

1 Probability

Each possible outcome of an experiment is called a *sample point*. The set of all sample points is called *sample space*, denoted by Ω . For example, in the experiment of 100 coin tosses, the sample space is $\{0, 1\}^{100}$. An *event* E is a subset of Ω . For example in the above experiment, the set of coins tossed with even number of heads is an event. Probability function P is a function from Ω to $[0, 1]$ such that $\sum_{x \in \Omega} P(x) = 1$. Probability of an event E is $\sum_{x \in E} P(x)$. Two events A and B are independent $Pr(A \cap B) = Pr(A) \times Pr(B)$.

A *random variable* X is a function from sample space to the real numbers, $X : \Omega \rightarrow \mathbb{R}$. Given a random variable X and a value α , $Pr[X = \alpha] = Pr(\{v \mid X(v) = \alpha\})$. The *expectation* of a random variable is defined as

$$E(X) = \sum_{\alpha} Pr(X = \alpha) \times \alpha.$$

$E(X)$ is *average* value of X .

Consider the experiment with n fair coin tosses. We can define a random variable X to be the number of heads. It can be shown that $E(X) = n/2$. An important property of expectation is that it is linear, i.e., If X and Y are two random variable and a is a real number, then

$$E(aX + Y) = aE(X) + E(Y).$$

This property can be used to compute the expectation of some random variables easily. Let X be the number of heads in n fair coin tosses. We will now compute $E(X)$. For $1 \leq i \leq n$, define random variable X_i as follows: $X_i = 1$, if the i th coin toss is a head, else X_i is zero.

$$E(X_i) = Pr(X_i = 1) \times 1 + Pr(X_i = 0) \times 0 = 1/2.$$

It is clear that $X = X_1 + X_2 + \dots + X_n$.

$$\begin{aligned} E(X) &= E(X_1) + E(X_2) + \dots + E(X_n) \\ &= 1/2 + 1/2 + \dots + 1/2 \\ &= n/2. \end{aligned}$$

Suppose we have two random variables X and Y which take the following values with equal probability.

$$\begin{aligned} X &: 47, 48, 49, 50, 51, 52, 53 \\ Y &: 20, 30, 40, 50, 60, 70, 80 \end{aligned}$$

The expectation of both X and Y is 50. The values taken by Y are more spread out than the values taken by X . We capture this fact by *variance*. The *variance* of a random variable X is defined as follows.

$$Var(X) = E([X - E(X)]^2).$$

Thus variance is (square of) the average distance of X from its Expectation.

Given a random variable X , we are often interested in computing probabilities such as $Pr(X > v)$, $Pr(|X - E(X)| > a)$. The following three inequalities help us to estimate this.

Markov's Inequality: Let X be a nonnegative random variable.

$$\Pr(X > v) \leq E(X)/v.$$

In other words,

$$\Pr(X > \alpha E(X)) \leq 1/\alpha.$$

Chebyshev's Inequality: Let X be random variable.

$$\Pr(|X - E(X)| \geq \delta) \leq Var(X)/\delta^2.$$

Chernoff's Bound: Let X_1, X_2, \dots, X_m be independent random variables that take values between 0 and 1. Let $E(X_1) = E(X_2) = \dots = E(X_m) = p$. Let $X = X_1 + X_2 + \dots + X_m$.

$$\Pr[|X/m - p| \geq \delta] \leq 2e^{-2\delta^2 m}.$$

2 Finite Fields

Let F be a set and $+$ and $*$ be functions from $F \times F$ to F . We call $+$ as addition and $*$ as multiplication. We say that $(F, +, *)$ is a field, if the following conditions hold.

- Both $+$ and $*$ are associative
- Both $+$ and $*$ are commutative
- $*$ distributes over $+$.
- There is an element $0 \in F$ such that $x + 0 = x$ for every $x \in F$. Such an element is called *additive identity*.
- There is an element $1 \in F$ such that $x * 1 = x$ for every $x \in F$. Such an element is called *multiplicative identity*.
- For every element in $x \in F$ there exists an element $y \in F$ such that $x + y = 0$. The element y is called the *additive inverse* of x and is often denoted as $-x$.
- For every non-zero element in $x \in F$ there exists an element $y \in F$ such that $x * y = 1$. The element y is called the *multiplicative inverse* of x and is often denoted as x^{-1} .

A field where F is a finite set is called *finite field*.

Consider $(\mathbb{Z}_n, +, *)$ where $\mathbb{Z}_n = \{0, 1, \dots, n-1\}$ and $+$ is defined as addition modulo n and $*$ is defined as multiplication modulo n . It is well known that \mathbb{Z}_n is a field if and only if n is a prime.

Let \mathbb{Z}_p be a field. For any $n > 1$ we can construct an *extension field* $(GF(p^n), +, *)$ as follows:

Fix an irreducible polynomial $r(x)$ of degree n over $GF(p)$. Now, an element of $GF(p^n)$ is a polynomial of degree $n-1$ over $GF(p)$. Operation $+$ is defined as addition of polynomials modulo $r(x)$, and the operation $*$ is defined as multiplication of polynomials modulo $r(x)$.

Thus for every prime p and every number $n > 0$, there exists a unique field (unique up to isomorphisms) $GF(p^n)$. More over, every finite field must be isomorphic to $GF(p^n)$ for some prime p and a natural number n .

3 Probabilistic Algorithms

A probabilistic algorithm is an algorithm that can toss coins during its computation. Note that the outcome of an probabilistic algorithm need not be unique. If we run the algorithm twice, we may get two different outputs, the outputs may depend on the result of coin tosses.

We say a probabilistic algorithm A computes a function f , if the A outputs the correct value of f with *high* probability. We have to define an appropriate notion of “high”. We consider the following definition.

Let $f : \Sigma^* \rightarrow \Sigma^*$. A probabilistic algorithm A computes f , if

$$\forall x, \Pr[A(x) = f(x)] \geq 2/3.$$

Above definition appears weak. The *error probability* of the above algorithm is very high—the algorithm can go wrong 1/3rd of time. However, as the next result shows, we can design a new algorithm B whose error probability is much small, and the running time of B is a little more than the running time of A .

Let A be a probabilistic algorithm that computes f in time $t(n)$. Then there exists a probabilistic algorithm B whose running time is $O(nt(n))$ and

$$\forall x, \Pr[A(x) = f(x)] \geq 1 - 1/2^n, |x| = n.$$

The algorithm B works as follows.

1. input: $x, |x| = n$. Let $m = 18n$.
2. for $i = 1$ to m compute
3. $a_i = A(x)$.
4. Output the majority value of a_1, \dots, a_m . If there is no majority output a_1 .

We claim that B computes f with very high probability. We define few random variables. For $1 \leq i \leq m$ define X_i as follows: $X_i = 1$, if $a_i = f(x)$, $X_i = 0$, if $a_i \neq f(x)$. Note that $E(X_i) \geq 2/3$. Let $X = X_1 + X_2 + \dots + X_m$. Note that B outputs a wrong value, if majority of a_i 's are wrong. Thus B outputs a wrong value if majority of X_i 's are zero. We can use Chernoff's bound to show that the probability of such event is very small.

$$\begin{aligned} \Pr[B \text{ outputs a wrong value}] &= \Pr[\text{majority of } a_i \text{'s are wrong}] \\ &= \Pr[\text{majority of } X_i \text{'s are zeros}] \\ &= \Pr[X \leq m/2] \\ &= \Pr[X/m \leq 1/2] \\ &\leq \Pr[|X/m - 2/3| \geq 1/6] \\ &\leq 2e^{-2m/36} \\ &\leq 1/2^n. \end{aligned}$$

Thus

$$\Pr[B(x) = f(x)] \geq 1 - 1/2^{|x|}.$$

A language L is in BPP if there is a probabilistic polynomial-time algorithm M such that for every x

$$x \in L \Rightarrow \Pr[M \text{ accepts } x] \geq 2/3,$$

$$x \notin L \Rightarrow \Pr[M \text{ accepts } x] \leq 1/3.$$

Often it helps to think a probabilistic machine M as follows: On any input, M first tosses all coins and obtains random bits. After this step the computation is deterministic. More formally, an alternate definition BPP is the following: A language L is in BPP if there is a deterministic polynomial-time machine M , and a polynomial p such that

$$x \in L \implies \Pr_{r \in \Sigma^{p(n)}} [M(x, r) \text{ accepts}] \geq 2/3,$$

and

$$x \notin L \implies \Pr_{r \in \Sigma^{p(n)}} [M(x, r) \text{ accepts}] \leq 1/3.$$

A language L is in RP if there is a deterministic polynomial-time machine M , and a polynomial p such that

$$x \in L \implies \Pr_{r \in \Sigma^{p(n)}} [M(x, r) \text{ accepts}] \geq 2/3,$$

and

$$x \notin L \implies \Pr_{r \in \Sigma^{p(n)}} [M(x, r) \text{ accepts}] = 0.$$

Again, the success probability can be amplified to $1 - 1/2^{q(n)}$ for any polynomial q , with a polynomial blowup in time.

4 Circuits

A circuit is an acyclic directed graph with n input nodes, m output nodes, and each internal node is labeled “AND”, “OR”, or “NOT”. Nodes labeled “AND” and “OR” have in degree at least two and nodes labeled “NOT” have indegree 1. Given circuit C , the size of the circuit is the size of the graph that represents the circuit.

Let f be a function from $\Sigma^n \rightarrow \Sigma^m$. Let C be a circuit with n input gates and m output gates. We say that C computes f if for every x , C on input x outputs $f(x)$.

Let f be a function from $\Sigma^* \rightarrow \Sigma^*$. Let $C = [C_1, C_2, \dots]$ be a family of circuits with C_n having n input gates. We say that the C computes f if for every x , the circuit $C_{|x|}$ on input x outputs $f(x)$.

Let L be a language and $C = [C_1, C_2, \dots]$ be a family of circuits. We say that C accepts L if C compute the characteristic function of L .

A language L is in P/poly, if there is a polynomial p and a family of circuits $C = [C_1, C_2, \dots]$ such that C accepts L and the size of C_n is bounded by $p(n)$ for every n .