Lecture 1

Jan 14, 2025

probability review

property testing

testing sortedness (0,1 arrays)

Question: uniformity testing

We say algorithm A is an (ε, δ)-tester for uniformity if the following holds with a probability

discrete probability space (Ω, P)

$\Omega = \{\omega_1, \omega_2, \ldots\}$

↓ Sample space, (finite or countable)

↘ elementary outcomes.

event: any subset of Ω

P: events $\longrightarrow [0,1]$

usually defined just by determining $p(\{\omega_i\})$ or $p(\omega_i)$ ↙

s.t:

* $\forall$ event $E \subseteq \Omega$

$$p(E) = \sum_{\omega \in E} p(\omega)$$

* $p(\Omega) = 1$

random variable → takes value in Ω

based on a probability distribution.

$X \sim P$

* Expected Value

$$E[X] = \sum_{x \in \Omega} p(x) \cdot x$$

$$\int_{x \in \Omega} p(x) \cdot x \, dx$$

* Variance $= E[(X - E[X])^2]$

$= E[X^2] - E[X]^2$

* linearity of expectation:

$$E[X_1 + X_2] = E[X_1] + E[X_2]$$

$$E[\alpha X] = \alpha E[X]$$

$\Rightarrow \text{Var}[\alpha X] = \alpha^2 \text{Var}[X]$

$\text{Var}[X + Y] \neq \text{Var}[X] + \text{Var}[Y]$

For two events $A, B \subseteq \Omega$:

* joint probability $\Pr[A \cap B]$

probability that both A and B happen.

* Conditional probability $\Pr[A \mid B]$

probability that A happens conditioned on that B happens

$$\Pr[A \mid B] = \frac{P(A \cap B)}{P(B)}$$

Bayes' Theorem $P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$

* Independence

A and B are independent events

$\Leftrightarrow \quad P(A|B) = P(A)$

$\Leftrightarrow \quad Pr(A \cap B) = P(A).P(B)$

Two random variable $X \in \mathcal{R}$ and $X' \in \mathcal{R}'$ are independent iff

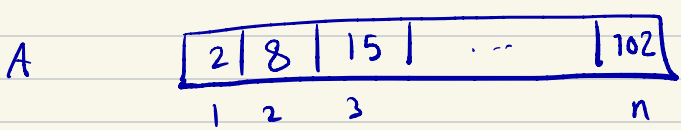
$\forall \; x \in \mathcal{R}, \; x' \in \mathcal{R}'$ two events $X = x$ and $X' = x'$ are independent.

Randomness in computation

One example of employing randomness to obtain faster algorithms is in designing sub-linear algorithm.

our example today:

Testing sortedness of an array



Def. A is sorted $\iff A[1] \leq A[2] \leq \cdots \leq A[n]$

Any deterministic algorithm needs to query $\Omega(n)$ cells in A to test sortedness.

Can we test sortedness with $o(n)$ queries?

Even a randomized algorithm would require $\Omega(n)$ queries to A.

what is the difficulty? perfectionism

Randomness

- cannot find needle in a haystack

  ↳ there must be substantial evidence of unsortedness.

- unlikely events may occur against all odds.

  ↳ It is ok to fail with small prob.

Property testing

property P = a set of objects

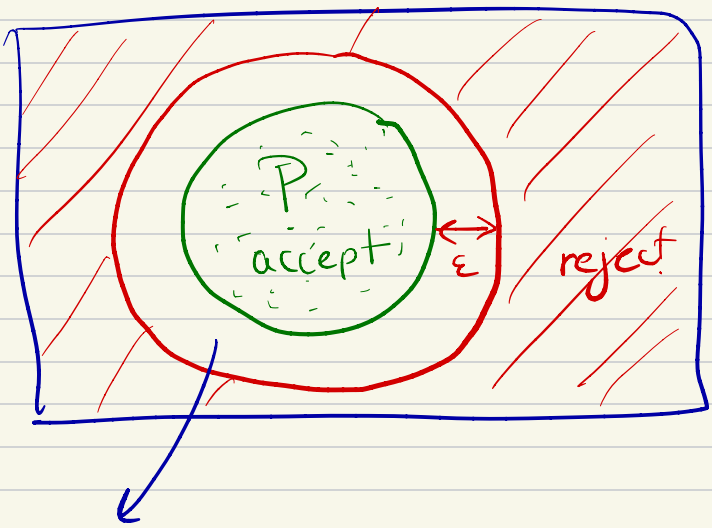
We say an object has property P iff the object is in P.

Suppose we have an underlying object O

Def We say an algorithm A is an (ε, δ)-tester for property P if the following holds with prob. at least $1-\delta$:

if $O \in P$, A outputs accept

if O is ε-far from P, A outputs reject



Both answers are correct.

* ε → make the difference detectable

* δ → leaves room for error whe we get unlucky.

* "ε-far" will be defined in the context of the problem.

For sortedness problem, suppose we have two arrays of length n: A and A'

- We say A and A' are ε-far iff one can change A in $> \varepsilon \cdot n$ entries to obtain A'.

- We say A is ε-far from being sorted if A is ε-far from

all sorted arrays.

Easy case: 0-1 array

proposed Algorithm t:

- Draw m samples uniformly at random from $[n]$
  ↳ $\{1, 2, \ldots, n\}$
- sort the samples: $i_1 \leq i_2 \leq \ldots \leq i_m$

If $A[i_1] \leq A[i_2] \leq \ldots \leq A[i_m]$

output accept

Else

output reject.

We aim to find m such that

$\mathcal{A}$ is an (ε, δ) tester.

$$\Pr[\text{wrong answer}] < \delta$$

step 1) if A is sorted, for every $i<j$ we have $A[i] < A[j]$, and there is no violation. Thus it is impossible for $\mathcal{A}$ to find one.

$$\Rightarrow \Pr[\text{outputting reject} \mid \text{sorted } A] = 0$$

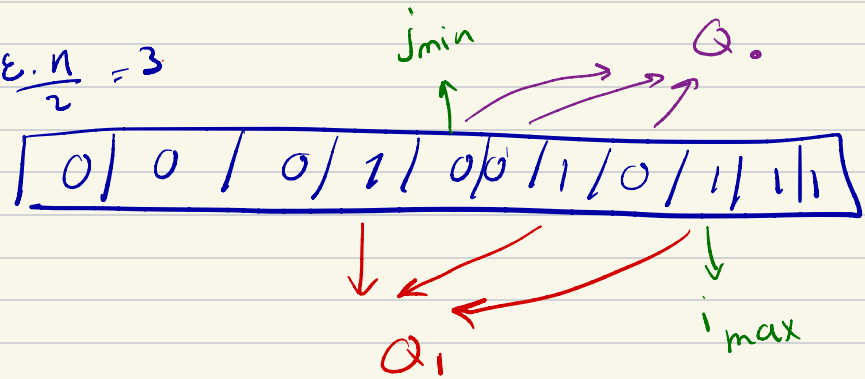
step 2) Next, we show that

$$\Pr\left[\text{Outputting accept} \mid A \text{ is } \varepsilon\text{-far from being sorted}\right] < \delta$$

Q_1 = { set of $\frac{\varepsilon \cdot n}{2}$ indices of the left most 1's }

Q_0 = { set of $\frac{\varepsilon \cdot n}{2}$ indices of the right most 0's }

Example: if $\frac{\varepsilon \cdot n}{2} = 3$



Lemma 1

if A is ε-far from being sorted $\Rightarrow$

all indices in Q_1 are smaller than all indices in Q_0

(

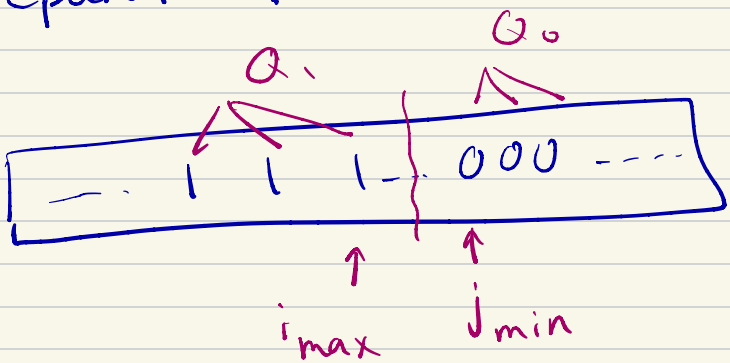
that is $i_{max} < j_{min}$

where j_{min} = min index in Q_0

& i_{max} = max index in Q_1

Pictorially, the lemma says:

if A is ε-far there is a separation:



Proof of Lemma 1:

By contradiction assume:

$$j_{min} < i_{max}$$

we show A cannot be ε-far from being sorted by constructing a sorted A' that is ε close to A.

$$A'[i] = \begin{cases} 0 & i \leq i_{max} \\ 1 & i > i_{max} \end{cases}$$

Obviously A' is sorted.

distance between A & A'

We change two groups in A to get $\acute{A}$

1) $A[i] = 1$ if $i \leq i_{max}$

2) $A[i] = 0$ if $i > i_{max}$

1) There are exactly $\frac{\varepsilon \cdot n}{2}$ 1's in A that appear before i_{max}

This holds by definition of Q_1.

2) There are less than $\frac{\varepsilon \cdot n}{2}$ 0's that appear after i_{max}

Since $j_{min} < i_{max}$

$\Rightarrow$ with $< \varepsilon \cdot n$ changes to A we get to A', a sorted array

$\Rightarrow$ A is not ε-far ~~X~~

Hence, $j_{min} > i_{max}$

(they cannot be equal either) $\square$

Suppose A is ϵ-far

$\Rightarrow$ indices in Q_1 $<$ indices in Q_0.

Lemma 1

if algorithm A samples

$i \in Q_1$ and $j \in Q_0$

$\Rightarrow$ A output reject

why? $A[i] = 1$, $A[j] = 0$

but $i < j$

Define two events

E_1 = at least one sampled index $\in Q_1$

E_0 = " " " " $\in Q_0$

We just showed

$$E_1 \wedge E_2 \Rightarrow \text{outputting reject}$$

Now, let's go back to bounding the probability of wrong answer:

$$\Pr[\text{outputting accept} \mid A \text{ is } \varepsilon\text{-far}]$$

$$\leq \Pr[\overline{E_1 \wedge E_2} \mid A \text{ is } \varepsilon\text{-far}]$$

$$\leq \Pr[\bar{E}_1 \vee \bar{E}_2 \mid A \text{ is } \varepsilon\text{-far}]$$

$$\leq \Pr[\bar{E}_1] + \Pr[\bar{E}_2] \quad \text{union bound}$$

$$\leq 2\Pr[\bar{E}_1] \quad \text{symmetry}$$

We pick m random index in [n]

$$\Pr[\text{one sample} \in Q_1] = \frac{|Q_1|}{n} = \frac{\varepsilon}{2}$$

$$\Pr[\text{one sample} \notin Q_1] = \frac{|Q_1|}{n} = 1 - \frac{\varepsilon}{2}$$

$$\Pr[E_1] = \Pr[\text{one sample among m samples} \notin Q_1]$$

$$= \left(1 - \varepsilon/2\right)^m$$

$$= e^{\varepsilon/2 \cdot m} \qquad \text{using } 1 - x < e^{-x}$$

$$\leq \frac{\delta}{2} \qquad \text{set } m = \frac{2}{\varepsilon} \log \frac{2}{\delta}$$

$\Rightarrow$ $\Pr[\text{outputting accept} \mid A \text{ being } \varepsilon\text{-far}] < \delta$

$\Rightarrow$ Hence, A with $m = \frac{2}{\varepsilon} \log 2/\delta$ is an (ε, δ)-tester for sortedness.