Online Learning
Perceptron Algorithm

Machine Learning – CSE446
Carlos Guestrin
University of Washington

April 29, 2013

©Carlos Guestrin 2005-2013

Challenge 1: Complexity of Computing Gradients

\[ w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \left\{ -\lambda w_i^{(t)} + \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid x^j, w)] \right\} \]
Challenge 2: Data is streaming

- Assumption thus far: Batch data

- But, e.g., in click prediction for ads is a streaming data task:
  - User enters query, and ad must be selected:
    - Observe $x_i$, and must predict $y_i$
  - User either clicks or doesn’t click on ad:
    - Label $y_i$ is revealed afterwards
      - Google gets a reward if user clicks on ad
  - Weights must be updated for next time:

Online Learning Problem

- At each time step $t$:
  - Observe features of data point:
    - Note: many assumptions are possible, e.g., data is iid, data is adversarially chosen… details beyond scope of course
  - Make a prediction:
    - Note: many models are possible, we focus on linear models
    - For simplicity, use vector notation
  - Observe true label:
    - Note: other observation models are possible, e.g., we don’t observe the label directly, but only a noisy version… Details beyond scope of course
  - Update model:
The Perceptron Algorithm

Classification setting: $y \in \{-1, +1\}$
Linear model
- Prediction:

Training:
- Initialize weight vector:
- At each time step:
  - Observe features:
  - Make prediction:
  - Observe true class:
- Update model:
  - If prediction is not equal to truth

Fundamental Practical Problem for All Online Learning Methods: **Which weight vector to report?**

- Suppose you run online learning method and want to sell your learned weight vector… Which one do you sell???
- Last one?
Choice can make a huge difference!!

![Graph showing mistake bounds](image)

[Freund & Schapire ’99]

Mistake Bounds

- Algorithm “pays” every time it makes a mistake:
  - How many mistakes is it going to make?
Linear Separability: More formally, Using Margin

- Data linearly separable, if there exists
  - a vector
  - a margin
- Such that

Perceptron Analysis: Linearly Separable Case

- Theorem [Block, Novikoff]:
  - Given a sequence of labeled examples:
    - Each feature vector has bounded norm:
      - If dataset is linearly separable:
        - Then the number of mistakes made by the online perceptron on this sequence is bounded by
Perceptron Proof for Linearly Separable case

Every time we make a mistake, we get gamma closer to $w^*$:
- Mistake at time $t$: $w(t+1) = w(t) + y(t)x(t)$
- Taking dot product with $w^*$:
- Thus after $m$ mistakes:

Similarly, norm of $w(t+1)$ doesn’t grow too fast:
- $||w(t+1)||^2 = ||w(t)||^2 + 2y(t)(w(t) \cdot x(t)) + ||x(t)||^2$
- Thus, after $m$ mistakes:

Putting all together:

Beyond Linearly Separable Case

Perceptron algorithm is super cool!
- No assumption about data distribution!
  - Could be generated by an oblivious adversary,
    no need to be iid
- Makes a fixed number of mistakes, and it’s done for ever!
  - Even if you see infinite data

However, real world not linearly separable
- Can’t expect never to make mistakes again
- Analysis extends to non-linearly separable case
- Very similar bound, see Freund & Schapire
- Converges, but ultimately may not give good accuracy (make many many many mistakes)
What you need to know

- Notion of online learning
- Perceptron algorithm
- Mistake bounds and proof
- In online learning, report averaged weights at the end