Lecture 1

ECE 4950 – Intro to
Quantum Information Science

Statements on quantum computing, program @ Cornell, other courses

---

Requirements for course:
regular homeworks due every 2 weeks, midterm, & final exam.

---

office hours – set up time

QKD – interest

QEC – "

q. chemistry – "

Review basics of probability theory needed for q. computing.

- Concept of a discrete, finite random variable X, takes values $x \in \mathcal{X}$ w/ probability $P_X(x)$, such that $P_X(x) \geq 0 \quad \forall x \in \mathcal{X}$

  & $\sum_{x \in \mathcal{X}} P_X(x) = 1$.

- Expected value or mean of R.V. X is

  $\mu \equiv \mathbb{E}\{X\} \equiv \mathbb{E}_{P_X}[X] \equiv \sum_{x \in \mathcal{X}} P_X(x) \cdot x$

  Give interpretation later of why this is expected value.

- Variance is

  $Var[X] = \sum_{x \in \mathcal{X}} P_X(x) (x - \mu)^2$

$$= \mathbb{E}[(X-\mu)^2] = \mathbb{E}[(X-\mathbb{E}[X])^2]$$

can work out that

$$\mathrm{Var}[X] = \mathbb{E}[X^2] - \mu^2$$

Standard deviation:

$$\sigma \equiv \Delta(X) \equiv \sqrt{\mathrm{Var}[X]}$$

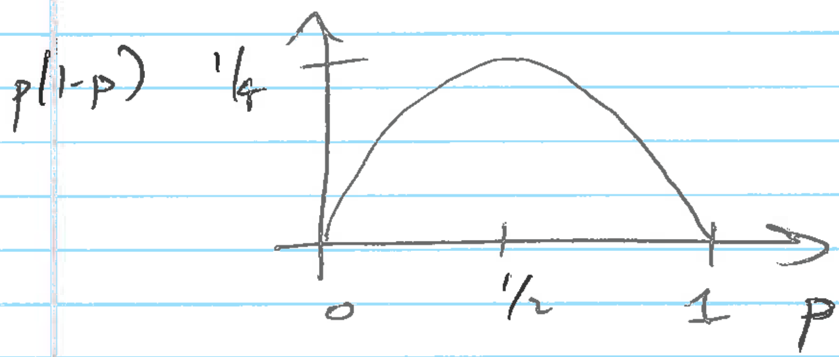
measure of the spread about the mean.

common in complexity theory & algorithms to consider a Bernoulli R.V., which takes value 1 w/ prob. $p \in [0,1]$ & value 0 w/ prob. $1-p$.

1 indicates success, while 0 indicates failure.

What is its mean? p

What is its variance? $p(1-p)$



(most variance when $p = 1/2$)

Why are Bernoulli RV's important for computation?

Many algorithms output ~~[illegible]~~ "yes" or "no" as an answer.

In randomized algorithms or quantum algorithms, the answer is ~~probabilistic~~ correct only w/ some probability

Examples of tasks w/ binary outputs:

1) binary classification in machine learning (decide cat or dog)

2) Property testing:
decide whether some object has a property or not
decide whether a number is prime
decide if a q. state is entangled

3) decide whether a solution exists to a problem or not

When an algorithm is probabilistic, how do we gain confidence that an answer is correct?

Repeat the algorithm!!

When we repeat the algorithm, we make two assumptions:

1) that the Bernoulli RVs corresponding to each trial are <u>independent</u>

2) that they are <u>identically distributed</u>

Together these are known as the <u>i.i.d.</u> assumption

As an example, suppose we have an algorithm A that decides whether a positive integer N is prime & it is correct w/ probability ~~...~~ $\geq 2/3$.

That is, if N is prime,

then A outputs prime w/ prob. $p_1 \geq 2/3$

& if N is not prime,

then A outputs not prime

w/ prob. $p_0 \geq 2/3$.

- Basic idea is that we can invoke the Chernoff bound to ~~p~~ prove that failure probability decreases exponentially fast w/ # of repetitions

- Recall Chernoff bound

Given are independent Bernoulli RVs

$X_1, \ldots, X_T$ w/ sum $S = \sum_{t=1}^{T} X_t$

Then, for $\delta \in (0,1)$

$$\Pr\left[S \le (1-\delta)\mathbb{E}[S] \right] \le e^{-\delta^2 \mathbb{E}[S]/2}$$

How to apply this?

Let A' be modified algorithm that repeats A T times.

~~[illegible]~~

This leads to Bernoulli RVs $Y_1, \dots, Y_T$. Algorithm A' decides prime if $\sum_{t=1}^{T} Y_t \ge T/2$ & not prime otherwise.

Then failure probability is

$$\Pr[A'(N) = \text{not prime} \mid N \text{ is prime}]$$

$$= \Pr\left[\sum_{t=1}^{T} Y_t < T/2 \mid N \text{ is prime}\right] \quad (*)$$

Consider that

$$\mathbb{E}\left[\sum_{t=1}^{T} Y_t\right] = T \cdot p_1 \geq T \cdot \frac{2}{3}$$

Set δ to be such that

$$\frac{T}{2} = (1-\delta) T \cdot \frac{2}{3} \Rightarrow \delta = 1/4$$

$\Rightarrow$ $(*)$ =

$$\Pr\left\{\sum_{t=1}^{T} Y_t < (1-\delta) T \cdot 2/3 \mid N \text{ is prime}\right]$$

$$\leq \Pr\left\{\sum_{t=1}^{T} Y_t < (1-\delta) T \cdot p_1 \mid N \text{ is prime}\right\}$$

$$\leq \exp(-\delta^2 \cdot T \cdot p_1 / 2)$$

$$\leq \exp\left(-\delta^2 T \cdot \frac{2}{3 \cdot 2}\right)$$

$$= \exp(-T/48)$$

QED

We can analyze the failure probability

$$\Pr[A'(N) = \text{prime} \mid N \text{ is not prime}]$$

in a similar way by using the other Chernoff bound

$$\Pr[S \geq (1+\delta)\mathbb{E}[S]] \leq e^{-\delta^2 \mathbb{E}[S]/(2+\delta)}$$

in a similar way (exercise)

Another situation that arises w/ a probabilistic ~~or~~ or q. algorithm is when the algorithm is successful w/ probability $p \in [0,1]$ & there is a way to verify the answer.

- For example, if we are trying to factor an integer into a product of primes, we can easily check whether the answer given is correct.
- Another example is in a search task, ~~where~~ in which we can efficiently check whether a proposed answer is correct.

In this case, we are interested in the number of trials / runs of the algorithm that are necessary until a success occurs.

T2

Such a situation is modeled by a geometric random variable

X is geometric if it is equal to the # of independent Bernoulli trials needed to observe one success

$$\Pr\{X=k\} = (1-p)^{k-1} p$$

(probability to have k-1 failures followed by one success)

$$\mathbb{E}\{X\} = \frac{1}{p} \quad b/c$$

$$\mathbb{E}\{X\} = \sum_{k=1}^{\infty} (1-p)^{k-1} p \cdot k$$

$$= p \sum_{k=1}^{\infty} (1-p)^{k-1} \cdot k$$

$$= p \sum_{k=0}^{\infty} (1-p)^{k-1} \cdot k$$

$$= p \cdot \left(-\frac{d}{dp} \sum_{k=0}^{\infty} (1-p)^k \right)$$

$$= p \cdot \left(-\frac{d}{dp} \frac{1}{p} \right)$$

$$= \frac{1}{p}$$

this is a critical parameter for "repeat until success" algorithms

---

We are also interested in using randomized or quantum algorithms for estimation tasks.

Given a random variable X (w. prob. dist. $P_X(x)$), we can sample from it & compute the sample mean

More formally, let $X_1, \dots, X_n$ be i.i.d. samples of X.

Sample mean is

$$\overline{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i$$

Hoeffding bound guarantees that

$$\Pr\left\{ \left| \overline{X}_n - \mathbb{E}[X] \right| \le \varepsilon \right\} \ge 1 - \delta$$

where $\varepsilon, \delta \in (0,1)$,
as long as

$$n \ge \frac{M^2}{2\varepsilon^2} \ln\left(\frac{2}{\delta}\right)$$

gives a guarantee on sample complexity of algorithm

where $M = b - a$ of each X takes values on finite interval $[a, b]$.