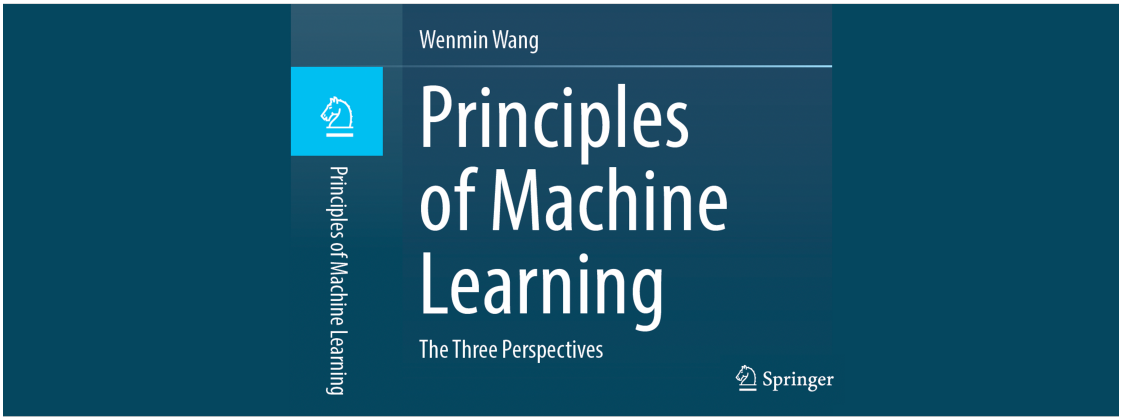


https://link.springer.com/book/10.1007/978-981-97-5333-8

Part II Frameworks

3 Probabilistic Framework

3 Probabilistic Framework

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 on 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 schools in probability theory

- Frequentist:
  probability is the limit of relative frequency of an event after many trials.
- Bayesian:
  probability is simply a measure of a degree of belief in an event.

Probability Learning Theory

Why does it need probability learning theory

- Machine learning must always deal with uncertain quantities.
- 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.

3 Probabilistic Framework

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, $X : \Omega \to \Sigma$, 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 $a \in \Sigma$.
- $\mathrm{Val}(X)$: the range of the random variable X, e.g., $a \in \mathrm{Val}(X)$.

Examples

Examples: 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: Random Variable

Throwing a dice, if

- $\text{Val}(X) = \{1, 2, 3, 4, 5, 6\}$ refers to the range of X is all number, then $X = 3$ means the number on dice is 3.
- $\text{Val}(X) = \{1, 0\}$ 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 Random Variable

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

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

Continuous Random Variable

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

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

Probability Relationship

Conditional Probability

$$P(X|Y) = \frac{P(X \cap Y)}{P(Y)}, \quad \text{where } P(Y) \neq 0.$$

Joint Probability

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

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

Expectations

Expectation of discrete random variable

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

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

Expectation of continuous random variable

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

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

Variance and Standard Deviation

Variance

Be the expectation of squared deviation of a random variable from its average.

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

Standard Deviation

Be 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.

3 Probabilistic Framework

About Computational Learning Theory

- Computational learning theory is a mathematical analysis theory about learnability, used for the design and analysis of ML 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.
- An important idea in computational learning theory is the introduction of concepts from computational theory into machine learning.

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

Learnability Theory

- Learnability theory is proposed by Leslie Valiant in a paper titled ”A Theory of the Learnable” in *Communications of the ACM*, 1984.
- It is the theoretical basis of computational learning theory, concerns:
  - 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 outside.
  - Deduction procedure:
    the algorithm for the concept to be learned is deduced.

PAC Learning

History of PAC Learning

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)

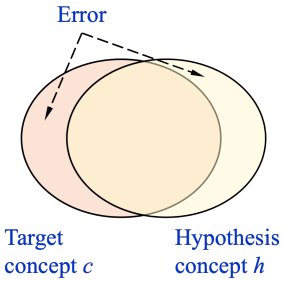
What is PAC Learning

- Be a framework for mathematical analysis of machine learning.
- With high probability (probably) and low generalization error (approximately correct).

Learning Model

Let $\mathcal{X}$ be the sample space, C be the concept class, $\boldsymbol{x}_i \in \mathcal{X}$, and $c \in C$ be a target class.

Without loss of generality, let c be a binary function, i.e., $c : \mathcal{X} \to \{0, 1\}$.



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 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.

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

"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 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 $c(\boldsymbol{x})$.
- $\text{size}(h) \leq (m \cdot \text{size}(c))^{\alpha} n^{\beta}$,

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

3 Probabilistic Framework

Bayes Theorem

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

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

where $P(X|Y)$ is likelihood probability, $P(Y)$ is prior probability, $P(X)$ is evidence, and $P(Y|X)$ is posterior probability.

Bayes theorem can be written as the following equation:

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

I.e., if Prior, Evidence, and Likelihood are known, its Posterior can be obtained.

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:

$$P\left(\text{Cause}|\text{Effect}\right) = \frac{P\left(\text{Effect}|\text{Cause}\right)P\left(\text{Cause}\right)}{P\left(\text{Effect}\right)},$$

where $P\left(\text{Effect}|\text{Cause}\right)$ is the likelihood of causal relationship, $P\left(\text{Cause}\right)$ is the prior of cause, and $P\left(\text{Effect}\right)$ is the evidence of effect.

If these aforementioned conditions are known, the posterior $P\left(\text{Cause}|\text{Effect}\right)$ can be inferred.

This is the theoretical basis of Bayesian learning.

3 Probabilistic Framework

Stochastic Process and Markov Property

Stochastic Process

A stochastic process (SP) is 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, each t is considered a point in time, and $S(t)$ is known as the state of stochastic process at time t.

Markov Property

A SP is said to have Markov property, if its next state $S(t+1)$ only depends on current state $S(t)$ but is irrelevant to past states $\{S(0), \ldots, S(t-1)\}$, i.e.

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

Markov Process

A Markov process (MP) is a SP with Markov property, represented as a 2-tuple, $\mathrm{MP} = \langle S, P \rangle$, where S is a state set, and P is the state transition probability.

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

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

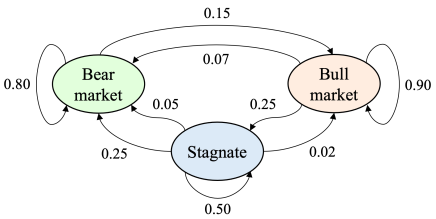
where $\mathbb{P}[\cdot]$ denotes the state transition matrix, that is:

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

Markov Chain

A Markov chain is a MP 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”. $\forall t$, $S(t) = s_t$ forms a countable set of values:

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



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

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

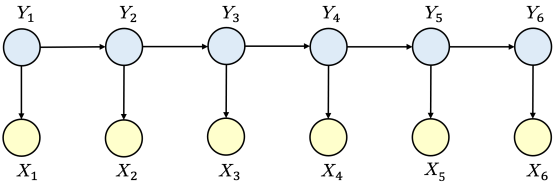
Hidden Markov Model

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

Let $\{Y_i\}_{i=1}^n$ and $\{X_i\}_{i=1}^n$ be discrete-time SPs, 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.

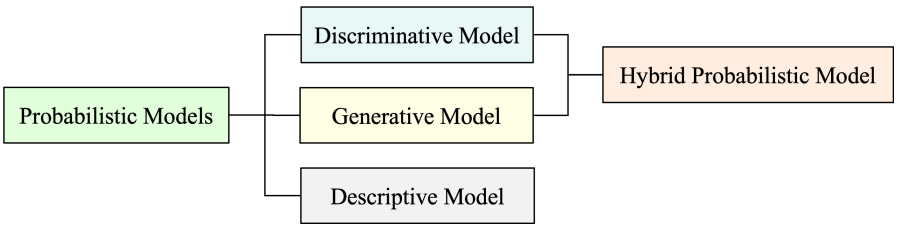


3 Probabilistic Framework

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 are used to the task of classification or regression.

Let training data be $S = \{(\boldsymbol{x}_i,\ y_i) \mid i = 1, \ldots, n\}$, where $\boldsymbol{x}_i \in X$, and $y_i \in Y$. Discriminative models are to model the decision boundary using $P(Y|X)$.

Example: Logistic regression

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

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

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

Another typical algorithm is conditional random field (CRF).

Generative Models

Generative models assume that their results are produced by latent variables through deterministic transformations.

Let $\boldsymbol{x} \in X$ be input. It is manifested as joint probability distribution $P(X, Y)$:

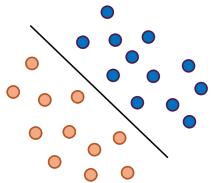
$$P(Y|X) = \frac{P(X, Y)}{P(X)} = \frac{P(X|Y)P(Y)}{P(X)}.$$

Therefore, we have:

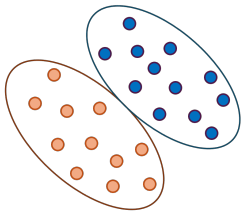
$$P(Y = 1 \mid X = \boldsymbol{x}) = \frac{P(X = \boldsymbol{x}, Y = 1)}{P(X = \boldsymbol{x})} = \frac{P(X = \boldsymbol{x} \mid Y = 1)P(Y = 1)}{P(X = \boldsymbol{x})},$$
$$P(Y = 0 \mid X = \boldsymbol{x}) = \frac{P(X = \boldsymbol{x}, Y = 0)}{P(X = \boldsymbol{x})} = \frac{P(X = \boldsymbol{x} \mid Y = 0)P(Y = 0)}{P(X = \boldsymbol{x})}.$$

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



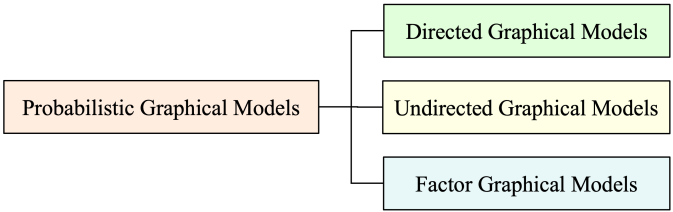
Generative model

3 Probabilistic Framework

Probabilistic Graphical Models

Probabilistic graphical models (PGMs) are 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.

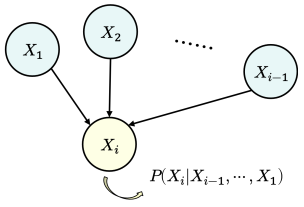


Directed Graphical Models

A directed graphical model (DGM) is a 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, $P_{\mathrm{DG}} = \{P_{X_i} | X_i \in X\}$.

DGMs are suitable for modeling problems of unidirectional dependencies.

Example: Bayesian networks



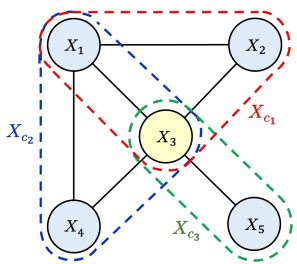
A Bayesian network (BN) is an ordered 2-tuple, $\mathrm{BN} = \langle \mathrm{DAG}, P_{\mathrm{BN}} \rangle$, where: the DAG is a directed acyclic graph, each probability $P_{X_i} \in P_{\mathrm{BN}}$ is a conditional probability of the following form:

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

Undirected Graphical Models

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

Example: Markov networks



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

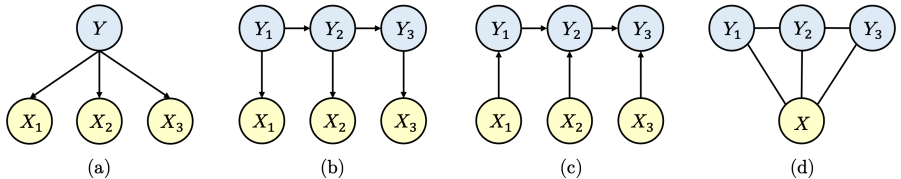
$$P_{\text{MN}}(X_1, \ldots, X_n) = \frac{1}{Z_{\text{MN}}} \prod_{k \in K} \phi_{c_k}(X_{c_k}).$$

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

Typical DGM and UGM

Typical DGMs are Bayesian network, naive Bayes classifier, hidden Markov model (HMM), and maximum entropy Markov model (MEMM).

A typical UGM is Markov network, also known as the Markov random field (MRF), and the conditional random field (CRF) is its special case.



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

3 Probabilistic Framework

Monte Carlo Methods

Monte Carlo (MC) methods are also called MC approximations or 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.

Include a broad class of algorithms, such as:

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.

MC methods have been played important roles in AI 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.

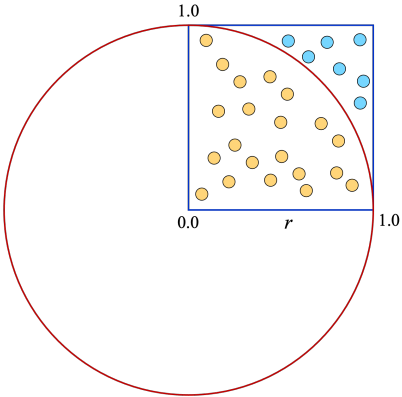
Metropolis suggested using the name Monte Carlo, which refers to the Monte Carlo Casino in Monaco where Ulam's uncle would go to gamble.

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*.

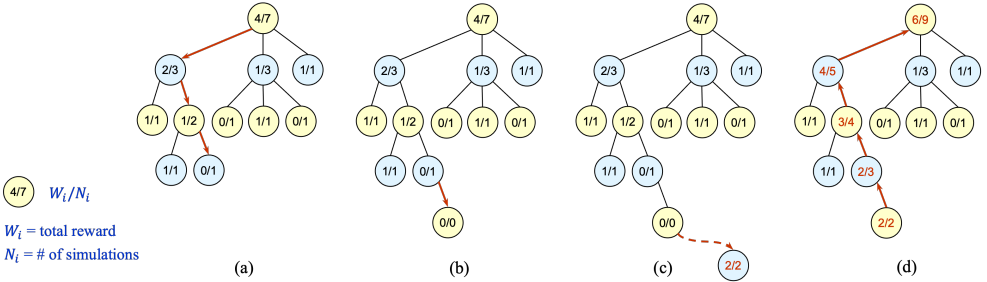
MCMC method adopts the form of transition probability: $T(x, x') = p(x'|x)$.

$$p(x') = \sum_x p(x, x')p(x) = \sum_x T(x, x')p(x), \quad \text{and} \quad T(x, x') = \sum_{k=1}^{K} \alpha_k B_k(x, x'),$$

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.



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

Thank You