LECTURE 6

Connection between multiplicative updates of on-line learning algs.

and

in vitro selection

- Loss of variety
  The curse of the multiplicative update

- The power of voting

- Can we select for
  groups of molecules that work together?
IN VITRO SELECTION

PROTEIN

RNA

GOAL: SELECT FOR RNA THAT BINDS TO PROTEIN

SCHEME:

START WITH UNIT AMOUNT OF RNA

LOOP

FUNCTIONAL SEPARATION INTO
GOOD RNA AND BAD RNA

AMPLIFY GOOD RNA TO UNIT AMOUNT
Let's name the unknown RNA strands as $w_1, w_2, \ldots, w_n$ with $n \approx 10^{20}$ (1 liter).

$w_i$: Fraction of RNA $i$.

All $w_i \geq 0$.

Tube represented as vector $(w_1, w_2, \ldots, w_n)$.

When tube has unit amount then $w_1 + w_2 + \ldots + w_n = 1$.

$x_i$: Fraction of RNA $i$ that is good.

$x_i \in [0,1]$.

Protein represented as vector $(x_1, x_2, \ldots, x_n)$.

Vectors $\overline{w}, \overline{x}$ unknown!

Blind computation!
Good RNA in tube $w$
When separation w.r.t. protein $x$:

$$w_1 x_1 + w_2 x_2 + \ldots + w_n x_n = \bar{w} \cdot \bar{x}$$

Bad RNA:

$$w_1 (1-x_1) + w_2 (1-x_2) + \ldots + w_n (1-x_n) = 1 - \bar{w} \cdot \bar{x}$$

Amplification:

Good part of RNA ($w_i x_i$) multiplied by factor $F$.

If precise, the all RNA multiplied by same factor $F$.

Final tube at end of loop:

$$F \cdot \bar{w} \cdot \bar{x}$$

Since final tube has unit amount of RNA:

$$F \cdot \bar{w} \cdot \bar{x} = 1 \quad \text{and} \quad F = \frac{1}{\bar{w} \cdot \bar{x}}$$

Update in each loop:

$$w_i := \frac{w_i x_i}{\bar{w} \cdot \bar{x}}$$
Assume $x_i$ is largest among $x_i$ all others ($i > 1$): $x_i \leq x_i, \sigma$

If $\sigma = 99\%$ then next best has $99\%$ of affinity of best

$$W_1 = \frac{W_i x_i}{\sum W_i x_i + W_2 x_2 + \ldots + W_n x_n} \leq x_i, \sigma$$

$$\geq \frac{W_i x_i}{W_i x_i + (1 - W_i) \sigma x_i}$$

$$= \frac{W_i}{W_i + (1 - W_i) \sigma}$$
> w[1]:=0.1; delta:=0.99; T:=1000;

\[ w_1 := .1 \]
\[ \delta := .99 \]
\[ T := 1000 \]

> for t from 1 to T-1 do
> w[t+1]:=w[t]/(w[t]+(1-w[t])*delta) ; od;
> plot([[n,w[n]] $n=1..T]);
\[ \delta = 0.99 \\
\eta_i = 0.01 \]

\[ \delta = 0.45 \\
\eta_i = 0.0001 \]
Find set of RNA strand that attaches to a number of different proteins

\[
P_1 \quad P_2 \quad P_3 \quad P_4 \quad P_5
\]

\[
\overline{x}_1 \quad \overline{x}_2 \quad \overline{x}_3 \quad \overline{x}_4 \quad \overline{x}_5
\]

In trial \( t \) functional separation based on \( \overline{x}_t \)

So far all \( \overline{x}_t \) the same
| $x_{E,1}$ | 0.9 | 0.8 | 0.96 | 0.2 | 0.04 |
| $x_{E,2}$ | 0.1 | 0.1 | 0.8  | 0.9 | 0.8  |
| $\mu \cdot x$ | 0.5 | 0.405 | 0.38 | 0.55 | 0.42 |

**Goal Tube:**

$$u = (0.5, 0.5, 0, \ldots, 0)$$

The two strands cooperatively solve the problem.
PROBLEM:

START WITH 1 LITER TUBE IN WHICH ALL STRANDS APPEAR WITH FREQUENCY
\( \approx 10^{-20} \)

GOAL: USE PCR AND TO ARRIVE AT TUBE

\( (\approx 0.5, \approx 0.5, \approx 0, \ldots, \approx 0) \)

PROBLEM:

IF YOU OVERTRAIN WITH \( p_1 \) AND \( p_2 \)
THEN \( w_1 \approx 0, w_2 \approx 0 \)

\( \begin{cases} \text{MULTIPLEX UPDATE TOO GOOD} & \text{A CURSE!} \\ \text{UPDATE TOO GOOD} & \text{A CURSE!} \end{cases} \)

IF YOU OVERTRAIN WITH \( p_4 \) AND \( p_5 \)
THEN \( w_1 \approx 0, w_2 \approx 1 \)

WANT BLIND COMPUTATION.

\( x_t, w_t \text{ UNKNOWN} \)

NEED SOME KIND OF FEEDBACK IN EACH TRIAL.
\textbf{FIX: IF GOOD SIDE REASONABLY LARGE}
\textbf{THEN DON'T DO PCR}

\textbf{IF } \overline{w_t} \cdot \overline{x_t} \geq \delta \\
\textbf{THEN } \overline{w_{th}} := \overline{w_t} \\
\textbf{RECOVER ALL RNA}

\textbf{ELSE } \overline{w_{th,i}} := \frac{w_{ti} \cdot x_{ti}}{\overline{w_t} \cdot \overline{x_t}} \\
\textbf{AMPLIFY GOOD RNA TO UNIT VOLUME}

\textbf{HOW EXACTLY CAN WE MEASURE } \overline{w} \cdot \overline{x} \textbf{?}

\textbf{SOME RNA MIGHT BE INACTIVE?}

\textbf{ALTERNATE}

$$\overline{w} \cdot \overline{x} \geq \alpha (1 - \overline{w} \cdot \overline{x})$$
On-Line Learning

<table>
<thead>
<tr>
<th></th>
<th>$E_1$</th>
<th>$E_2$</th>
<th>$E_3$</th>
<th>$E_n$</th>
<th>prediction</th>
<th>true label</th>
<th>loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>day 1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>day 2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>day 3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>day $t$</td>
<td>$x_{t,1}$</td>
<td>$x_{t,2}$</td>
<td>$x_{t,3}$</td>
<td>$x_{t,n}$</td>
<td>$\tilde{y}_t$</td>
<td>$y_t$</td>
<td>$</td>
</tr>
</tbody>
</table>

Protocol of the Master Algorithm

For $t = 1$ To $T$ Do

Receive $x_t \in \{0, 1\}^n$

Predict $\tilde{y}_t \in \{0, 1\}$

Get label $y_t \in \{0, 1\}$

Incur loss $|y_t - \tilde{y}_t| \in \{0, 1\}$
In biology, RNA strands are experts on each day, different protein $\bar{x}_t$.

Goal is to classify proteins so that # of wrong classifications minimized.

Also we might have set of labeled proteins

$$(\bar{x}_1, y_1), (\bar{x}_2, y_2), \ldots, (\bar{x}_T, y_T)$$

Training data

Goal: classify unlabeled proteins well

$\bar{x}_{T+1}, \bar{x}_{T+2}, \ldots$
Halving Algorithm

- Predicts with majority

- If mistake then number of consistent experts is halved
A run of the Halving Algorithm

<table>
<thead>
<tr>
<th>$E_1$</th>
<th>$E_2$</th>
<th>$E_3$</th>
<th>$E_4$</th>
<th>$E_5$</th>
<th>$E_6$</th>
<th>$E_7$</th>
<th>$E_8$</th>
<th>majority</th>
<th>true label</th>
<th>loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>0</td>
<td>1</td>
<td>x</td>
<td>x</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>1</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

consistent

For any sequence with a consistent expert, HA makes $\leq \log_2 n$ mistakes
What if no expert is consistent?

For any sequence \( S = (x_1, y_1), (x_2, y_2), \ldots, (x_T, y_T) \)
- \( L_A(S) \) is total loss of alg. A and
- \( L_i(S) \) is the total loss of expert \( E_i \)

Want bounds of the form:

\[ \forall S : \ L_A(S) \leq a \ \min_i L_i(S) + b \ \log(n) \]

where \( a, b \) are constants

Bounds loss of algorithm
relative to
loss of best expert
Can't wipe out experts!
One weight per expert

**Weighted Majority Algorithm**

- Predicts with larger side
- Weights of wrong experts are multiplied by $\beta \in (0, 1]$
Number of mistakes of the WM algorithm

\[ M_{t,i} = \text{\# of mistakes of } E_i \text{ before trial } t \]
\[ w_{t,i} = \beta^{M_{t,i}} \text{ weight of } E_i \text{ at trial } t \]
\[ W_t = \sum_{i=1}^{n} w_{t,i} \text{ total weight at trial } t \]

Minority \( \leq \frac{1}{2} W_t \)
Majority \( \geq \frac{1}{2} W_t \)

If no mistake then
minority multiplied by \( \beta \)
\[ W_{t+1} \leq 1 \cdot W_t \]
If mistake then
majority multiplied by $\beta$

$$W_{t+1} \leq \frac{1}{2} W_t + \beta \left( \frac{1}{2} W_t \right)_{\text{majority}}$$

$$= \frac{1 + \beta}{2} W_t$$

$$W_{T+1}^{\text{total final weight}} \leq \left( \frac{1 + \beta}{2} \right)^{M} W_1$$

$$W_{T+1} = \sum_{j=1}^{n} w_{T+1,j} = \sum_{j=1}^{n} \beta^{M_j} \geq \beta^{M_i}$$

$$\left( \frac{1 + \beta}{2} \right)^{M} \frac{W_1}{n} \geq \beta^{M_i}$$
\[ M \leq \frac{\ln \beta}{\ln \frac{2}{1+\beta}} M_i + \frac{1}{\ln \frac{2}{1+\beta}} \ln n \]

\[ M \leq \beta = \frac{2.63}{1/e} \min_i M_i + 2.63 \ln n \]

For all sequences, loss of the master algorithm is comparable to the loss of the best expert.

Relative loss bounds [F]
Learning Disjunctions

<table>
<thead>
<tr>
<th></th>
<th>(d_1)</th>
<th>(d_2)</th>
<th>(d_3)</th>
<th>(d_4)</th>
<th>(true) label</th>
<th>(d_1 \lor d_3)</th>
<th>(d_3 \lor d_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_{t,1})</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>(x_{t,2})</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>(x_{t,3})</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>(x_{t,4})</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Variable \(d_1 \lor d_3\) becomes \(v = (\frac{1}{2}, 0, \frac{1}{2}, 0)\)

\(d_1 \lor d_3\) is one on \(x_t\) iff \(x_t \cdot v \geq \frac{1}{2}\)
Want to do as well as best $k$ out of $n$ literal (monotone) disjunction

Use one expert per disjunction: $\binom{n}{k}$ experts

WM Algorithm makes at most

$$2.63 M + 2.63 k \ln \frac{n_k}{k}$$

mistakes where $M$ is number of mistakes of best disjunction of size $k$

Exponential in $k$
ONE EXPERT PER $k$-LITERAL DISJ
CORRESPONDS TO MAKING ONE MOLECULE
OUT OF EVERY SET OF $k$ RNA-STRANDS

FUNCTIONAL SEPARATION BASED ON COMMON PERFORMANCE

PROBLEMS:
COMBINATORIAL BLOW UP
DURING FUNCTIONAL SEPARATION
SPLIT INTO ITS $k$ PIECES
HOW TO PUT SAME $k$ BACK TOGETHER
Normalized Winnow

(n normalized weights)

At trial \( t \):

\[
\text{If } x_t \cdot v_t \geq \frac{1}{2k} \text{ then predict } 1 \\
\text{else predict } 0
\]

If mistake then multiply wrong side by \( \beta \) and renormalize.

The number of mistakes is at most

\[
3.71 A + 3.71 k \ln \frac{n}{k}
\]

where \( A \) is number of attribute errors of best disjunction of size \( k \).

Number of attribute errors of a disjunction w.r.t. the sequence is minimum number of attributes that need to be changed so that the disjunction is consistent.

\[ A \leq k M \]
DON'T KNOW HOW TO LEARN

1) DISJUNCTIONS WITHOUT THRESHOLDING
   IF ONE STRAND TOO PLENTIFULL THEN THIS SHOULD INHIBIT PCR

2) DON'T KNOW HOW TO LEARN CONJUNCTIONS WITHOUT CELL WALLS
3 STRANDS OF RNA
WANT TO KEEP PERCENTAGES

\[
\begin{array}{|c|c|}
\hline
A & C \\
25\% & 15\% \\
\hline
B & \text{not represented} \\
60\% & \text{not represented} \\
\hline
\end{array}
\]

ITERATIVE PCR WILL FAVOR ONE

SOLUTION:
ONE STRAND WITH RIGHT PERCENTAGES
OF A, B, C

A B B B C A B B

DUPLICATE SAME STRAND
CHOP INTO CONSTITUENTS WHEN NEEDED
UN-COUPLED

A & B COMPETE

COUPLED

A & B SHOULD COOPERATE