Machine Learning Coms-4771

Online learning: Weighted Majority and Perceptron

Lecture 8

Recap: Predicting from Expert Advice

Online Learning Model: View learning as a sequence of trials:

- N experts give their advice
- Learner makes its prediction
- True outcome is revealed

Can we do nearly as well as the best expert in hindsight?

Weighted Majority Algorithm:

- Start with all experts having weight 1: $w_1 = w_2 = \ldots = w_N = 1$
- Predict based on weighted majority vote: Output 1 if $\sum_{i:x_i=1} w_i \geq \sum_{i:x_i=0} w_i$, otherwise output 0.
- Penalize mistakes by cutting weight in half. If expert i made a mistake, set $w_i \leftarrow w_i/2$.

M = number of mistakes made by the algorithm, m = number of mistakes of the best expert so far

Theorem: $M \leq 2.4(m + \log N)$

Randomized Weighted Majority Algorithm

Parameter $\epsilon \in (0, 1)$.

- Start with all experts having weight 1: $w_1 = w_2 = \ldots = w_N = 1$
- Output expert i-th prediction with probability w_i/W, where $W = \sum_{i=1}^{N} w_i$ is the total weight (i.e., expert i is selected with probability proportional to w_i).
- Update weights: For each expert i who made a mistake, set $w_i \leftarrow (1 - \epsilon)w_i$.

Algorithm in Action $\epsilon = 1/2$:

Experts	E_1	E_2	E_3	E_4	E_5	E_6	prediction	outcome
Weights	1	1	1	1	1	1		
Advice	1	1	0	0	0	0	0 $(\frac{2}{3} : \frac{1}{3})$	1
Weights	1	1	1/2	1/2	1/2	1/2		
Advice	0	1	1	1	1	0	1 $(\frac{3}{8} : \frac{5}{8})$	0
Weights	1	1/2	1/4	1/4	1/4	1/2		

The larger the probability of a mistake, the larger the amount by which the weight is reduced.

Randomized Weighted Majority Analysis

Theorem: For any $\epsilon \in (0, 1/2]$, on any sequence of trials,

$$M \leq (1+\epsilon)m + \frac{\ln N}{\epsilon}$$

where M is the *expected* number of mistakes made by the algorithm, m is the number of mistakes made by the best expert so far.

Proof:

- F_t = fraction of the total weight on the *wrong* answers in trial t = probability that the algorithm makes a mistake in trial t. The expected number of mistakes so far $M = \sum_{t=1}^{T} F_t$.
- After trial t, the total weight W drops by a factor of $(1 - F_t\epsilon)$ (since F_t fraction made a mistake and these decrease their weight by ϵ).
- Since W is at least as large as the weight of the best expert so far, $W \geq (1-\epsilon)^m$.
- Since initially $W = N$, after T trials we have

$$N \prod_{t=1}^{T} (1 - F_t\epsilon) \geq (1-\epsilon)^m$$

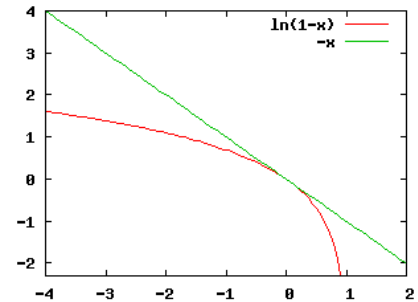
Randomized Weighted Majority Analysis (continued)

Taking logs

$$\ln(N) + \sum_{t=1}^{T} \ln(1 - F_t\epsilon) \geq m \ln(1 - \epsilon)$$

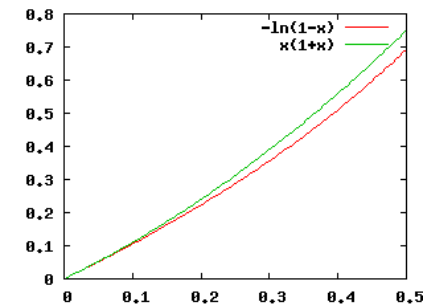
Since $-x \geq \ln(1 - x)$, we have

$$\ln(N) + \sum_{t=1}^{T}(-F_t\epsilon) = \ln(N) - \epsilon M \geq m \ln(1 - \epsilon)$$



Rearranging

$$M \leq \frac{-\ln(1 - \epsilon)}{\epsilon} m + \frac{\ln(N)}{\epsilon}$$



The theorem follows from the fact that $-\ln(1 - \epsilon) \leq \epsilon(1 + \epsilon)$ for $\epsilon \in [0, 1/2]$.

How do we choose ϵ?

- There is a tradeoff (using the slightly better bound at the end of the proof):

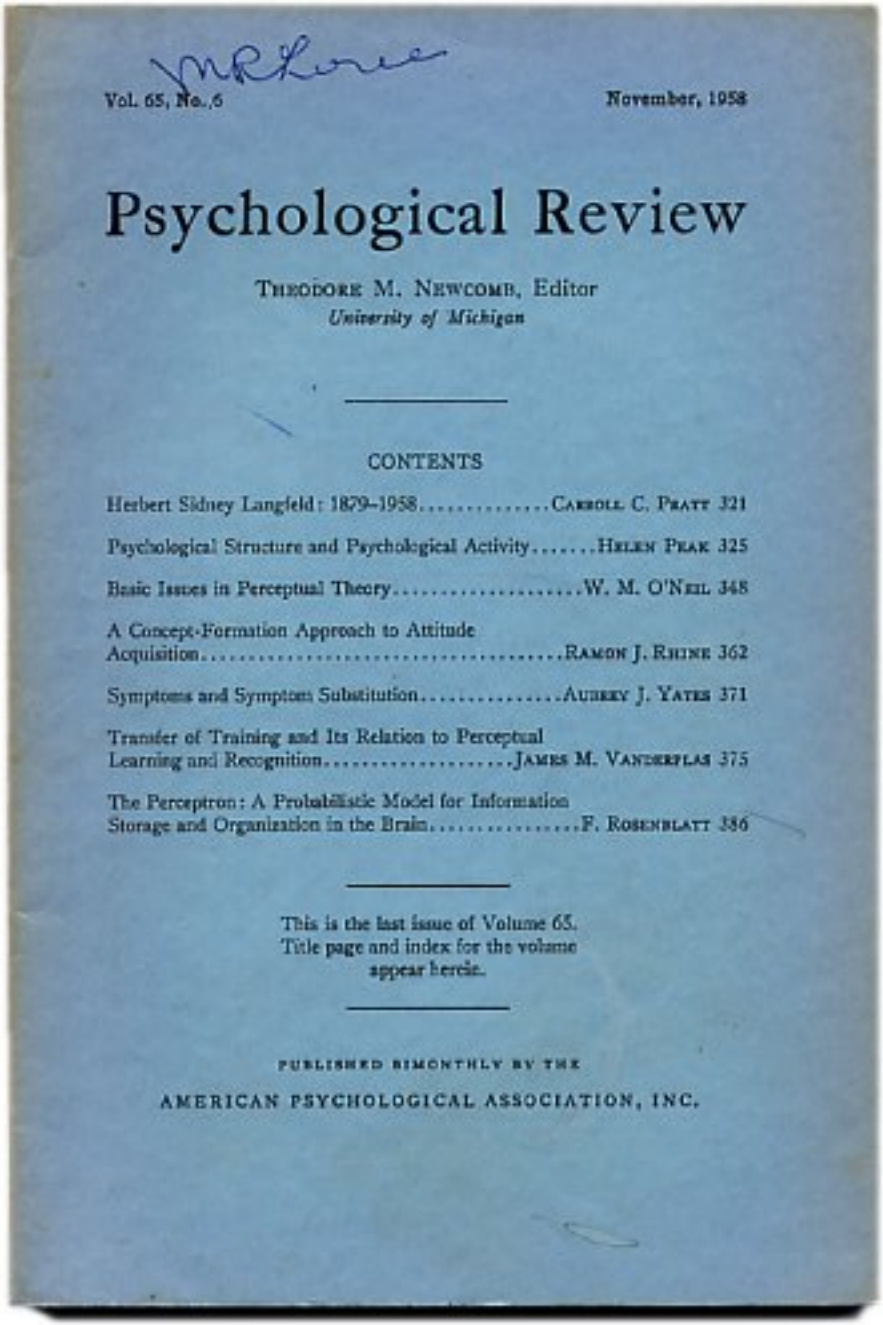
ϵ	M
$1/2$	$1.39m + 2\ln N$
$1/4$	$\underbrace{1.15}_{\text{competitive ratio}}m + 4\ln N$

  By adjusting ϵ, we can make the ratio close to 1 at the expense of the additive constant (second term).

- For a given m, the best setting of ϵ in the bound is $\ln(N)/m$, giving the bound $M \leq m + 2\sqrt{m\ln(N)}$. (Taking the derivative of the bound in the theorem statement and setting it to 0, $m = \ln(N)\epsilon^2$)

- Guess and doubling trick: If we don't know m, start with $m = 4\ln N$ and $\epsilon = 1/2$. Once every expert has made at least $4\ln N$ mistakes, restart with $m = 8\ln N$ (and $\epsilon = 1/2\sqrt{2}$).

Perceptron Algorithm



Vol. 65, No. 6 November, 1958

Psychological Review

THEODORE M. NEWCOMB, Editor
University of Michigan

CONTENTS

This is the last issue of Volume 65.
Title page and index for the volume appear herein.

PUBLISHED BIMONTHLY BY THE
AMERICAN PSYCHOLOGICAL ASSOCIATION, INC.

Frank Rosenblatt. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain (Psychological Review, 1958).

Thousands of citations

(The original cover can be had for just $2300 around 2nd Ave and 55th Street ;)

Perceptron

Winnow can learn linear threshold functions for $\{0,1\}^n$. Perceptron learns a linear threshold function $f : \mathbb{R}^n \to \{0,1\}$ of the form

$$f(\mathbf{x}) = \mathbf{1}(\mathbf{w} \cdot \mathbf{x} \geq \theta),$$

for $\mathbf{w} \in \mathbb{R}^n$, $\theta \in \mathbb{R}$.

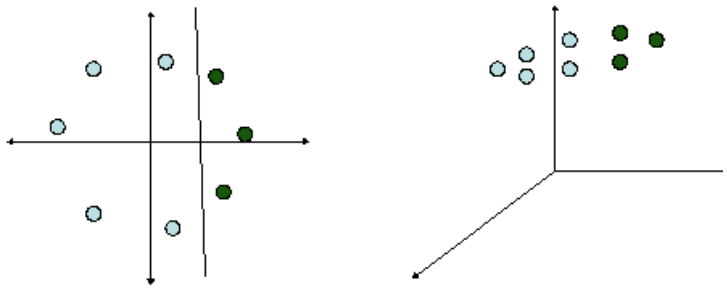
Geometrically, $f(x)$ defines a hyperplane separating $\mathbb{R}^n$ into two halfspaces.

First observation: θ can be made 0 by adding a dummy variable to $\mathbf{x}$ that is always 1:

$$\mathbf{1}(\sum_{i=1}^{n} w_i x_i \geq \theta) = \mathbf{1}(\sum_{i=0}^{n} w_i x_i \geq 0)$$

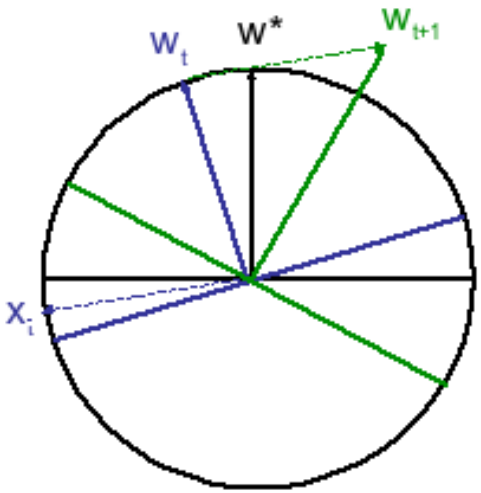
for $w_0 = -\theta$ and $x_0 = 1$.

So it's enough to find a hyperplane going through the origin.



Perceptron Algorithm

Sequence of labeled examples
$(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m) \in \mathbb{R}^n \times \{0, 1\}$



Start with $\mathbf{w}_1 = \mathbf{0}$ (the all-zeros vector), set $t = 1$. For each i from 1 to m:

- Given example $\mathbf{x}_i$, predict positive iff $\mathbf{w}_t \cdot \mathbf{x}_i > 0$.
- On a mistake on positive, update: $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \mathbf{x}_i$, increment t.
- On a mistake on negative, update: $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \mathbf{x}_i$, increment t.

Scale all examples $\mathbf{x}_i$ so that $\|\mathbf{x}_i\| = 1$. Doesn't affect which side of the plane they are on.

Intuitively right: $\mathbf{w}_{t+1} \cdot \mathbf{x}_i = (\mathbf{w}_t + \mathbf{x}_i) \cdot \mathbf{x}_i = \mathbf{w}_t \cdot \mathbf{x}_i + 1$ (similarly for negatives), so we are moving in the right direction (by 1).

Theorem For any sequence consistent with a linear threshold function $\mathbf{w}^* \cdot \mathbf{x} > 0$, where $\|\mathbf{w}^*\| = 1$, the number of mistakes M made by the online Perceptron algorithm is at most $1/\gamma^2$, where

$$\gamma = \min_{\mathbf{x}_i} \mid \mathbf{w}^* \cdot \mathbf{x}_i \mid,$$

the min distance of any example to the plane $\mathbf{w}^* \cdot \mathbf{x} = 0$ (called the *margin* of $\mathbf{w}^*$). (Recall that all $\|\mathbf{x}_i\| = 1$.)

Claim 1: Every time we make a mistake $\mathbf{w}_t \cdot \mathbf{w}^*$ goes up by at least γ.
If $\mathbf{x}_i$ is positive, then we get
$\mathbf{w}_{t+1} \cdot \mathbf{w}^* = (\mathbf{w}_t + \mathbf{x}_i) \cdot \mathbf{w}^* = \mathbf{w}_t \cdot \mathbf{w}^* + \mathbf{x}_i \cdot \mathbf{w}^* \geq \mathbf{w}_t \cdot \mathbf{w}^* + \gamma$, by definition of γ.
Similarly for negative $\mathbf{x}_i$, we get $(\mathbf{w}_t - \mathbf{x}_i) \cdot \mathbf{w}^* = \mathbf{w}_t \cdot \mathbf{w}^* - \mathbf{x}_i \cdot \mathbf{w}^* \geq \mathbf{w}_t \cdot \mathbf{w}^* + \gamma$.
So, after M mistakes $\mathbf{w}_{M+1} \cdot \mathbf{w}^* \geq \gamma M$.

Claim 2: Every time we make a mistake, $\|\mathbf{w}_t\|^2$ goes up by at most 1.
If $\mathbf{x}_i$ was positive, we get $\|\mathbf{w}_t + \mathbf{x}_i\|^2 = \|\mathbf{w}_t\|^2 + 2\mathbf{w}_t \cdot \mathbf{x}_i + \|\mathbf{x}_i\|^2 \leq \|\mathbf{w}_t\|^2 + 1$.
The last inequality is due to the fact that $\mathbf{w}_t \cdot \mathbf{x}_i$ was negative (since we made a mistake on $\mathbf{x}_i$). Similarly for negatives. So after M mistakes, $\|\mathbf{w}_{M+1}\| \leq \sqrt{M}$.

Now $\mathbf{w}_t \cdot \mathbf{w}^* = \|\mathbf{w}_t\| \cos(\mathbf{w}_t, \mathbf{w}^*) \leq \|\mathbf{w}_t\|$ (since $\cos(\mathbf{w}_t, \mathbf{w}^*) \leq 1$). So $\gamma M \leq \sqrt{M}$, and $M \leq 1/\gamma^2$.

- If data is separable by a large margin, then Perceptron is a good algorithm to use.
- If there is no perfect separator or only most data is separable by a large margin: Can bound the total number of mistakes we make in terms of the total distance TD_γ we have to move points to make them separable by margin γ:

  $$M \leq 1/\gamma^2 + (2/\gamma)\text{TD}_\gamma$$

  (can't say that we are making only a small multiple of the number of mistakes made by $\mathbf{w}^*$, but we are doing well in terms of TD_γ)
- What if our data doesn't have a good linear separator? Kernels (Sanjoy's lectures in a couple of weeks)