Hypothesis Testing

Some Problems: (Given samples of p)	Complexity (in terms of $n = \lvert D \rvert$)
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 $\underbrace{\exists\, q \in \mathcal{H} \text{ for which}}_{\mathcal{H} \text{ contains a good approx to } p}$ $\text{dist}(p,q)$ is small ⟸ strong assumption

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

maybe ok to output h_i as long as it is within ε of closest h_j to p?

More general Goal:

Given set of hypotheses $\mathcal{H}$

+ p via samples

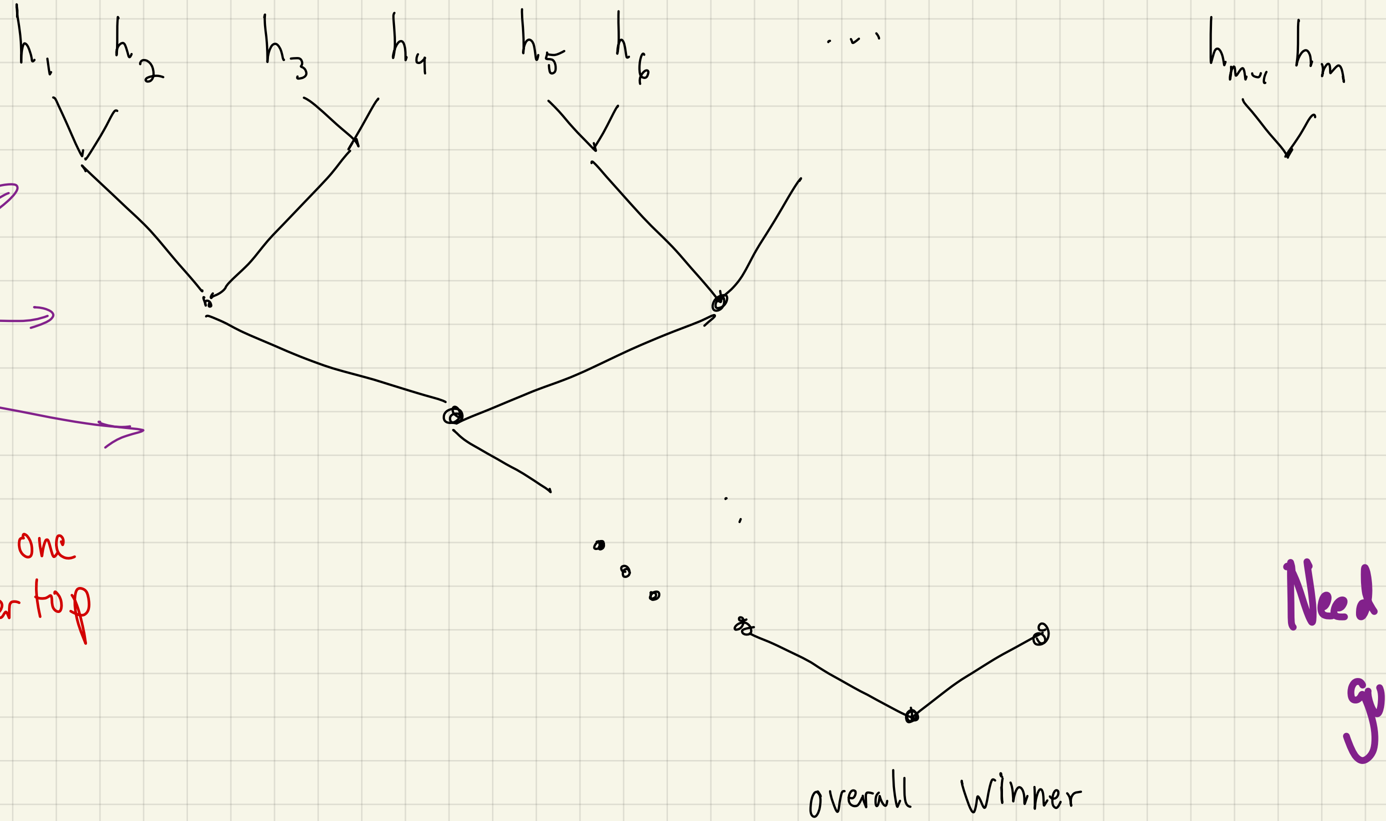
find $h \in \mathcal{H}$ closest to p

Find best hypothesis via "tournament"?

"winner" advances at each phase

or: at least ε-close to winner

i.e. if $|h_1 - h_2| < \varepsilon$ can output either one else output closer to p



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"

⋮

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

How to fix:

- 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 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 $\|h_j - p\|_1 < 12\varepsilon'$

ie. if both h_1, h_2 far, no guarantees

if one ε'-close & one really far, will output ε'-close hypothesis

e.g. $\geq 12\varepsilon'$

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

e.g. one is ε'-close other is $\leq 10\varepsilon'$-close

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

getting kind of complicated just to specify ☹

Actually a bit stronger:

Thm 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 or tie

will not actually use this

(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

(3) If h_i ties whp then $\|h_i - h_j\| \leq 10\varepsilon' \Rightarrow \|h_i - p\| \leq 11 \cdot \varepsilon'$

Can use $\varepsilon' \approx \frac{\varepsilon}{12}$?

Proof of subtool:

h₁ & h₂ are close
can determine h_1 & h_2 close w/o samples from p

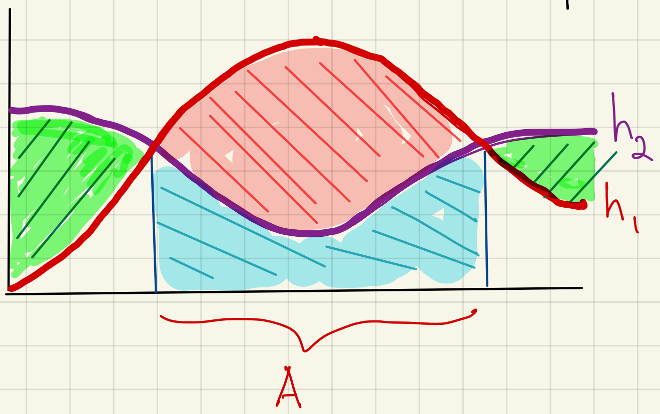
idea: wlog h_1 is ε'-close to p

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

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)

samples

explicit description

Algorithm Choose: Input p, h_1, h_2

First some definitions:

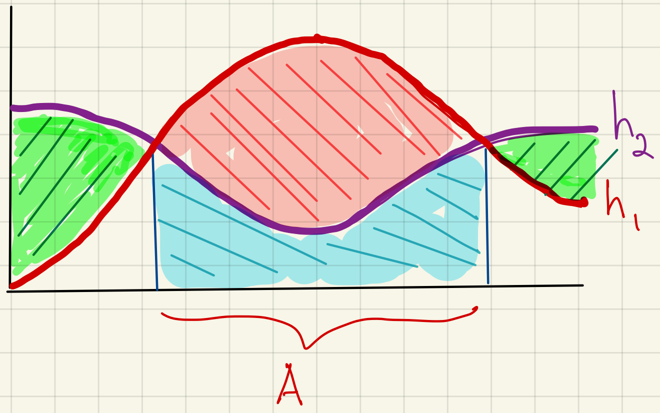
$A = \{x \mid h_1(x) > h_2(x)\}$

$a_1 = h_1(A)$ ← 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

(note that A is not necessarily "consecutive")

$\|h_1 - h_2\|_1 \leq 10\varepsilon'$

1. if $\underbrace{a_1 - a_2}_{\frac{1}{2}L_1 \text{ distance}} \leq 5\varepsilon'$ 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 →

is p more like h_1 or h_2 in A?

note need "5" to be bigger than $2 \cdot 3/2 = 3$

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 $\leq 11\varepsilon'$-close to A (note "12" should be $\geq$ "11")

so "tie" is ok

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

Algorithm Choose:

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

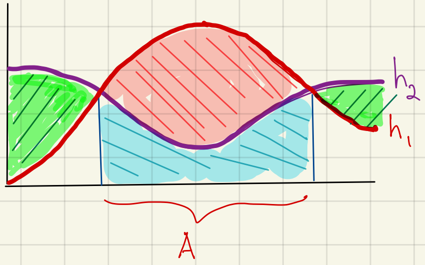
$a_1 = h_1(A)$

$a_2 = h_2(A)$

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

```
1. if $a_1 - a_2 \leq 5\varepsilon'$ 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 $h_1$
```

$\begin{cases} \text{if } p = h_1, & E[\alpha] = a_1 \\ \text{if } p = h_2, & E[\alpha] = a_2 \end{cases}$



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

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

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

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

Algorithm Choose:

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

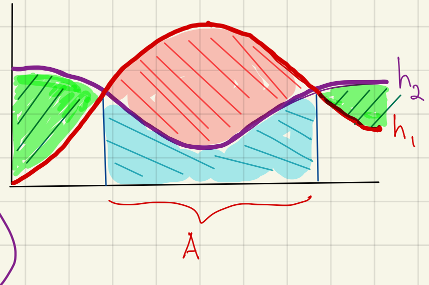
$a_1 = h_1(A)$

$a_2 = h_2(A)$

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 $h_1$
```

$\begin{cases} \text{if } p = h_1, & E[\alpha] = a_1 \\ \text{if } p = h_2, & E[\alpha] = a_2 \end{cases}$



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 $\forall p \in \mathcal{D}$

$\exists q \in \mathcal{C}$ s.t. $\|p-q\|_1 \le \varepsilon$

$\uparrow$ smaller set of distributions (specific to $\mathcal{D}$)

$\uparrow$ 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

big improvement: union bnd over size of $\mathcal{C}$ not $\mathcal{D}$! $\Longrightarrow$ $O\left(\frac{1}{\varepsilon^2}\log|\mathcal{C}|\right)$ samples of p + outputs $h \in \mathcal{C}$

s.t. $\|h-p\|_1 \le 12\varepsilon$ with prob $\ge \frac{9}{10}$

Thm $\exists$ algorithm, given $p \in \mathcal{D}$, which takes $O\left(\frac{1}{\varepsilon^2} \log |\mathcal{C}|\right)$ samples of p + outputs $h \in \mathcal{C}$ s.t. $\|h-p\|_1 \le 12\varepsilon$ with prob $\ge \frac{9}{10}$

Pf.

Since $p \in \mathcal{D}$, $\exists$ $q \in \mathcal{C}$ s.t. $\|p-q\|_1 \le \varepsilon'$ (could be more than one)

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

- best q_{opt} either wins or ties all matches
  ties are to other q's which have low error
  $\le$ error of $q_{opt} + 10\varepsilon'$
  $\le 11\varepsilon'$
- if q' is $\ge 12\varepsilon$-far from p,
  $\Rightarrow$ loses to q_{OPT} (doesn't survive)

So all surviving q are $\le 11\varepsilon'$-close to q_{opt} + $\le 12\varepsilon$-close to p

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

Applications:

Example 1: learning distribution of a coin

domain = $\{0, 1\}$

need to learn bias

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

biases of coin

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

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

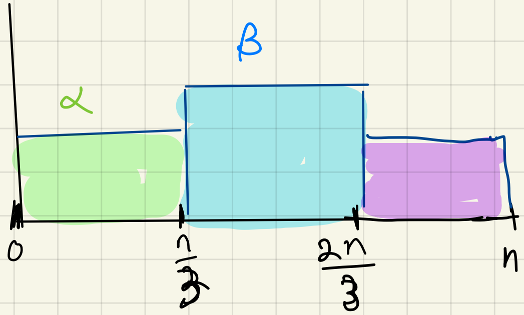
then picking $\tilde{p} = \frac{i}{k}$ gives $\|p - \tilde{p}\|_1 \leq \frac{2}{k}$

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

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

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

Example 2: 3 - bucket distributions



now need to specify α and β

so $\mathcal{C} = \{ (\frac{i}{k}, \frac{j}{k}) \mid i, j \in \{0, \dots, k\}\}$

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

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

Example 3: monotone distributions

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

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

Example 4: Poisson Binomial Distributions

$PBD(p_1 \cdots p_n)$: $X = \sum_{i=1}^{n} X_i$ X_i independent r.v.'s

$E[X_i] = p_i$ (<u>not</u> identically distributed)

e.g. 1) all p_i's $= 1/2$ $X \sim$ Binomial

2) $p_1 = 1/2$ $p_2 = 1$ $p_3 = \ldots = p_n = 0$

$\Pr[X=0] = 0$

$\Pr[X=1] = 1/2$ $X \sim 1 + \textcircled{\$}$

$\Pr[X=2] = 1/2$

$\Pr[X>2] = 0$

Birge bucketing twice with $O(\frac{\log n}{\varepsilon})$ choices for breakpoint

↓

PBD unimodal $\Longrightarrow$ $O(\frac{1}{\varepsilon^3} \log n)$ samples

Structure Thm:

PBD "looks like" (to w/in ε L_1-error) either:

1) [$\frac{1}{\varepsilon}$-sparse] supported almost completely (as fctn of ε] on interval of length $O(1/\varepsilon^3)$

$\Longrightarrow$ small cover $\left(\frac{1}{\varepsilon}\right)^{O(\log^2(1/\varepsilon))}$ $\Longrightarrow$ $\frac{1}{\varepsilon^2} \log^3(1/\varepsilon)$ learner $\Big\}$ can test in $O(1/\varepsilon^{3/2})$

testing:

2) [$\frac{1}{\varepsilon}$-heavy] PBD looks like translated binomial on large enough # vars

easy to learn → $\Longrightarrow$ learn binomial, which puts almost all wt on $\sqrt{n}$ places in middle $\Big\}$ can test in $O(n^{1/4})$!