Lecture 16: Hypothesis Testing
Some Problems: (Given samples of $p$)

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

- $p$ $\epsilon$-close to $q$
  - or $2\epsilon$-far from $q$

(Given samples of $q$) is $p = q$

- or $p$ $\epsilon$-far from $q$

(Given samples of $q$) is $p$ $\epsilon$-close to $q$

- or $2\epsilon$-far from $q$

Complexity (in terms of $n = |D|$)

- $\sqrt{n}$

- $\frac{n}{\log n}$

- $2^{\frac{1}{3}} n$

- $\frac{n}{\log n}$

- $\sqrt{n}$

- $\frac{n}{\log n}$
Other problems considered:

estimate entropy, support size, independence?

represented well via K-histogram?

monotone hazard rate

•

•
A useful tool:

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

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

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

Question:

- How many samples needed in terms of \( |\mathcal{H}| \) and domain size?
- Is this the same as testing closeness, uniformity?
- Do lower bounds apply?

\( \text{NO! guaranteed } p \text{ close to some } q \in \mathcal{H} \)
What we want:

Given \( h_1, h_2 \) explicit distributions \( p \) 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 \( \mathcal{H} \)

\( \mathcal{H} \) \& \( p \) via samples

find \( h \in \mathcal{H} \) closest to \( p \)

or pretty close to best
Find best hypothesis via "tournament"?

"winner" advances at each phase.

\[
\begin{align*}
&h_1, h_2, h_3, h_4, h_5, h_6, \ldots, h_{m_*}, h_m \\
&\downarrow \quad \downarrow \quad \downarrow \\
&h_5 \\
&\downarrow \\
&h_3 \\
&\downarrow \\
&h_1 \\
&\downarrow \\
&h_2 \\
&\downarrow \\
&p
\end{align*}
\]

maybe \( p = h_2 \)

\[
\begin{align*}
&\| p - h_1 \| < 2\varepsilon + h_1 \text{ "wins"} \\
&\| p - h_2 \| = 2\varepsilon + h_2 \text{ "wins"} \\
&\| p - h_3 \| = 3\varepsilon + h_3 \text{ "wins"}
\end{align*}
\]

overall winner \( \leq O(\log n, \varepsilon) \) distance from best hypothesis?
• 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 close to p
A "subtool" for comparing two hypotheses:

**Theorem**

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 $O(\log(\frac{1}{\delta'})/(\varepsilon')^2)$ 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 + one is really far ($\geq 12\varepsilon'$) we will output close one

- if one is $\varepsilon'$ close other is $10\varepsilon'$ close output either one?
  - but both $\leq 10\varepsilon'$ close so not too bad

getting kind of complicated just to specify?
Actually a bit stronger: (focus on case where \( \varepsilon \) is close)

\[
\text{Thm} \quad \hat{p} \text{ given via samples } \quad h_1, h_2 \text{ fully known and } \hat{p} \text{ is } \varepsilon'\text{-close to at least one of } h_1, h_2
\]

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

1. If \( h_i \) more than \( 12\varepsilon'\)-far from \( \hat{p} \), unlikely to output \( h_i \) as winner
2. If \( h_i \) more than \( 10\varepsilon'\)-far from \( \hat{p} \), unlikely to output \( h_i \) as winner

\( \text{\textbf{BUT COULD TIE}} \)
Proof of subtool:

**Idea:**
- wlog, $h_1$ is $\epsilon$-close to $p$
- if $h_2$ is close to $h_1$, then $h_2$ is pretty close to $p$
- it can win or tie
- else $h_2$ is far from $h_1$, then also far from $p$

- red+green areas are big

- estimate $\Pr[x \in A]$ see if it is more like $h_1$ or more like $h_2$

$A = \text{part of domain where } h_1(x) > h_2(x)$
Algorithm: Choose: \( \text{Input } p, h_1, h_2 \)

First some definitions:

\[ A = \sum_{x \in A} \left| h_1(x) - h_2(x) \right| \]
\[ a_1 = h_1(A) \quad \text{← red + blue areas} \]
\[ a_2 = h_2(A) \quad \text{← blue area} \]
\[ \|h_1 - h_2\|_1 = 2(a_1 - a_2) \]

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

2. draw \( m = 2\log_{\frac{1}{8\epsilon'}} \) samples \( S_1, \ldots, S_m \) from \( P \)

3. \( \lambda = \frac{1}{m} \left| \sum_{i=1}^{m} S_i \in A \right| \quad \text{← estimates} \frac{\text{Pr} \left[ x \in A \right]}{\text{Pr} \left[ x \in A^c \right]} \)

4. if \( \lambda > a_1 - \frac{3}{2}\epsilon' \) return \( h_1 \)
   else if \( \lambda < a_2 + \frac{3}{2}\epsilon' \) return \( h_2 \)
   else declare "tie"

\[ C = \sum_{x} \min \left( h_1(x), h_2(x) \right) \]
\[ \text{green} = \sum h_2(x) - C \]
\[ = 1 - C \]
\[ \text{red} = \sum h_1(x) - C \]
\[ = 1 - C \]
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 to each other
  \[\Rightarrow \delta \leq \varepsilon' \Rightarrow \|h_1 - h_2\| \leq 10\varepsilon'\]
  
  $\Rightarrow$ both $\leq \|h_1 - h_2\| = 2\varepsilon'$

  \[\Rightarrow \text{both } \leq 10\varepsilon' \Rightarrow \text{"tie" is OK output}\]

  o otherwise reach step 2:

  \[\|h_1 - h_2\| > 10\varepsilon'\] (if $a_1 > a_2$)

  \[\|h_1 - h_2\| > 10\varepsilon'\] (if $a_1 > a_2$)

Algorithm: Choose:

\[A = \frac{1}{\varepsilon} \times \frac{1}{\varepsilon} \geq h_2(x)^2\]

$a_i = h_i(A)$

1. if \(a_i - a_2 \leq \varepsilon\) declare "tie" + return $h$

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

3. \[\lambda - \frac{1}{m} \leq \lambda \leq \frac{1}{m}\]

4. if $\lambda > a_i - \frac{3}{2} \varepsilon'$ return $h_1$

else if $\lambda < a_2 + \frac{3}{2} \varepsilon'$ return $h_2$

else declare "tie" + return $h$

\[\text{if } p = h_1, \ E[A] = a_1\]
\[\text{if } p = h_2, \ E[A] = a_2\]

\[\text{green area } = \text{red area } = a_1 - a_2\]

\[\text{green + red area } = a_2\]

\[\text{blue } = a_2\]

\[\text{blue + red area } = a_1\]
Why does it work?

- $h_1$ or $h_2$ is $\epsilon'$-close to $A$ (given)
- If "tie" in step 1, algorithm does right thing
- Otherwise reach step 2: $\|h_1 - h_2\| \geq 10\epsilon'$ ($a_1 - a_2 \geq 5\epsilon'$)

**Algorithm Choose:**

\[
A = \frac{1}{2} \left( h_1(x) + h_2(x) \right)
\]
\[
a_i = h_1(A)
\]
\[
a_2 = h_2(A)
\]
\[
\text{note } \|h_1 - h_2\|_1 = 2(a_1 - a_2)
\]

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

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

3. $x \leftarrow \frac{1}{m} \sum_i 1_{s_i \in A}\$

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

\[
\begin{align*}
\text{if } p \neq h_1, & \quad E[\alpha] = a_1 \\
\text{if } p = h_1, & \quad E[\alpha] = a_2
\end{align*}
\]

- Green area = red area = $a_1 - a_2$
- Light green area = $a_2$
- Green + red area = $a_1$
- Blue + red area = $a_1$
The cover method - a method for learning distributions

**Def.** $C$ is an "$\epsilon$-cover" of $\mathcal{D}$ if:

$$
\forall \ p \in \mathcal{D} \\
\exists \ q \in C \text{ s.t. } \|p - q\|_1 \leq \epsilon
$$

Why useful? Hopefully $C$ much smaller than $\mathcal{D}$ allows to approx $\mathcal{D}$

Note $C$ not unique

**Thm.** $\exists$ algorithm, given $\phi \in \mathcal{D}$ which takes $O(\frac{1}{\epsilon^2} \log |C|)$ samples of $p$ + outputs $h \in C$

s.t. $\|h - p\|_1 \leq 6\epsilon$ with prob $\geq \frac{9}{10}$
Theorem 3: algorithm, given $\phi \in \mathcal{D}$, which takes $O(\frac{1}{\varepsilon^2} \log |\mathcal{C}|)$ samples of $p$ and outputs $h \in \mathcal{C}$ such that $\|h - p\|_1 \leq 6\varepsilon$ with prob $\geq \frac{9}{10}$

Proof:

Since $p \in \mathcal{D}$, $\exists q_{opt} \in \mathcal{C}$ such that $\|p - q_{opt}\|_1 \leq \varepsilon$

Run "Choose" on $p$ with every pair $q_1, q_2 \in \mathcal{C}$.

$q_{opt}$ wins or ties all games.

If $q_1$ is $\geq 6\varepsilon$-far from $p$, then $\geq 5\varepsilon$-far from $q_{opt}$.

Equivalently, if $q_1$ wins or ties all games,

\[ \Rightarrow \leq 5\varepsilon \text{ far from } q_{opt} \leq 6\varepsilon \text{ far from } p \]

Need all matches to give correct output (estimate of $\mathcal{D}$)

Union bound on $O(\frac{1}{\varepsilon^2} \log |\mathcal{C}|)$ many matches.
Applications:

Example 1: learning distribution of a coin

domain = \{0, 1\}^3

need to learn bias

Here \( \Theta = [0, 1] \)

if use \( C = \{0, 1, \frac{1}{3}, \frac{2}{3}, \frac{3}{3}, \ldots, \frac{k}{3}\} \)

A bias \( \hat{p} \) let \( \frac{i}{K} \leq \hat{p} \leq \frac{i+1}{K} \)

So pick \( \hat{p} \leftarrow \text{closest} \frac{i}{K} \) then \( |p - \hat{p}| = \frac{1}{K} \leq \varepsilon \)

picking \( k = \Theta \frac{1}{\varepsilon} \)

\( |C| = k + 1 = \Theta \frac{1}{\varepsilon} \)

need \( O\left(\frac{1}{\varepsilon^2 \log \frac{1}{\varepsilon}}\right) \) samples

using cover method
Example 2: 2-bucket distributions

now need to specify $\alpha$ and $\beta$

so $C = \sum \left( \frac{i}{k}, \frac{j}{k} \right)$ for $i, j \in \{0, \ldots, k\}$

$|C| = \Theta\left(\frac{1}{\varepsilon}^2\right)$

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

Example 3: monotone distributions

Birge $\Rightarrow$ $C = \sum \left( \frac{i_1}{k}, \ldots, \frac{i_{\log n}}{k} \right)$ for $i_1, i_2, \ldots \in \{0, \ldots, k\}$

$|C| = \Theta\left(\frac{1}{\varepsilon^3 \log n} \varepsilon^2\right)$ $\Rightarrow$ # samples is $O\left(\frac{1}{\varepsilon^3} \log n \log \frac{1}{\varepsilon}\right)$