$, then

$$P(X, Y) = P(X|Y)\, P(Y)\,.$$

- For n random variables, then

$$\begin{aligned} P(X_1, \ldots, X_n) &= P(X_1)\, P(X_2|X_1)\, P(X_3|X_2, X_1) \cdots P(X_n|X_{n-1}, \ldots, X_1) \\ &= P(X_1) \prod_{i=2}^{n} P(X_i|X_{i-1}, \ldots, X_1). \end{aligned}$$

Expectations

The expectation (or expected value) of X is the probability-weighted average.

Expectation of discrete random variable

$$\mathbb{E}\left[X\right] = \sum_{a \in X} a \cdot P\left(X = a\right).$$

where, $P\left(\cdot\right)$ is a probability mass function.

Expectation of continuous random variable

$$\mathbb{E}\left[X\right] = \int_{a \in X} x \cdot p\left(x\right) dx,$$

where, $p\left(\cdot\right)$ is a probability density function.

Variance and Standard Deviation

Variance

It’s the expectation of the squared deviation of a random variable from its average.

$$\operatorname{var}(X) = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right].$$

Standard Deviation

The standard deviation is usually denoted by σ, its relationship with variance is:

$$\sigma = \sqrt{\operatorname{var}(X)}.$$

So, variance $\operatorname{var}(X)$ can also be denoted by σ^2.

Independent and Identically Distributed

Definition: Independent and Identically Distributed

A set of random variables is called independent and identically distributed (i.i.d.), only if each random variable within it has the same probability distribution and the random variables are mutually independent.

About Computational Learning Theory

- Computational learning theory is a mathematical analysis theory about learnability, mainly used for the design and analysis of machine learning algorithms.
- According to this theory, if a learning algorithm can complete its learning task in polynomial time, it is considered feasible.
- Computational learning theory originated from learnability theory, and later evolved into probably approximately correct (PAC) learning, as well as Occam learning.
- An important idea in computational learning theory is the introduction of concepts from computational theory into machine learning. Therefore, it can be said:

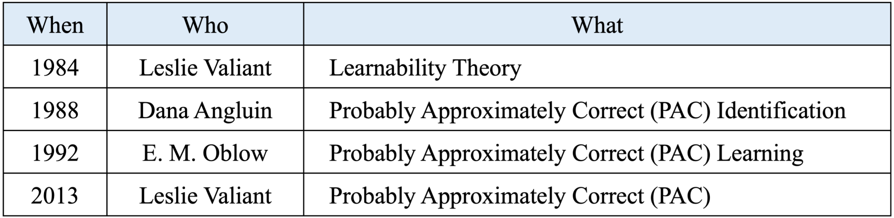
Computational learning theory is to machine learning, as computational theory is to computer science.

Learnability Theory

- There is a rich theory of computability in computer science, but no corresponding theory to explain why machines could learn before 1984.
- Leslie Valiant, a computer scientist, published a paper titled ”A Theory of the Learnable” in *Communications of the ACM*, in 1984.
- Learnability theory is the theoretical basis of computational learning theory, that concerns the questions such as:
  - why is machine learnable,
  - what information is required to support learning,
  - what computation is required for learning to be possible.
- Two components in learning machines by the theory:
  - Learning protocol: the manner in which information is obtained from the outside.
  - Deduction procedure: the algorithm for the concept to be learned is deduced.

PAC Learning

Historical Development



When	Who	What
1984	Leslie Valiant	Learnability Theory
1988	Dana Angluin	Probably Approximately Correct (PAC) Identification
1992	E. M. Oblow	Probably Approximately Correct (PAC) Learning
2013	Leslie Valiant	Probably Approximately Correct (PAC)

What is PAC learning

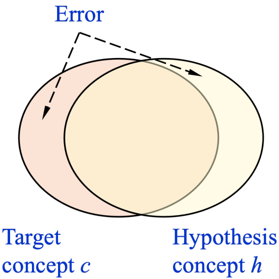
Be a framework for mathematical analysis of machine learning.

- Learner receives samples and must select a hypothesis from possible functions.
- Goal: with high probability (probably), the hypothesis will have low generalization error (approximately correct).

Learning Model

Let $\mathcal{X}$ be the sample space, and C be the concept class. Let $\boldsymbol{x}_i \in \mathcal{X}$ be a sample, and $c \in C$ be a target concept. Without loss of generality, let c be a binary function, $c : \mathcal{X} \to \{0, 1\}$. That is: if $\boldsymbol{x}_i$ is a positive sample, then $c(\boldsymbol{x}_i) = 1$; otherwise $c(\boldsymbol{x}_i) = 0$.

Let training sample $S = \{(\boldsymbol{x}_i, c(\boldsymbol{x}_i)) \mid i = 1, \ldots, n\}$.



First, finding a hypothesis set $H = \{h \mid h : \mathcal{X} \to \{0, 1\}\}$ that approximates the target concept, where h is called a hypothesis concept, and has a certain error with c.

Then, using S for training to minimize the error between h and c, that is, h is approximately equal to c with high probability.

This is why it is called “probably approximately correct (PAC)” learning.

PAC Learning Model

PAC Learning

A concept class C is said to be PAC learnable, if there exists an algorithm $\mathcal{A}$ and a polynomial function $\text{poly}(\cdot, \cdot, \cdot, \cdot)$, such that for any $\epsilon > 0$ and $\delta > 0$, for all distributions D on $\mathcal{X}$, any target concept $c \in C$, and any sample size $m \geq \text{poly}\left(1/\epsilon, 1/\delta, n, \text{size}\left(c\right)\right)$, the following expression holds:

$$P_{S \sim D^n}\left[R\left(h\right) \leq \epsilon\right] \geq 1 - \delta.$$

If $\mathcal{A}$ further runs in $\text{poly}(1/\epsilon, 1/\delta, n, \text{size}\left(c\right))$, then C is said to be efficiently PAC learnable.

When such an algorithm $\mathcal{A}$ exists, it is called a PAC learning algorithm for C.

Occam’s Razor

Occam’s razor is a law of parsimony,

> “Simpler solutions are more likely to be correct than complex ones”.

The idea is attributed to William of Occam (1287-1347), who was born in Occam, Surrey, England, and became a Franciscan Friar, scholastic philosopher, and theologian. William of Occam made drastic reforms to the obscure and lengthy explanations of his scholastic philosophy predecessors, so he was praised as “Occam’s razor”.

Similarly, in the scientific field, Occam’s razor is used as an abductive heuristic rule for developing theoretical models.

In 1987, Anselm Blumer et al. published a paper “Occam’s razor” in *Information Processing Letters*, proposed the principle is very much alive in machine learning.

Occam Learning Framework

Occam Learning

Let C be the concept class containing the target concept $c \in C$, and H be the hypothesis set. Then for constants $\alpha \geq 0$ and $0 \leq \beta \leq 1$, the learning algorithm $\mathcal{A}$ is an α-β Occam algorithm that uses H to learn C if and only if:

given a set of n samples $S = \{\boldsymbol{x}_i\}_{i=1}^{n}$, uses the concept $c(\boldsymbol{x})$ for labeling, we get n training samples $\{(\boldsymbol{x}_i, c(\boldsymbol{x}_i))\}_{i=1}^{n}$, which are used to train the learning algorithm $\mathcal{A}$ to get a hypothesis $h \in H$, so that

- h on S is consistent with c, that is, $\forall \boldsymbol{x} \in S$, $h(\boldsymbol{x})$ is consistent with $c(\boldsymbol{x})$.
- $\mathrm{size}(h) \leq (m \cdot \mathrm{size}(c))^{\alpha} n^{\beta}$.

Where m is the maximum length for any sample $\boldsymbol{x} \in S$.

Bayes Theorem

Let X and Y be discrete random variables, then Bayes theorem is expressed as:

$$P\left(Y|X\right) = \frac{P\left(X|Y\right) P\left(Y\right)}{P\left(X\right)}.$$

Where, $P\left(X|Y\right)$ is likelihood probability, $P\left(Y\right)$ is prior probability, $P\left(X\right)$ is evidence, and $P\left(Y|X\right)$ is posterior probability.

Bayes theorem can be written as the following equation:

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}.$$

That is to say, if the Prior and Evidence are known, and the Likelihood is also known, then its Posterior can be obtained.

For continuous random variables, the Bayes theorem also holds.

Bayesian Learning

People often want to infer the cause based on the known effect. According to Bayes theorem, it can be represented in the following form:

$$P\left(\mathrm{Cause}|\mathrm{Effect}\right) = \frac{P\left(\mathrm{Effect}|\mathrm{Cause}\right)P\left(\mathrm{Cause}\right)}{P\left(\mathrm{Effect}\right)}.$$

Where, $P\left(\mathrm{Effect}|\mathrm{Cause}\right)$ is the likelihood of causal relationship, $P\left(\mathrm{Cause}\right)$ is the prior of cause, and $P\left(\mathrm{Effect}\right)$ is the evidence of effect. If these conditions are known, the posterior $P\left(\mathrm{Cause}|\mathrm{Effect}\right)$ can be inferred.

This is the theoretical basis of Bayesian learning.

For machine learning, let hypothesis set be H, to get a hypothesis $h \in H$, satisfying $h : X \to Y$, if expressed in conditional probability, it is $h \sim P\left(Y|X\right)$. Under the conditions that satisfy Bayes theorem, corresponding results can be obtained.

Stochastic Process and Markov Property

Stochastic Process

A stochastic process (SP) is usually defined as a set of random variables on a probability space:

$$\mathrm{SP} = \{S(t) \mid t \in T\} .$$

Where: T is an index set, usually interpreted as time, and each t is considered a point in time; $S\,(t)$ is known as the state of the stochastic process at time t.

Markov Property

Given a stochastic process, if the next state $S(t+1)$ only depends on the current state $S(t)$ and is irrelevant to the past states $\{S(0), \ldots, S(t-1)\}$, then the stochastic process is said to have Markov property. It is expressed as:

$$P(S(t+1) \mid S(0), \ldots, S(t-1), S(t)) = P(S(t+1) \mid S(t)).$$

Markov Process

A Markov process (MP) is a stochastic process with the Markov property, which can be represented as a 2-tuple: $\mathrm{MP} = \langle S, P \rangle$. Where, S is a state set, and P is the transition probability of the current state transitioning to the next state.

A Markov process with a discrete index set is expressed as:

$$P\left(s_{t+1} | s_t\right) = \mathbb{P}\left[S\left(t+1\right) = s_{t+1} | S\left(t\right) = s_t\right].$$

Where $\mathbb{P}[\cdot]$ denotes the transition matrix, that is:

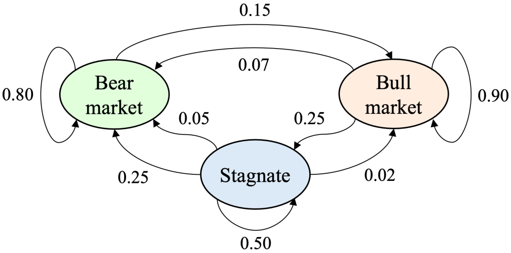
$$\mathbb{P} = \begin{bmatrix} P_{00} & P_{01} & \cdots & P_{0t} \\ P_{10} & P_{11} & \cdots & P_{1t} \\ \vdots & \vdots & \ddots & \vdots \\ P_{t0} & P_{t1} & \cdots & P_{tt} \end{bmatrix}.$$

Markov Chain

A Markov chain is a Markov process with a sequence of discrete random variables, which defines its serial dependencies only in adjacent states and their change periods, as if in a "chain". $\forall t$, $S\left(t\right) = s_t$ forms a countable set of values:

$$S(0) = s_0,\ S(1) = s_1,\ \cdots,\ S(t) = s_t,$$

that is referred to as a chain state space.



A specific Markov chain is defined by $P\left(S(t+1) = s_{t+1} \mid S(t) = s_t\right)$ that satisfies the Markov property.

Be usually represented by a directed graph, where each edge is marked as the probability from $S(t)$ to $S(t+1)$.

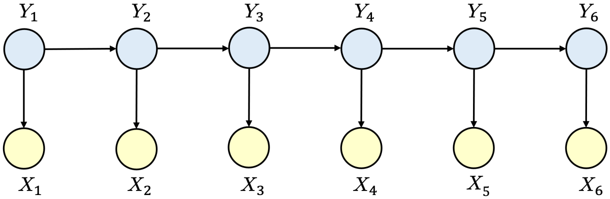
Hidden Markov Model

A hidden Markov model (HMM) is a Markov model with hidden (unobservable) states.

Let $\{Y_i\}_{i=1}^{n}$ and $\{X_i\}_{i=1}^{n}$ be discrete-time stochastic processes, then the set of pair $(\{Y_i\}, \{X_i\})_{i=1}^{n}$ is a hidden Markov model, if it satisfies the following two conditions:

- Y_i is a Markov process with hidden states.
- $P(X_i \mid Y_1 = y_1, \ldots, Y_{i-1} = y_{i-1}, Y_i = y_i) = P(X_i \mid Y_i = y_i)$.

Where $P(X_i \mid Y_i = y_i)$ is referred to as the output probability.

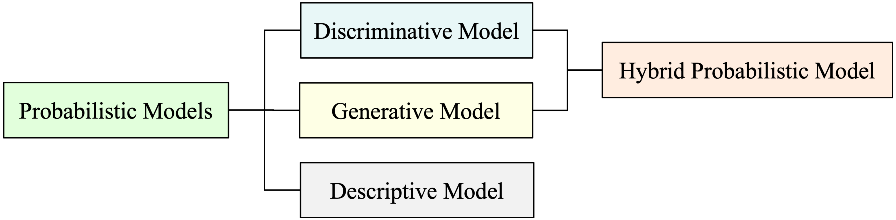


Probability Models

A probability models incorporates random variables and probability distributions into the model of events.

Three probabilistic models in machine learning:

- discriminative model
- generative model
- descriptive model



Discriminative Models

Discriminative models, also known as conditional models, are used to complete classification or regression tasks in supervised learning.

Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in X$ be input, $y_1, \ldots, y_n \in Y$ be target output, $S = \{(\boldsymbol{x}_i,\ y_i) \mid i = 1, \ldots, n\}$ is training data. It models the decision boundary directly using the conditional probability $P\left(Y|X\right)$.

Example: Logistic regression

Let $w_1, \ldots, w_n \in W$ be parameters, it is directly estimated from the training data S:

$$P\left(Y = 1 \mid X, W\right) = \frac{1}{1 + \exp\left(w_0 + \sum_{i=1}^{n} w_i \boldsymbol{x}_i\right)},$$

$$P\left(Y = 0 \mid X, W\right) = \frac{\exp\left(w_0 + \sum_{i=1}^{n} w_i \boldsymbol{x}_i\right)}{1 + \exp\left(w_0 + \sum_{i=1}^{n} w_i \boldsymbol{x}_i\right)}.$$

Another typical algorithm is conditional random field (CRF).

Generative Models

The generative models assume that their results are produced by some latent variables through deterministic transformations, and can be learned in unsupervised manner.

Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in X$ be input, $y_1, \ldots, y_n \in Y$ be output. A generative model is manifested as joint probability distribution $P(X, Y)$, that is:

$$P(Y|X) = \frac{P(X, Y)}{P(X)} = \frac{P(X|Y)P(Y)}{P(X)}.$$

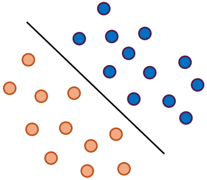
Therefore, we have:

$$P(Y = 1 \mid X = \boldsymbol{x}) = \frac{P(X = \boldsymbol{x}, Y = 1)}{P(X = \boldsymbol{x})} = \frac{P(X = \boldsymbol{x} \mid Y = 1)P(Y = 1)}{P(X = \boldsymbol{x})},$$

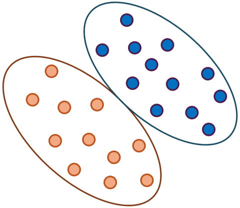
$$P(Y = 0 \mid X = \boldsymbol{x}) = \frac{P(X = \boldsymbol{x}, Y = 0)}{P(X = \boldsymbol{x})} = \frac{P(X = \boldsymbol{x} \mid Y = 0)P(Y = 0)}{P(X = \boldsymbol{x})}.$$

Discriminative Models vs. Generative Models

Discriminative Models	Generative Models
• Learning the boundary between classes • Computing conditional probability $P(Y\|X)$	• Modelling the distribution for each classe • Computing joint probability $P(X,Y)$



Discriminative model



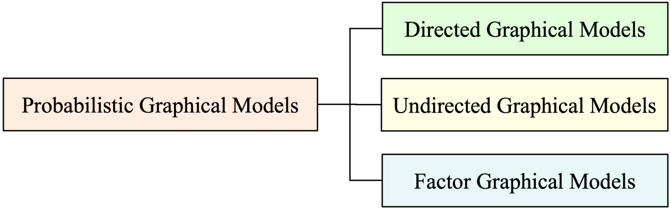
Generative model

Probabilistic Graphical Models

Probabilistic graphical models (PGMs) are also known as graphical models or structured probabilistic models.

PGMs combine probability theory and graph theory, and use graphs to represent the conditional dependency structure between random variables.

Three types of PGMs: directed graphical models, undirected graphical models, and factor graph models.



Probabilistic Graphical Models

Directed Graphical Models

Undirected Graphical Models

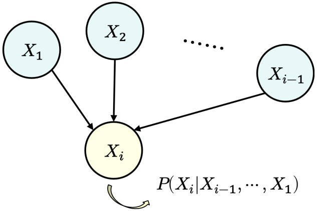
Factor Graphical Models

Directed Graphical Models

A directed graphical model (DGM) is an ordered 2-tuple, $\mathrm{DGM} = \langle \mathrm{DG}, P_{\mathrm{DG}} \rangle$, where, DG is a directed graph, P_{DG} is a set of probability distributions for the directed graph, $P_{\mathrm{DG}} = \{P_{X_i} | X_i \in X\}$.

The directed graph models are suitable for modeling problems of unidirectional dependencies between random variables.

Example: Bayesian networks



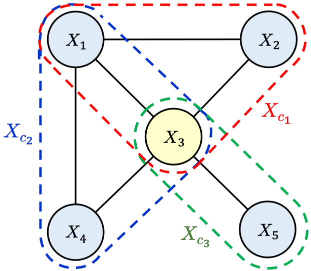
A Bayesian network (BN) is an ordered 2-tuple, $\mathrm{BN} = \langle \mathrm{DAG}, P_{\mathrm{BN}} \rangle$, where: the DAG is a directed acyclic graph, each probability $P_{X_i} \in P_{\mathrm{BN}}$ is a conditional probability of the following form:

$$P_{X_i} = P\left(X_i | X_{i-1}, \ldots, X_1\right) = P\left(X_i | \mathrm{Parents}\left(X_i\right)\right).$$

Undirected Graphical Models

An undirected graphical model (UGM) is an ordered 2-tuple, $\mathrm{UGM} = \langle \mathrm{UG}, P_{\mathrm{UG}} \rangle$, where, UG is an undirected graph, $P_{\mathrm{UG}} = \{\phi_{c_k}(X_{c_k}) \,|X_{c_k} \subseteq X \text{ and } k = 1, \ldots, K\}$, where: X_{c_k} is a clique composed of several random variables; K is the maximum number of cliques; $\phi_{c_k}(X_{c_k})$ is a non-negative potential function, $\phi_{c_k} : \mathrm{Val}(X_{c_k}) \to \mathbb{R}_+$.

Example: Markov networks



A Markov network (MN) is an ordered 2-tuple $\mathrm{MN} = \langle \mathrm{UG}, P_{\mathrm{MN}} \rangle$, where UG is an undirected graph, P_{MN} is a joint probability distribution as follows:

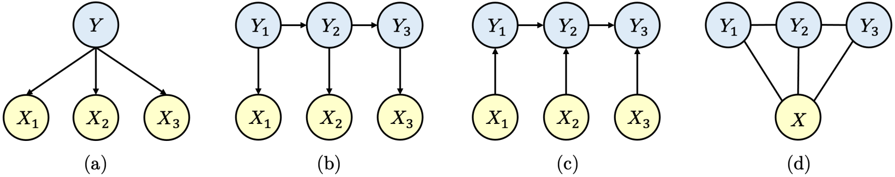
$$P_{\mathrm{MN}}(X_1, \ldots, X_n) = \frac{1}{Z_{\mathrm{MN}}} \prod_{k \in K} \phi_{c_k}(X_{c_k}).$$

$$Z_{\mathrm{MN}} = \sum_{X_i \in \mathcal{X}} \prod_{k \in K} \phi_{c_k}(X_{c_k}).$$

Typical Directed and Undirected Graphical Models

The typical directed graph models are Bayesian network, naive Bayes classifier, hidden Markov model (HMM), and maximum entropy Markov model (MEMM).

The undirected graph model is Markov network, also known as the Markov random field (MRF), and the conditional random field (CRF) is its special case.



Where, (a) naive Bayes classifier, (b) Hidden Markov model,
(c) Maximum entropy Markov model, and (d) Conditional random fields model.

Monte Carlo Methods

Monte Carlo methods, also known as Monte Carlo approximations or Monte Carlo simulations.

Basic idea: when a problem is expressed as the probability of a random event or the expected value of a random variable, it can be estimated through repeated random sampling and used as an approximate solution.

Be a general term for a broad class of algorithms, including: Monte Carlo integration, rejection sampling, adaptive rejection sampling, importance sampling, sampling importance resampling, Gibbs sampling, slice sampling, Markov chain Monte Carlo, Hamiltonian Monte Carlo, and Langevin Monte Carlo.
Monte Carlo methods have been played important roles in artificial intelligence and machine learning.

Why Called Monte Carlo

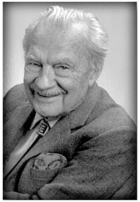
In late-1940s, Stanislaw Ulam, John von Neuman, and Nicholas Metropolis, were worked for Manhattan Project.

The idea of Monte Carlo method was proposed by Ulam, and implemented by von Neuman. Being secret, the work required a code name.

Metropolis suggested using the name Monte Carlo, which refers to the Monte Carlo Casino in Monaco where Ulam's uncle would go to gamble.

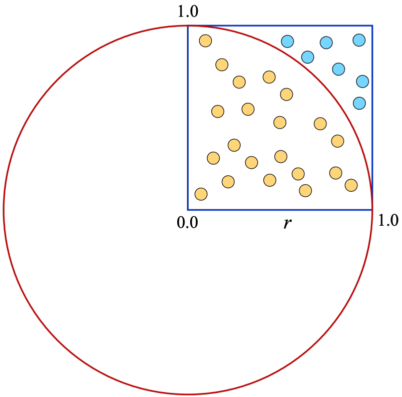
Stanislaw Ulam

John von Neuman



Nicholas Metropolis

Case Study: Calculating π by Monte Carlo Method



Consider a unit square with side length $r = 1.0$, and inscribe a circle with the radius r.

Generate a large number of random points within the square.

Count the number of points inside the quarter circle and unit square respectively.

The ratio of two numbers is an estimate of the ratio of the two areas, $\frac{\pi}{4}$.

Multiply the ratio by 4 to estimate π.

Markov Chain Monte Carlo (MCMC)

For high-dimensional probability distributions, MCMC provides a method by combining the Markov chain with Monte Carlo sampling.

Most used MCMC methods are *Gibbs sampling* and *Metropolis-Hastings algorithm*.

MCMC method adopts the form of transition probability: $T(x, x') = p(x'|x)$.

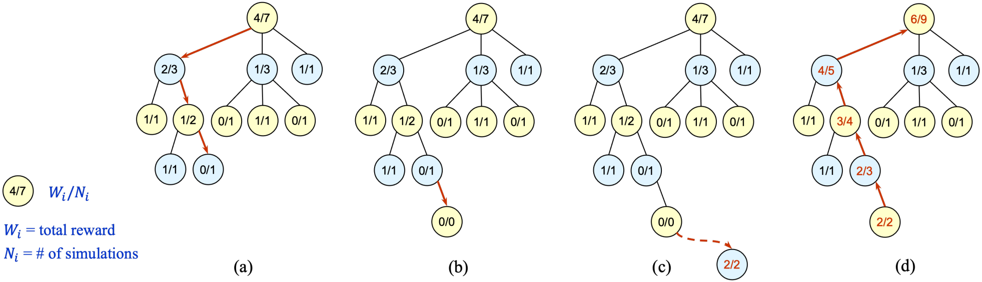
$$p(x') = \sum_x p(x, x')p(x) = \sum_x T(x, x')p(x).$$

$$T(x, x') = \sum_{k=1}^{K} \alpha_k B_k(x, x').$$

Where, the mixture coefficients α_k $(k = 1, \ldots, K)$, satisfies $\alpha_k \geq 0$, and $\sum_k \alpha_k = 1$; B_k is base transitions.

Monte Carlo Tree Search (MCTS)

MCTS combines random simulation with game tree search, used for some types of decision-making processes, especially for computer games such as Chess, Go.



W_i/N_i

W_i = total reward

N_i = # of simulations

Each loop in MCTS includes four steps: (a) selection, (b) expansion, (c) simulation, and (b) backpropagation.

Thank You