Today

Boosting and Experts.

Routing and Experts.

Poll.

Given a weak learning method (produce ok hypotheses.)
produce a great hypothesis.

Can we do this?

(A) Yes
(B) No

If yes. How?

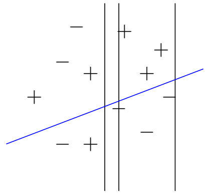
Multiplicative Weights!

The endpoint to a line of research.

Learning.

Learning just a bit.

Example: set of labelled points, find hyperplane that separates.



Looks hard.

Get $1/2$ on correct side? Easy.
Arbitrary line. And Scan.

Useless. A bit more than $1/2$

Weak Learner: Classify $\geq \frac{1}{2} + \varepsilon$ points correctly.

Not really important but ...

Experts Picture

Weak Learner/Strong Learner

Input: n labelled points.

Weak Learner:
produce hypothesis correctly classifies $\frac{1}{2} + \varepsilon$ fraction

Strong Learner:
produce hypothesis correctly classifies $1 + \mu$ fraction
That's a really strong learner!
produce hypothesis correctly classifies $1 - \mu$ fraction

Same thing?

Can one use weak learning to produce strong learner?

Boosting: use a weak learner to produce strong learner.

Boosting/MW Framework

Experts are points. "Adversary" weak learner.

Points want to be misclassified.

Learner wants to maximize probability
of classifying random point correctly.
Strong learner algorithm will come from adversary.

Do $T = \frac{2}{\gamma^2} \log \frac{1}{\mu}$ rounds

1. Row player: multiplicative weights($1-\gamma$) on points.
2. Column: run weak learner on row distribution.
3. Hypothesis $h(x)$: majority of $h_1(x), h_2(x), \ldots, h_T(x)$.

Claim: $h(x)$ is correct on $1-\mu$ of the points ! ! !

Cool!

Really? Proof?

Some intuition

Intuition 1: Each point classified correctly independently in each round with probability $\frac{1}{2}+\varepsilon$.

After enough rounds, majority rule correct for almost all points.

Intuition 2:
Say some point classified correctly $\leq 1/2$ of time.

High probability of choosing such point in distribuiont.

In limit, whole distribution becomes such point.

This subset will be classified correctly with probability $1/2+\varepsilon$.

Some details...

Weak learner learns over distributions of points not points.

Make copies of points to simulate distributions.

Used often in machine learning.

Adaboost proof.

Claim: $h(x)$ is correct on $1-\mu$ of the points ! !

Let S_{bad} be the set of points where $h(x)$ is incorrect.

majority of $h_t(x)$ are wrong for $x \in S_{bad}$.

$x \in S_{bad}$ is a good expert – loses less than $\frac{1}{2}$ the time.

$W(T) \geq (1-\varepsilon)^{\frac{T}{2}} |S_{bad}|$

Each day, weak learner gets $\geq \frac{1}{2}+\gamma$ payoff.

$\to L_t \geq \frac{1}{2}+\gamma$.

$\to W(T) \leq n(1-\varepsilon)^L \leq ne^{-\varepsilon L} \leq ne^{-\varepsilon(\frac{1}{2}+\gamma)T}$

Combining

$|S_{bad}|(1-\varepsilon)^{T/2} \leq W(T) \leq ne^{\varepsilon(\frac{1}{2}+\gamma T)}$

Example.

Set of points on unit ball in d-space.

Learner: learns hyperplanes through origin.

Can learn if
there is a hyperplane, $\mathscr{H}$, that separates all the points.

and find $\frac{1}{2}+\varepsilon$ weighted separating plane.

Experts output is average of hyperplanes ...a hyperplane!

$\frac{1}{2}+\varepsilon$ separating hyperplane?
Assumption: margin γ.

Random hyperplane?
Not likely to be exactly normal to $\mathscr{H}$.
Should get $\frac{1}{2}+\gamma/\sqrt{d}$

$O(\frac{d \log n}{\gamma^2})$ to find separating hyperplane.

Weak learner: random Wow. That's weak.

Calculation..

$|S_{bad}|(1-\varepsilon)^{T/2} \leq ne^{\varepsilon(\frac{1}{2}+\gamma)}T$

Set $\varepsilon = \gamma$, take logs.

$\ln\left(\frac{|S_{bad}|}{n}\right) + \frac{T}{2}\ln(1-\gamma) \leq -\gamma T(\frac{1}{2}+\gamma)$

Again, $-\gamma-\gamma^2 \leq \ln(1-\gamma)$,

$\ln\left(\frac{|S_{bad}|}{n}\right) + \frac{T}{2}(-\gamma-\gamma^2) \leq -\gamma T(\frac{1}{2}+\gamma) \to \ln\left(\frac{|S_{bad}|}{n}\right) \leq -\frac{\gamma^2 T}{2}$

And $T = \frac{2}{\gamma^2}\log\frac{1}{\mu}$,

$\to \ln\left(\frac{|S_{bad}|}{n}\right) \leq \log\mu \to \frac{|S_{bad}|}{n} \leq \mu$.

The misclassified set is at most μ fraction of all the points.

The hypothesis correctly classifies $1-\mu$ of the points ! ! !

Claim: Multiplicative weights: $h(x)$ is correct on $1-\mu$ of the points !

Claim: Weak learning $\to$ strong learning!

not so weak after all.

Better weak learner?

Hyperplane that separates weighted average of +/- points?

Change loss a bit, and get better results.

Toll/Congestion

Given: $G=(V,E)$.
Given $(s_1,t_1)\ldots(s_k,t_k)$.
Row: choose routing of all paths.
Column: choose edge.
Row pays if column chooses edge on any path.

Matrix:
row for each routing: r
column for each edge: e

$A[r,e]$ is congestion on edge e by routing r

Offense: (Best Response.)
Router: route along shortest paths.
Toll: charge most loaded edge.

Defense: Toll: maximize shortest path under tolls.
Route: minimize max congestion on any edge.

Two person game.

Row for every roting. ($A[r,e]$)

An exponential number of rows!

Two person game with experts won't be so easy to implement.

Version with row and column flipped may work.

$A[e,r]$ - congestion of edge e on routing r.

m rows. Exponential number of columns.

Multiplicative Weights only maintains m weights.

Adversary only needs to provide best column each day.

Runtime only dependent on m and T (number of days.)

Congestion minimization and Experts.

Will use gain and $[0,\rho]$ version of experts:

$G \geq (1-\varepsilon)G^* - \frac{\rho \log n}{\varepsilon}$.

Let $T = \frac{k \log n}{\varepsilon^2}$

1. Row player runs multiplicative weights:
   $w_i = w_i(1+\varepsilon)^{g_i/k}$.
2. Route all paths along shortest paths.
3. Output the average of all routings: $\frac{1}{T}\sum_t f(t)$.

Claim: The congestion, c_{max} is at most $(1+\varepsilon)C^* + \varepsilon$.

Proof:
$G \geq G^*(1-\varepsilon) - \frac{k \log n}{\varepsilon}$

$G^* = c_{\max} T$ — Best row payoff against average routing.

$G \leq C^* T$ — each day, gain is average congestion $\leq C^*$
since each day cost is toll solution which is at most C^*
$C^* T \geq c_{max} T(1-\varepsilon) - \frac{k \log n}{\varepsilon}$
For $T = \frac{k \log n}{\varepsilon^2}$
$\rightarrow C^* \frac{1}{1-\varepsilon} + \varepsilon \geq c_{\max}$ plus $\frac{1}{1-\varepsilon} \leq 1+\varepsilon \rightarrow c_{max} - C^* \leq \varepsilon C + \varepsilon$ □

Better setup.

Runtime: $O(km)$ to route in each step.
$O(k \log n(\frac{1}{\varepsilon^2}))$ steps
$\rightarrow O(k^2 m \log n)$ to get a constant approximation.

Homework: $O(km \log n)$ algorithm.

Fractional versus Integer.

Did we solve path routing?
Yes? No?

No! Average of T routings.
We approximately solved fractional routing problem.

No solution to the path routing problem that is $(1+\varepsilon)$ optimal!

Homework 2. Problem 1.

Decent solution to path routing problem?

Randomized Rounding

For each s_i, t_i, choose path p_i with probability $f(p_i)$.

Congestion $c(e)$ edge rounds to $\tilde{c}(e)$.

Edge e.
used by paths $p_1, \ldots, p_m$.
Let $X_i = 1$,
if path p_i is chosen.
otherwise, $X_i = 0$.

Rounded congestion, $\tilde{c}(e)$, is $\sum_i X_i$.

Expected Congestion: $\sum_i E(X_i)$.
$E(X_i) = 1Pr[X_i = 1] + 0Pr[X_i = 0] = f(p_i)$
$\rightarrow \sum_i E(X_i) = \sum_i f(p_i) = c(e)$.
$\rightarrow E(\tilde{c}(e)) = c(e)$.

Concentration (law of large numbers)
$c(e)$ is relatively large ($\Omega(\log n)$)
$\rightarrow \tilde{c}(e) \approx c(e)$.

Concentration results? later.

See you on Thursday.