Classification

Computational Linguistics: Jordan Boyd-Graber
University of Maryland
PERCEPTRON

Slides adapted from Mohri

Motivation

- On-line learning:
  - update parameters with each example
  - no distributional assumption.
  - worst-case analysis (adversarial).
  - mixed training and test.
  - Performance measure: mistake model, regret.

General Online Setting

- For $t = 1$ to T:
  - Get instance $x_t \in X$
  - Predict $\hat{y}_t \in Y$
  - Get true label $y_t \in Y$
  - Incur loss $L(\hat{y}_t, y_t)$
- Classification: $Y = \{0, 1\}$, $L(y, y') = |y' - y|$
- Regression: $Y \subset \mathbb{R}, L(y, y') = (y' - y)^2$

General Online Setting

- For $t = 1$ to T:
  - Get instance $x_t \in X$
  - Predict $\hat{y}_t \in Y$
  - Get true label $y_t \in Y$
  - Incur loss $L(\hat{y}_t, y_t)$
- Classification: $Y = \{0, 1\}$, $L(y, y') = |y' - y|$
- Regression: $Y \subset \mathbb{R}, L(y, y') = (y' - y)^2$
- **Objective**: Minimize total loss $\sum_t L(\hat{y}_t, y_t)$

Perceptron Algorithm

- Online algorithm for classification
- Very similar to logistic regression (but 0/1 loss)
- But what can we prove?

Perceptron Algorithm

$\vec{w}_1 \leftarrow \vec{0}$;
for $t \leftarrow 1 \dots T$ **do**
  Receive x_t;
  $\hat{y}_t \leftarrow \text{sgn}(\vec{w}_t \cdot \vec{x}_t)$;
  Receive y_t;
  if $\hat{y}_t \neq y_t$ **then**
    $\vec{w}_{t+1} \leftarrow \vec{w}_t + y_t \vec{x}_t$;
  else
    $\vec{w}_{t+1} \leftarrow w_t$;
return w_{T+1}

Algorithm 1: Perceptron Algorithm (Rosenblatt, 1958)

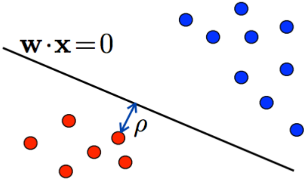
Objective Function

- Optimizes

$$\frac{1}{T}\sum_t \max\left(0, -y_t(\vec{w} \cdot x_t)\right) \tag{1}$$

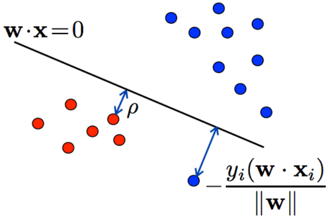
- Convex but not differentiable

Margin and Errors



- If there's a good margin ρ, you'll converge quickly

Margin and Errors



- If there's a good margin ρ, you'll converge quickly
- Whenever you se an error, you move the classifier to get it right
- Convergence only possible if data are separable

How many errors does Perceptron make?

- If your data are in a R ball and there is a margin

$$\rho \leq \frac{y_t(\vec{v} \cdot \vec{x}_t)}{||v||} \tag{2}$$

  for some $\vec{v}$, then the number of mistakes is bounded by R^2/ρ^2
- The places where you make an error are support vectors
- Convergence can be slow for small margins

Why study Perceptron?

- Simple algorithm
- Bound independent of dimension and tight
- Foundation of deep learning
- Proof techniques helped usher in SVMs
- Generalizes to structured prediction