Ensemble Methods

Scribe: James Rutledge

1 Notation for Ensemble Methods

- \( T \): Number of models/classifiers
- \( \mathbf{x} \): Test example
- \( y \): Test label (binary)
- \( X_t \): Bootstrap training dataset \( t \)
- \( h_t(x) \): Hypothesis about \( X \) from model \( t \)
- \( r \): Probability of error for each model
- \( R \): Number of votes for wrong class

2 Bagging (Bootstrap Aggregation)

**Idea:** Partition data into 3 sets: a training set, a validation set, and a testing set.
Create \( T \) models, using replacement, with \( n \) examples each. Since we are sampling with replacement, we will get very different data sets.

**Why \( n \) examples?**
- Probability that we didn’t choose a data point: \( (\frac{n-1}{n})^n \)
- Aside: \( \lim_{x \to \infty} (1 + \frac{x}{n})^n = e^x \)
- So, the probability that we didn’t choose a data point = \( e^{-1} \approx .3 \).

**Algorithm Idea:**

**Training** -
- for \( t = 1, 2, \ldots, T \):
  - create bootstrap sample \( X_t \)
  - train on model \( X_t \) to get model \( h_t(x) \)

**Testing** -
- for \( \mathbf{x} \) in testset:
  \( h_t(x) = \text{argmax} \sum_{t=1}^T \mathbb{1}(h_t(x) = y) \)

\( R = \sum_{t=1}^T \mathbb{1}(h_t(x) = \tilde{y}) \) //where \( \tilde{y} \) is the wrong class.

**Probability**

- \( \text{P}(R=k) = \binom{T}{k} r^k (1-r)^{T-k} \)
- \( r^k \) represents being wrong \( k \) times.
- \( (1-r)^{T-k} \) represents being right \( T-k \) times.
- Overall wrong = \( \text{P}(R > \frac{T}{2}) \)
  \( = \sum_{k=\frac{T}{2}+1}^T \binom{T}{k} r^k (1-r)^{T-k} \)

3 Random Forests

**Idea:** choose a different subset of features for every classifier \( t \).

**Goal:** decorrelate models
In practice: choose $\sqrt{p}$ features

- without replacement for each model
- results in every model having independent data points and independent features.

4 Boosting

Idea: try to learn from past mistakes. See where we got a lot of errors and update weights to reduce the number of errors.

**Training** -

- start with equal weights on all data points.
- for $T$ iterations:
  - learn classifier $t$, using weights
  - change example weights

**Testing** -

- get predictors for $T$ classifiers
- vote based on training performance