Lecture 16:

Hypothesis Testing

Some Problems: (Given samples of p)

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

	Complexity
is $p = q$ (e.g. $q = U_D$) or ε-far from q	$\sqrt{n}$
is p ε-close to q or ε-far from q	$\frac{n}{\log n}$
(Given samples of q) is $p = q$ or p ε-far from q	$n^{2/3}$
(Given samples of q) is p ε-close to q or ε-far from q	$\frac{n}{\log n}$
is p monotone or ε-far from monotone	$\sqrt{n}$
is p ε-close to monotone or ε-far from monotone	$n/\log n$

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 $\underbrace{\exists\, q \in \mathcal{H} \text{ for which}}_{\mathcal{H} \text{ contains a good approx to } p}$ dist(p,q) is small

$\Leftarrow$ strong assumption

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

Question:

How many samples needed in terms of $|\mathcal{H}|$ + domain size?

Is this the same as testing closeness, uniformity?
Do lower bounds apply?

NO!

$\Big\}$ p is guaranteed to be close to some $q \in \mathcal{H}$

What we want:

Given h_1, h_2 explicit

p via samples

procedure that outputs h_i that is closer to p

What if both are roughly same distance?

maybe either one is ok?

or maybe not...

More general Goal:

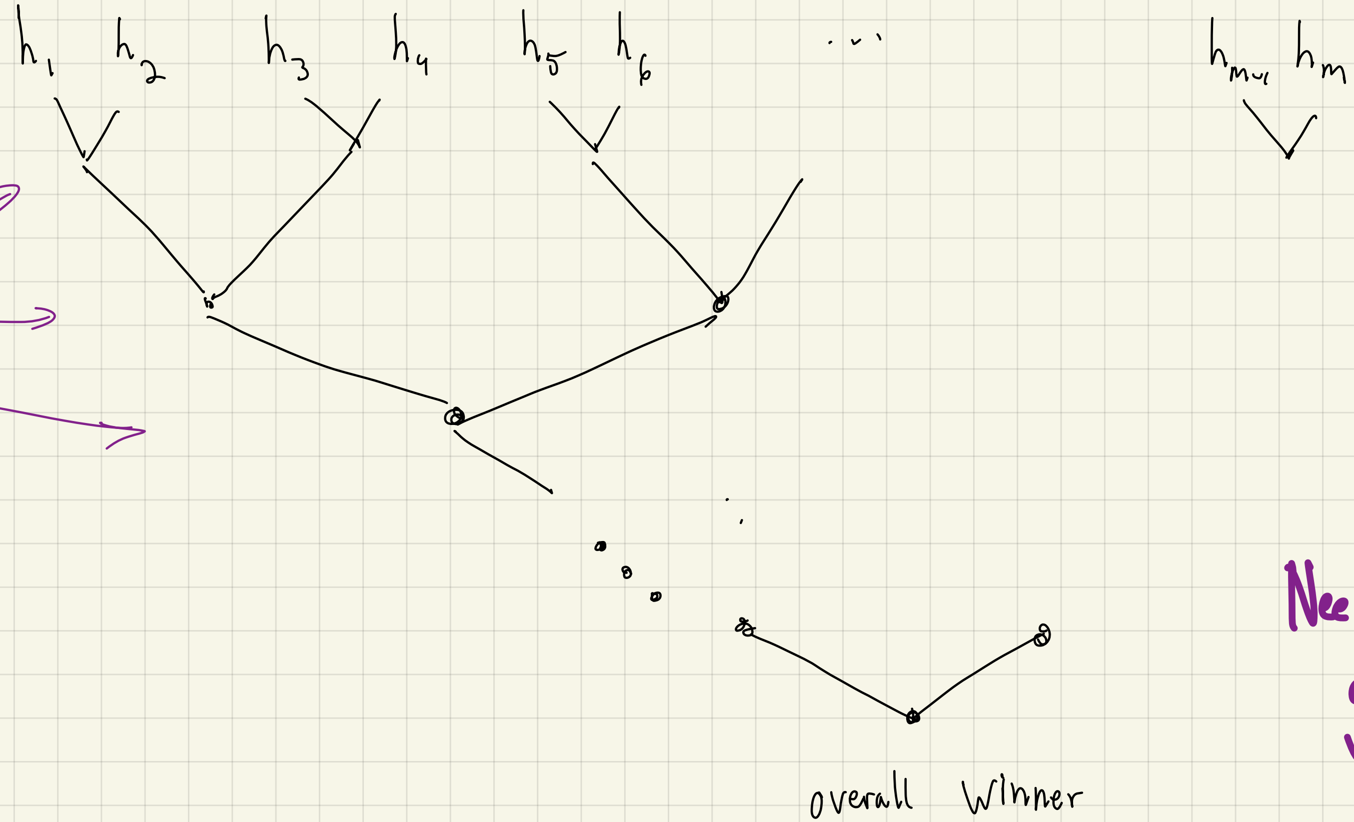
Given set of hypotheses $\mathcal{H}$

+ p via samples

find $h \in \mathcal{H}$ closest to p

Find best hypothesis via "tournament"?

"winner" advances at each phase



Need stronger guarantee!

maybe $p = h_1$

$\|p-h_2\|_1 = \varepsilon$ & h_2 "wins"

then $\|p-h_3\|_1 = 2\varepsilon$ & h_3 "wins"

then $\|p-h_5\|_1 = 3\varepsilon$ & h_5 "wins"

⋮

$\Bigg\}$ overall winner could be $O(\log n \cdot \varepsilon)$ far from best hypothesis?

- won't use simple tournament ← instead compare every pair
- will add notion of "tie"

Output hypothesis that <u>wins</u> or <u>ties</u> every match

(hopefully there is one, & it is the right one)

A "subtool" for comparing two hypotheses:

<u>Thm</u> given (1) sample access to p

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

(3) accuracy parameter ε', confidence parameter δ'

then Algorithm "choose" takes $O(\log(\frac{1}{\delta'})/(\varepsilon')^2)$ samples + 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 ε'-close + one really far (e.g. $\geq 12\varepsilon'$), will output ε'-close hypothesis

if both ε'-close (e.g. one is ε'-close, other is $\leq 10\varepsilon'$-close) then output $12\varepsilon'$-close hypothesis

if at least one is close, will output pretty close hypothesis

getting kind of complicated just to specify ☹

Actually a bit stronger:

<u>Thm</u> p given via samples
h_1, h_2 fully known & p is ε'-close to at least one of h_1, h_2
ε', δ' given
Algorithm "choose" takes $O\left(\left(\log \frac{1}{\delta'}\right)\left(\frac{1}{\varepsilon'}\right)^2\right)$ samples & outputs $h \in \{h_1, h_2\}$ such that:

(1) If h_i more than $\underbrace{12\varepsilon'\text{-far}}_{\text{very bad}}$ from p, $\underbrace{\text{unlikely}}_{2e^{-m(\varepsilon')^2/2}}$ to output h_i as winner <u>or</u> tie

(2) If h_i more than $\underbrace{10\varepsilon'\text{-far}}_{\text{not that bad}}$ from p, unlikely to output h_i as winner

might tie but won't win

Can use $\varepsilon' \approx \frac{\varepsilon}{10}$?

Proof of subtool:

idea: wlog h_1 is ε'-close to p

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

h_1 & h_2 are close
can determine h_1 & h_2 close w/o samples from p

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 & h_2
& will output h_1

Since you know h_1 & 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: Input p, h_1, h_2

samples — explicit description

First some definitions:

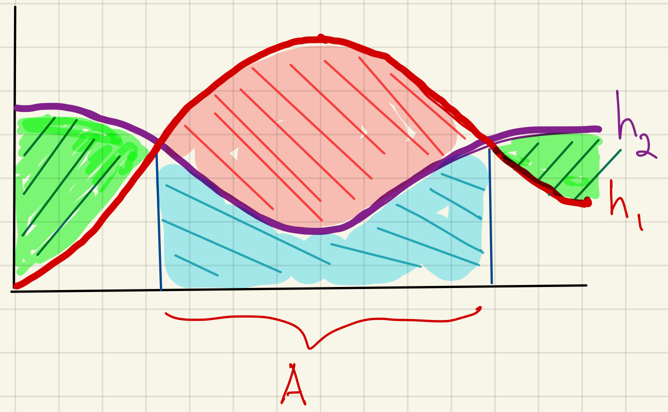
$$A = \{x \mid h_1(x) > h_2(x)\}$$

$a_1 = h_1(A)$ ← red + blue areas

$a_2 = h_2(A)$ ← blue area

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

will give factor of 2 in constants



green area = red area = $a_1 - a_2$

L_1-dist = green + red = 2·red

1. if $\underbrace{a_1 - a_2}_{\frac{1}{2}L_1 \text{ distance}} \le 5\varepsilon'$ declare "tie" & return h_1

   (no samples needed)

2. draw $m = \dfrac{2 \log \frac{1}{\delta'}}{(\varepsilon')^2}$ samples $s_1 \ldots s_m$ from p

3. $\alpha \leftarrow \frac{1}{m} \left| \{ i \mid s_i \in A \} \right|$

   if $p = h_1$, $E[\alpha] = a_1$

   if $p = h_2$, $E[\alpha] = a_2$

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_1

another additive error in constants

Why does it work?

- h_1 or h_2 is ε'-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 $\le 11\varepsilon'$-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:

$A = \{x \mid h_1(x) > h_2(x)\}$

$a_1 = h_1(A)$

$a_2 = h_2(A)$

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

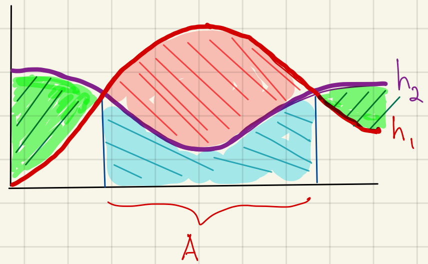
1. if $a_1 - a_2 \le 5\varepsilon'$ declare "tie" + return h (no samples needed)
2. draw $m = \frac{2 \log \frac{1}{\delta'}}{(\varepsilon')^2}$ samples $s_1 \dots s_m$ from p
3. $\alpha \leftarrow \frac{1}{m} |\{i \mid s_i \in A\}|$
4. if $\alpha > a_1 - \frac{3}{2}\varepsilon'$ return h_1

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

   else declare "tie" + return h_1

if $p = h_1$, $E[\alpha] = a_1$

if $p = h_2$, $E[\alpha] = a_2$



green area = red area = $a_1 - a_2$

L_1-dist = green + red

blue area = a_2

blue + red area = a_1

Why does it work?

- h_1 or h_2 is ε'-close to A (given)

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

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

$$E[\alpha] = \Pr_{x \in p}[x \in A] \equiv p(A)$$

assume (Chernoff) that with high prob $|\alpha - E[\alpha]| \le \frac{\varepsilon'}{2}$

h_1 assigns a_1 weight to A
h_2 " a_2 " " A

if p is ε'-close to h_1, assigns $\ge a_1 - \varepsilon'$ weight to A

$$\& \quad \alpha \ge a_1 - \varepsilon' - \frac{\varepsilon'}{2} = a_1 - \frac{3\varepsilon'}{2}$$ return h_1 whp

" " " " " " h_2, " $\le a_2 + \varepsilon'$ weight to A

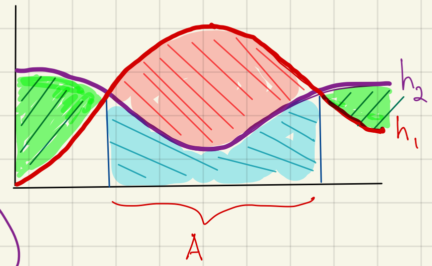
$$\& \quad \alpha \le a_2 + \varepsilon' + \frac{\varepsilon'}{2} \le a_2 + \frac{3\varepsilon'}{2}$$ return h_2 whp

Algorithm Choose:

$A = \{x \mid h_1(x) > h_2(x)\}$
$a_1 = h_1(A)$
$a_2 = h_2(A)$
note $\|h_1 - h_2\|_1 = 2(a_1 - a_2)$

1. if $a_1 - a_2 \le 5\varepsilon'$ declare "tie" + return h (no samples needed)
2. draw $m = \frac{2 \log \frac{1}{\delta'}}{(\varepsilon')^2}$ samples $s_1 \dots s_m$ from p
3. $\alpha \leftarrow \frac{1}{m} |\{i \mid s_i \in A\}|$
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_1

{ if $p = h_1$, $E[\alpha] = a_1$
{ if $p = h_2$, $E[\alpha] = a_2$



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

The cover method - a method for learning distributions

def. $\mathcal{C}$ is an "ε-cover" of $\mathcal{D}$ if $\forall p \in \mathcal{D}$ $\exists q \in \mathcal{C}^{\mathcal{D}}$ s.t. $\|p-q\|_1 \leq \varepsilon$

smaller set of distributions

big set of distributions

why useful?

hopefully $\mathcal{C}$ is much smaller than $\mathcal{D}$, allows us to approximate $\mathcal{D}$

note $\mathcal{C}$ not unique

Thm $\exists$ algorithm, given $p \in \mathcal{D}$, which takes $O\left(\frac{1}{\varepsilon^2} \log |\mathcal{C}|\right)$ samples of p + outputs $h \in \mathcal{C}^{\mathcal{D}}$ s.t. $\|h-p\|_1 \leq 6\varepsilon$ with prob $\geq \frac{9}{10}$

big improvement: union bnd over size of $\mathcal{C}$ not $\mathcal{D}$!

Thm $\exists$ algorithm, given $p \in \mathcal{D}$, which takes

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

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

Pf.

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

(could be more than one)

run "Choose" on p with every pair $q_1, q_2 \in \mathcal{C}^{\mathcal{D}}$

if best q_{OPT} doesn't win all of it's "matches" then it ties

with others that are not so bad

if q' is $\geq 6\varepsilon$-far from p, then $\geq \underbrace{6\varepsilon-\varepsilon}_{5\varepsilon}$-far from best q_{OPT}

$\Rightarrow$ loses to q_{OPT}

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

need all matches to give correct output — union bound on $\binom{|\mathcal{C}|}{2}$ matches

$\square$

Applications:

Example 1: learning distribution of a coin

domain $= \{0,1\}$

need to learn bias

Here $\mathcal{D} = \mathbb{R}$

biases of coin

if use $\mathcal{C} = \{0, \frac{1}{k}, \frac{2}{k}, \ldots, \frac{k-1}{k}, 1\}$

then $\forall$ bias p, let $\frac{i}{k} \le p \le \frac{i+1}{k}$

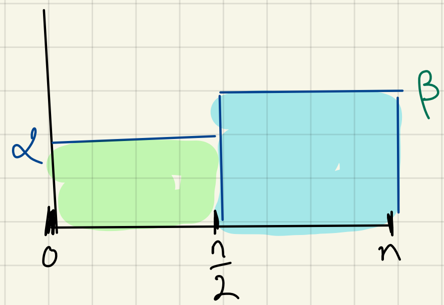
then picking $\tilde{p} = \frac{i}{k}$ gives $\|p - \tilde{p}\|_1 \le \frac{2}{k}$

So using $k = \Theta(\frac{1}{\varepsilon})$ gives $\|p - \tilde{p}\|_1 \le \varepsilon$

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

samples needed by cover method is
$O\left(\frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon}\right)$

Example 2: 2-bucket distributions



now need to specify α and β

so $\mathcal{C} = \left\{ \left(\frac{i}{k}, \frac{j}{k}\right) \middle| \; i, j \in \{0, \dots, k\} \right\}$

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

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

Example 3: monotone distributions

Birge $\Rightarrow$ $\mathcal{C} = \left\{ \left(\frac{i_1}{k}, \dots, \frac{i_{(\log n/\varepsilon)}}{k}\right) \middle| \; i_1, i_2, \dots \in \{0 \dots k\} \right\}$

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