Lecture 1: 242
Intro to Machine Learning W08
By Manfred K. Warmuth

Class Web Page:
http://www.soe.ucsc.edu/classes/cmps242
Winter 08

My Web Page:
Google "manfred"

---

What is Machine Learning?
- Learning/Inference from Data via Machine
- Very powerful
- 1984 did not happen because we could not do it
- Now we can!!!

---

Privacy, Who is in control?
Tracking ethical issues everywhere

- Machine learning is the "atom bomb" of computer science
- Imagine a rogue government in charge of power
MACHINE LEARNING

- PERVERSIVE

- POWERFUL TOOL THAT CAN BE USED EITHER WAY
  - WILL IT EMPOWER THE INDIVIDUAL?
    - LAPTOP, CELL PHONE, OPEN SOURCE
    - OR THE COOPERATIONS / GOVERNMENTS?
    - CENTRALIZED SEARCH ENGINES

IN THIS CLASS:
LEARNING FROM EXAMPLES

\((\overline{X}_m, y_m)\)

\(\mathcal{A}\) LABEL: REAL OR BINARY

INSTANCE:

- VECTOR OF FEATURES / ATTRIBUTES
- REAL OR BINARY

REAL LABELS: REGRESSION
BINARY: CLASSIFICATION

BATCH MODEL
- ALL DATA AT ONCE

ON-LINE MODEL
- DATA RECEIVED ON-LINE
Plot of a training data set of \( N = 10 \) points, shown as blue circles, each comprising an observation of the input variable \( x \) along with the corresponding target variable \( y \). The green curve shows the function \( \sin(2\pi x) \) used to generate the data. Our goal is to predict the value of \( y \) for some new value of \( x \), without knowledge of the green curve.

\[
y(x) = \sin(2\pi x) + \text{noise}
\]

**Input Dimension:** 1

**Target Function Not Known to Learner**

**Learner Only Gets Data**
\[ \hat{y}(x, \overline{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M \]

\( \overline{w} \): PARAMETER VEC.

\[ L(\overline{w}) = \frac{1}{2} \sum_{n=1}^{N} (\hat{y}(x_n, \overline{w}) - y_n)^2 \]

SQUARE LOSS ERROR

**LOSS**

The error function (1.2) corresponds to (one half of) the sum of the squares of the displacements (shown by the vertical green bars) of each data point from the function \( y(x, w) \).

WHAT IS A GOOD PARAMETERIZE FUNCTION CLASS?  
WHAT ARE GOOD LOSS FUNCTIONS?
1.1. Example: Polynomial Curve Fitting

Which fit is better?

Which prediction is better?
Graphs of the root-mean-square error, defined by (1.3), evaluated on the training set and on an independent test set for various values of $M$.

$$E_{RME} = \sqrt{\frac{2L(w)}{N}}$$

Root Mean Squared Error / Loss

LARGE $M$: Training Error ↓

TEST ↓
Figure 1.6  Plots of the solutions obtained by minimizing the sum-of-squares error function using the $M = \text{polynomial}$ for $N = 15$ data points (left plot) and $N = 100$ data points (right plot). We see that increasing the size of the data set reduces the over-fitting problem.

**Fix:**

**Modify Loss Function**

$$L(w) = \frac{1}{2} \sum_{n} (g(x_n, w) - y_n)^2 + \frac{\lambda}{2} \|w\|^2$$

$$\|w\|^2 = w^T w = \sum_{m} w_m^2$$

"Long" weight vectors
Fitted data with $M = 9$

No regularization:

$\lambda = 0$

$\ln \lambda = -\infty$

Small $\lambda$: overfitting

Large $\lambda$: bad fit
TRAIN

TEST

CAN'T CHOOSE >
BASED ON TEST DATA

TEST DATA JUST FOR
REPORTING RESULTS
(CAN ONLY BE USED ONCE)

TRAINING SET

VALIDATION SET

TEST SET

LESS FOR TRAINING

USE THIS TO
CHOOSE >

REPORT RESULTS

- WASTEFULL

- AVERAGE OVER MANY SPLITS
  - AVERAGE OVER ALL CHOICES OF SINGLE VALIDATION PT
WHAT YOU WILL LEARN IN CLASS

ALL BASIC ML ALGS:

- LINEAR & LOGISTIC REGRESSION
  \[ \text{SQUARE LOSS} \]
  \[ M = 1 \]

- BAYESIAN METHODS

- SUPPORT VECTOR MACHINES FOR CLASSIFICATION

- KERNEL METHODS

  \[(x_1, x_2, \ldots, x_M) \rightarrow (x_i x_1, x_i x_2, \ldots, x_i x_M)\]

- BOOSTING
  COMBINING MANY WEAK CLASSIFIERS

- ON-LINE LEARNING
ASSUMPTION OF BATCH MODEL

- DATA IS STATIONARY

- IN MANY PRACTICAL SETTINGS
  DATA IS CONTINUOUSLY CHANGING

ONLINE-LEARNING

LOOP

- GET NEXT DT
- PREDICT BASED ON CURRENT MODEL
- UPDATE MODEL

TENSION 1 - HOW MUCH SHOULD LEARNER LISTEN TO RECENT VS. PAST DATA

BEFORE 1 - HOW COMPLEX A MODEL?
- HOW MUCH REGULARISATION?
BATCH:
- TRAINING AND TEST DATA GENERATED BY SAME DISTRIBUTION
- IF MODEL CLASS NOT TOO COMPLEX AND ENOUGH EXAMPLES
  MODEL THAT DOES BEST ON TRAINING DATA NOT TOO MUCH WORSE ON TEST DATA

ON-LINE:
- ALL IS IN FLUX
- NO STATISTICAL ASSUMPTIONS
- STILL CAN BOUND "REGENCY"

TOTAL LOSS OF ON-LINE - TOTAL LOSS OF BEST OFF-LINE CHOSEN IN HIND SIGHT

- BOUNDS HOLD FOR ARBITRARY SEQUENCES OF EXAMPLES
WHAT YOU WILL LEARN

- TECHNIQUES FOR DERIVING & ANALYSING
  ON-LINE LEARNING ALGS
    • BREGMAN DIVERGENCES
    • BREGMAN PROJECTIONS
  • HOW TO PROVE REGRET BOUNDS
    OR RELATIVE LOSS BOUNDS

RECURRING THEME

- HOW TO COMBINE MANY RULES OF THUMB
  • EXPERT SETTING
  • BOOSTING
  • BUG MACHINE :-)  
    (END OF CLASS)
OUTLINE:

TODAY:
- EXPERT SETTING
- VARIOUS METHODS FOR PROVING
  RELATIVE LOSS BOUNDS

LECTURE 2:
- APPLICATIONS
  - DISK SPIN DOWN
  - CACHING
- HOW TO MEASURE ON-LINENESS
- SHIFTING EXPERT SETTING
  - LONG TERM MEMORY
- HW1 (PRACTICAL)

LECTURE 3:
ON-LINE & BATCH ALG
FOR LINEAR & LOGISTIC REGRESSION
- DERIVATION OF UPDATES
- ANALYSIS
- BREGMAN DIVERGENCES

LECTURE 4:
HW1 DUE
On-Line Learning

<table>
<thead>
<tr>
<th>Experts</th>
<th>Prediction</th>
<th>True Label</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_1$</td>
<td>$E_2$</td>
<td>$E_3$</td>
<td>$E_n$</td>
</tr>
<tr>
<td>day 1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>day 2</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>
</tr>
<tr>
<td>day $t$</td>
<td>$x_{t,1}$</td>
<td>$x_{t,2}$</td>
<td>$x_{t,3}$</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 $|y_t - \tilde{y}_t| \in \{0, 1\}$
CASE 1: THERE IS A CONSISTENT EXPERT

GIVEN SEQUENCE \((x_t, y_t)\) s.t

\[x_{t,i} = y_t \text{ for all } t\]

LOSS OF OFF-LINE COMPARATOR IS ZERO

NOISE-FREE CASE
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>1</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>x</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>1</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>consistent</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For any sequence with a consistent expert, HA makes $\leq \log_2 n$ mistakes

---

GAME AGAINST NATURE (ADVERSARY)

WHICH Chooses the $x_t$ & $y_t$

IF THERE IS ONE CONSISTENT EXPERT

THEN Alg. $\leq \log_2 n$ MISTAKES
Case 2:

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 $E_i$

**Relative Loss**

Want bounds of the form:

$$\forall S : \quad L_A(S) \leq a \min_i L_i(S) + b \log(n)$$

where $a, b$ are constants

Bounds loss of algorithm
relative to
loss of best expert

$$a = 1$$

$$L_A(S) - \min_i L_i(S) \quad \text{CALLED REGRET}$$
Can't wipe out experts!
One weight per expert

Weighted Majority Algorithm

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

- $\beta$ is fitness factor
- HA : $\beta = 0$
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 \( \beta \)
\[ W_{t+1} \leq 1 W_t \]
If mistake then
majority multiplied by $\beta$

$$W_{t+1} \leq W_t + \beta \cdot \frac{1}{2} W_t$$

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

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

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

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

\[ M \leq \frac{2.63 \min_i M_i}{e^{\frac{M^y}{a}}} + \frac{2.63 \ln n}{b} \]

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

Relative loss bounds

\[ M \leq 2M^x + 2\sqrt{M^x \ln (N)} + 1 \log_2 n \]

With fancy choice of \( \beta \) that depends on \( u_i, M^y \):

\( \uparrow \) Necessary for deterministic prediction
SUMMARY OF ANALYSIS METHOD

\[ w_{t+1,i} = \frac{1}{w_{t+1,i} \beta^{M_{t+1,i}}} \text{ UNNORMALIZED WEIGHTS} \]

UNNORMALIZED POTENTIAL:

\[ p_{t+1} = -\sum_{i}^{\text{M}_{t+1}} \beta^{M_{t+1,i}} \]

\[ \frac{p_{t+1}}{p_{t}} = \begin{cases} \frac{1+\beta}{2} & \text{if mistake in trial } t \\ \geq 1 & \text{if no mistake} \end{cases} \]