Lecture 16:

Hypothesis Testing
Some Problems: 

Given samples of $p$

- $p = q$ (e.g., $q = U_0$)
  - or \( \varepsilon \)-far from $q$

- $p$ \( \varepsilon \)-close to $q$
  - or \( \varepsilon \)-far from $q$

\[ \text{Complexity (in terms of } n = |Q|) \]

\[ \sqrt{n} \]

\[ \frac{n}{\log n} \]

\[ n^{\frac{2}{3}} \]

Given samples of $q$

- $p = q$
  - or \( \varepsilon \)-far from $q$

- $p$ \( \varepsilon \)-close to $q$
  - or \( \varepsilon \)-far from $q$

- $p$ monotone
  - or \( \varepsilon \)-far from monotone

- $p$ \( \varepsilon \)-close to monotone
  - or \( \varepsilon \)-far from monotone

\[ n^{\frac{2}{3}} \]

\[ \frac{n}{\log n} \]
Other problems considered:

estimate entropy, support size
independence?
represented well via K-histogram?
monotone hazard rate
• • •
A useful tool:

Given: (1) collection of distributions (via complete description) \( \mathcal{H} \)

(2) Samples of \( p \) such that \( \exists q \in \mathcal{H} \) for which \( \text{dist}(pq) \) is small

Goal: Output \( h \in \mathcal{H} \) s.t. \( \text{dist}(p,h) \) small

Question:

How many samples needed in terms of \(|\mathcal{H}| \times \) domain size?

Is this the same as testing closeness, uniformity? Is \( p \) guaranteed to be close to some \( q \)?

Do lower bounds apply? NO!
What we want:

Given $h_1, h_2$ explicit $p$ via samples

procedure that outputs $h_1$ that is closer to $p$

What if both are roughly same distance?

maybe either one is ok?

or maybe not...

More general Goal:

Given set of hypotheses $H$ + $p$ via samples

find $h \in H$ closest to $p$
Find best hypothesis via "tournament"?

“Winner” advances at each phase

\[ h_1, h_2, h_3, h_4, h_5, h_6, \ldots \]

Need stronger guarantee!

\[ \text{overall winner} \]

maybe \( p = h_i \)

\[ ||p - h_2||_1 = 3 \Rightarrow h_2 \ \text{“wins”} \]

Then \[ ||p - h_3||_1 = 2 \Rightarrow h_3 \ \text{“wins”} \]

Then \[ ||p - h_5||_1 = 3 \Rightarrow h_5 \ \text{“wins”} \]

\[ \ldots \]
won't use simple tournament ← instead compare every pair
will add notion of "tie"

Output hypothesis that wins or ties every match
(hopefully there is one, it is the right one)
A "subtool" for comparing two hypotheses:

Thm. Given (1) sample access to $p$

(2) $h_1, h_2$ hypothesis distributions (fully known to algorithm)

(3) accuracy parameter $\varepsilon'$, confidence parameter $\delta'$

then Algorithm "choose" takes $O(\log(1/\delta')/(\varepsilon')^2)$ samples and outputs

$h \in \{h_1, h_2\}$ satisfying:

if one of $h_1, h_2$ has $\|h_i - p\|_1 < \varepsilon'$

then with prob $\geq 1 - \delta'$, output $h_j$ has $\|h_j - p\|_1 < 12\varepsilon'$

i.e. if both $h_1, h_2$ far, no guarantees

if one $\varepsilon'$ close and one really far $j$ will output $\varepsilon'$-close hypothesis

if both $\varepsilon'$ close then output $12\varepsilon'$-close hypothesis

i.e. one is $\varepsilon'$-close

other is $\leq 10\varepsilon'$-close

getting kind of complicated just to specify 😐
Actually a bit stronger:

**Thm**

\[ p \] given via samples
\[ h_1, h_2 \] fully known + \( p \) is \( \varepsilon' \)-close to at least one of \( h_1, h_2 \)

\( \varepsilon' \) given

Algorithm "choose" takes \( O((\log \frac{1}{\delta})(\frac{1}{\varepsilon})^2) \) samples + outputs \( h \in \{h_1, h_2, h_3\} \) such that:

1. If \( h \) more than \( 12\varepsilon' \)-far from \( p \), unlikely to output \( h \) as winner

   \[ \text{very bad} \]

   \( \frac{2e^{-m(\varepsilon')^2}}{2e^{-m(\varepsilon')^2}} \)

   or tie

2. If \( h \) more than \( 10\varepsilon' \)-far from \( p \), unlikely to output \( h \) as winner

   \[ \text{not that bad} \]

   \[ \frac{2e^{-m(\varepsilon')^2}}{2e^{-m(\varepsilon')^2}} \]

Can use \( \varepsilon' = \frac{\varepsilon}{10} \)?
Proof of subtool:

idea: why $h_1$ is $\varepsilon$-close to $p$

if $h_2$ is $10\varepsilon$-close to $p$, then ok to output "tie" or either $h_1$ or $h_2$ as "winner"

else, if $h_2$ is not $10\varepsilon$-close to $p$ but is $12\varepsilon$-close, ok to "tie" or output $h_1$ as "winner"

else $h_2$ is $12\varepsilon$-far from $p$ + $11\varepsilon$-far from $h_1$

so samples from $p$ will fall in "difference" between $h_1$ and $h_2$,

it will output $h_1$

$h_1$ and $h_2$ are close

you can determine $h_1$ and $h_2$ close w/o samples from $p$

Since you know $h_1$ and $h_2$, you know

where to look for this difference:

does $p$ assign prob to $A$ more like $h_1$ or $h_2$?

(here you use samples)
Algorithm. Choose: $\text{Input } p, h_1, h_2$

First some definitions:

$$A = \sum \{ x | h_1(x) > h_2(x)^2 \}$$

$$a_1 = h_1(A) \quad \text{← red + blue areas}$$

$$a_2 = h_2(A) \quad \text{← blue area}$$

Note $\|h_1 - h_2\|_1 = 2(a_1 - a_2)$

1. If $a_1 - a_2 \leq \frac{1}{2} \varepsilon$ declare "tie" & return $h_1$

2. Draw $m = 2 \log \frac{4}{\delta} \varepsilon$ samples $s_1 \ldots s_m$ from $p$

3. $d = \frac{1}{m} \sum s_i \in A_2 \{ \varepsilon \}$

4. If $d > a_1 - \frac{3}{2} \varepsilon$ return $h_1$

   Else if $d < a_2 + \frac{3}{2} \varepsilon$ return $h_2$

   Else declare "tie" & return $h_1$
Why does it work?

- $h_1$ or $h_2$ is $\varepsilon'$-close to $A$ (given)
- If "tie" in step 1:
  
  $h_1 + h_2$ are $10\varepsilon'$-close (note $L_1$ dist = 2(a_1-a_2))
  
  $\Rightarrow$ both are $\leq 11\varepsilon'$-close to $A$
  
  So "tie" is ok

- Otherwise reach step 2: $\|h_1 - h_2\|_1 > 10\varepsilon'$ ($a_1 - a_2 > 5\varepsilon'$)

Algorithm

Choose:

$A = \frac{1}{2} \left\{ h_1(A) > h_2(A) \right\}$

$a_i = h_i(A)$

$a_2 = h_2(A)$

note $\|h_1 - h_2\|_1 = 2(a_1 - a_2)$

1. if $a_1 - a_2 \leq 5\varepsilon'$ declare "tie" & return $h$
   (no samples needed)

2. draw $m = 2 \log \frac{1}{\varepsilon'}$ samples $s_1...s_m$ from $p$

3. $x \leftarrow \frac{1}{m} \sum s_i e^{A s_i}$

4. if $x > a_1 - \frac{3}{2} \varepsilon'$ return $h_1$
   
   else if $x < a_2 + \frac{3}{2} \varepsilon'$ return $h_2$
   
   else declare "tie" & return $h_i$

- if $p = h_1$, $E[x] = a_1$
  - if $p = h_2$, $E[x] = a_2$

green area = red area = $a_1 - a_2$
L_1 dist = green + red
blue area = $a_2$
blue + red area = $a_1$
Algorithm: Choose:

\[ \begin{align*}
A &= \frac{2}{3} \times \left( h(x) > h_2(x) \right) \\
a_1 &= h_1(A) \\
a_2 &= h_2(A) \\
\text{note } ||h_1-h_2||_1 &= 2(a_1-a_2)
\end{align*} \]

1. if \( a_1 = a_2 \leq 5 \varepsilon' \) declare "tie" & return \( h \) (no samples needed)

2. draw \( m = 2 \log \frac{1}{\varepsilon'} \) samples \( s_1, \ldots, s_m \) from \( p \) [\( \varepsilon' \)]

3. \( \alpha \leftarrow 1 \cdot \frac{1}{m} \left\lvert \{ s \in S \mid s \in A \} \right\rvert \)

4. if \( \alpha > a_1 - \frac{3}{2} \varepsilon' \) return \( h_1 \)

else if \( \alpha < a_2 + \frac{3}{2} \varepsilon' \) return \( h_2 \)

else declare "tie" & return \( h \)

- if \( p = h_1 \), \( \mathbb{E}[\alpha] = a_1 \)
- if \( p = h_2 \), \( \mathbb{E}[\alpha] = a_2 \)

Why does it work?

- \( h_1 \) or \( h_2 \) is \( \varepsilon' \)-close to \( A \) (given)

- If "tie" in step 1, algorithm does right thing

- Otherwise reach step 2: \( ||h_1-h_2||_1 > 10 \varepsilon' \) (\( a_1-a_2 > 5 \varepsilon' \))

\[
\mathbb{E}[\alpha] = \Pr_{x \in p} \left[ x \in A \right] = p(A)
\]

Assume (Chernoff) that with high prob \( |\alpha - \mathbb{E}[\alpha]| \leq \frac{\varepsilon'}{2} \)

\( h_1 \) assigns \( a_1 \) weight to \( A \)

\( h_2 \) " " " " " \( A \)

if \( p \) is \( \varepsilon' \)-close to \( h_1 \), assigns \( \geq a_1 - \varepsilon' \) weight to \( A \)

\[ \alpha \geq a_1 - \frac{\varepsilon'}{2} = a_1 - \frac{3 \varepsilon'}{2} \]

" " " " " \( h_2 \), " " " " " \( \leq a_2 + \varepsilon' \) weight to \( A \)

\[ \alpha \leq a_2 + \frac{\varepsilon'}{2} \leq a_2 + \frac{3 \varepsilon'}{2} \]

return \( h_2 \) etc
**The cover method** - a method for learning distributions

```
def C is an "ε-cover" of D if
∀ p ∈ D ⇒ ∃ q ∈ C s.t. ||p - q||_1 ≤ ε
smaller set of distributions
```

Why useful?

Hopeful $C$ is much smaller than $D$, allows us to approximate $D$

Note $C$ not unique

**Thm** $\exists$ algorithm, given $p \in D$, which takes

$O\left(\frac{1}{\epsilon^2 \log |C|}\right)$ samples of $p$ + outputs $h \in C$

s.t. $||h - p||_1 \leq 6\epsilon$ with prob $\geq \frac{9}{10}$
**Thm** 3 algorithm, given \( \phi \in \mathcal{D} \), which takes 

\[ O\left( \frac{1}{\varepsilon^2 \log |C|} \right) \] 

samples of \( p \) + outputs \( h \in \mathbb{C}^n \) 

s.t. \( \|h - p\|_1 \leq 6\varepsilon \) with prob \( \geq \frac{9}{10} \)

**Pf.**

Since \( p \in \mathcal{D} \), \( \exists q \in \mathbb{C}^n \) s.t. \( \|p - q\|_1 \leq \varepsilon \) 

(could be more than one)

run “Choose” on \( p \) with every pair \( q_1, q_2 \in \mathbb{C}^n \) 

if \( q_{\text{best}} \) doesn’t win all of its “matches” then it ties 

with others that are not so bad

if \( q' \) is \( \geq 6\varepsilon \)-far from \( p \), then \( \geq (6\varepsilon - \varepsilon) = 5\varepsilon \)-far from \( q_{\text{best}} \) 

\( \Rightarrow \) loses to \( q_{\text{best}} \)

So all surviving \( q \) are \( \leq 5\varepsilon \)-close to \( q_{\text{best}} \) \( \Rightarrow \) \( \leq 6\varepsilon \)-close to \( p \).

Need all matches to give correct output — union bound on \( |C| \) matches.
Applications:

Example 1: learning distribution of a coin

domain = \{0,1\}

need to learn bias

Here \( D = \mathbb{R} \)

if use \( C = \{0, \frac{1}{k}, \frac{2}{k}, \ldots, \frac{k-1}{k}, 1\} \)

then 4 bias \( p \), let \( \frac{1}{k} \leq p \leq \frac{k+1}{k} \)

then picking \( \hat{p} = \frac{i}{k} \) gives \( \|p - \hat{p}\|_1 \leq \frac{2}{k} \)

so using \( k = \Theta(\frac{1}{\varepsilon}) \) gives \( \|p - \hat{p}\|_1 \leq \varepsilon \)

\( |C| = k+1 \) # samples needed by cover method is \( \Theta(\frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon}) \)
Example 2: 2-bucket distributions

now need to specify \( \alpha \) and \( \beta \)

so \( C = \sum (i, j) | i, j \in \mathbb{Z}, ... , n \) \n
\( |C| = \Theta((\frac{1}{\varepsilon})^2) \)

\# samples is \( O(\frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon}) \)

Example 3: monotone distributions

Birge \( \Rightarrow \) \( C = \sum (\frac{i}{k}, ... , \frac{i(\log n)}{\varepsilon k}) | i, j, ... \in 0...k \) \n
\( |C| = \Theta\left(\frac{1}{\varepsilon^3 \log n \log \frac{1}{\varepsilon}}\right) \Rightarrow \# \text{ samples is } O(\frac{1}{\varepsilon^3 \log n \log \frac{1}{\varepsilon}}) \)