Slides modified from: MLSS’03: Gunnar Rätsch, 
Introduction to Boosting
http://www.boosting.org

AdaBoost: Introduction

• Classifiers
  • Supervised Classifiers
    • Linear Classifiers
      • Perceptron, Least Squares Methods
      • Linear SVM
    • Nonlinear Classifiers
      • Part I: Multi Layer Neural Networks
      • Part II: Pol. Class., RBF, Nonlinear SVM
    • Nonmetric Methods - Decision Trees
  • **AdaBoost**
    • Unsupervised Classifiers
AdaBoost: Agenda

- Idea AdaBoost
  - Combine many low-accuracy classifiers (weak learners) to create a high-accuracy classifier (strong learners)

---

AdaBoost: Introduction

- Example: 2 classes of apples

The World:

Data:
\[ \{(\mathbf{x}_n, y_n)\}_{n=1}^N, \quad \mathbf{x}_n \in \mathbb{R}^d, \quad y_n \in \{-1, 1\} \]

Unknown target function:
\[ y = f(x) \text{ (or } y \sim P(y|x)) \]

Unknown distribution:
\[ x \sim p(x) \]

Objective:
Given new \( x \), predict \( y \)

**Problem:** \( P(x, y) \) is unknown!
AdaBoost: Introduction

The Model:

- Hypothesis class: \( \mathcal{H} = \{ h : \mathbb{R}^d \rightarrow \{ \pm 1 \} \} \)
- Loss: \( l(y, h(x)) \) (e.g., \( I[y \neq h(x)] \))

- Objective: Minimize the true (expected) loss – ("generalization error")

\[ h^* = \arg \min_{h \in \mathcal{H}} L(h) \text{ with } L(h) = \mathbb{E}_{x,y} \left[ l(y, h(x)) \right] \]

- Problem: Only a data sample is available, \( P(x, y) \) is unknown!

- Solution: Find empirical minimizer \( \hat{h}_N = \min_{h \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} l(y_i, h(x_i)) \)

How can we efficiently construct complex hypotheses with small generalization errors?

---

AdaBoost: Framework

Algorithm

Idea:
- Simple Hypotheses are not perfect!
- Hypotheses combination \( \rightarrow \) increased accuracy

Problems:
- How to generate different hypotheses?
- How to combine them?

Method:
- Compute distribution \( d_1, \ldots, d_N \) on examples
- Find hypothesis on the weighted training sample \( (x_1, y_1, d_1), \ldots, (x_N, y_N, d_N) \)
- Combine hypotheses \( h_1, h_2, \ldots \) linearly:

\[ f = \sum_{i=1}^{T} \alpha_i h_i \]
AdaBoost: Frame work

Input: $N$ examples $\{(x_1, y_1), \ldots, (x_N, y_N)\}$.
$L$ a learning algorithm generating hypothesis $h(x)$ (classifiers)
$T$ maxNumber of hypotheses in the ensemble

Initialize: $d_n$ weight of example $n$ ($d$ is a distribution with $\sum_{n=1}^{N} d_n^{(t)} = 1$)

$$d_n^{(1)} = 1/N \text{ for all } n = 1,\ldots,N$$

Do for $t = 1,\ldots,T$,

1. Train base learner according to example distribution $d^{(t)}$ and obtain hypothesis $h_t : x \mapsto \{\pm 1\}$.
2. Compute weighted error $e_t = \sum_{n=1}^{N} d_n^{(t)} \mathbb{1}[y_n \neq h_t(x_n)]$
3. Compute hypothesis weight $a_t = \frac{1}{2} \ln \frac{1-e_t}{e_t}$
4. Update example distribution $d_n^{(t+1)} = d_n^{(t)} \exp (-a_t y_n h_t(x_n)) / Z_t$

$Z_t$ is a normalization factor

Output: final hypothesis $f_{\text{AdaBoost}}(x) = \sum_{t=1}^{T} a_t h_t(x)$

AdaBoost: Decision Stumps

- A family of weak learners,

  e.g. Decision stump:
  - can perform a single test on a single attribute with threshold $\Theta$.
  - parameterize all decision stumps as follows:

  $$f^j(x; \theta) = \begin{cases} 
  1 & \text{if } x_j > \theta \\
  -1 & \text{else}
  \end{cases}, \quad j = 1,\ldots,d$$
AdaBoost: Example

• Example: natural apples vs. plastic apples

How to classify?

class A

class B

light

heavy

not red

red

AdaBoost

• Example: natural apples vs. plastic apples

1st hypothesis
Weak classifier (cuts on coordinate axes)
AdaBoost

• Example:

Recomputing weightings of the training patterns

AdaBoost

• Example:

2nd hypothesis
AdaBoost

• Example:

Recompute weighting

3rd hypothesis
AdaBoost

- Example:
  Recompute weighting
  4th hypothesis

- Example:
  Combination of hypotheses
AdaBoost

- Example:

![Decision surface](image1)

AdaBoost

- Example

![Final decision function](image2)
AdaBoost: Framework

**Input:**
- \( N \) examples \( \{(x_1, y_1), \ldots, (x_N, y_N)\} \)
- \( L \) a learning algorithm generating hypothesis \( h(x) \) (classifiers)
- \( T \) maxNumber of hypotheses in the ensemble

**Initialize:**
- \( d_n \) weight of example \( n \) (\( d \) is a distribution with \( 1 = \sum_{n=1}^{N} d_n^{(t)} \))
- \( d_n^{(1)} = 1/N \) for all \( n = 1, \ldots, N \)

**Do for** \( t = 1, \ldots, T \),
1. **Train base learner** according to example distribution \( d^{(t)} \) and obtain hypothesis \( h_t : x \mapsto \{\pm 1\} \).
2. **compute weighted error** \( \varepsilon_t = \sum_{n=1}^{N} d_n^{(t)} \mathbf{1}(y_n \neq h_t(x_n)) \)
3. **compute hypothesis weight** \( \alpha_t = \frac{1}{2} \ln \frac{1+\varepsilon_t}{1-\varepsilon_t} \)
4. **update example distribution** \( d_n^{(t+1)} = d_n^{(t)} \exp (-\alpha_t y_n h_t(x_n)) / Z_t \)

**Output:** final hypothesis \( f_{ens}(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \)

---

**AdaBoost**

\( t = 1 \)
\( d_n^{(1)} = 1/10 \quad N = 10 \)

\( \varepsilon_1 = \sum_{n=1}^{N} d_n^{(1)} \mathbf{1}(y_n \neq h_1(x_n)) = 0.3 \)

\( \alpha_1 = \frac{1}{2} \ln \frac{1+\varepsilon_1}{1-\varepsilon_1} = 0.424 \)

\( f_{ens}(x) = \alpha_1 h_1(x) \)
AdaBoost

$t = 2$

$$d_n^{(2)} = d_n^{(1)} \exp \left( -\alpha_1 y_n h_1(x_n) \right) / Z_1$$

$Z_1$ is a normalization factor

$$\epsilon_j = \sum_{i=1}^w d_n^{(j)} \mathbf{1}(y_{x_i} \neq h_j(x_{x_i}))$$

$$\alpha_j = \frac{1}{2} \ln \frac{1 - \epsilon_j}{\epsilon_j}$$

$$f_{	ext{ens}}(x) = \alpha_1 h_1(x) + \alpha_2 h_2(x)$$

AdaBoost

$t = 3$

.........
AdaBoost: Framework

- Weak Learners used with Boosting
  - Decision stumps (axis parallel splits)
  - Decision trees (e.g. C4.5 by Quinlan 1996)
  - Multi-layer Neural networks (e.g. for OCR)
  - Radial basis function networks (e.g. UCI benchmarks, etc)

Decision trees:
- Hierarchical and recursive partitioning on the input space
- Many approaches, usually axis parallel splits

AdaBoost: AdaBoost vs. SVM

- Comparison AdaBoost vs. SVM

These decision lines are for a low noise case with similar generalization errors. In AdaBoost, RBF networks with 13 centers were used.
AdaBoost: Application

- Application
  - DT C4.5 as weak classifier
  - Spam, Zip Code OCR
  - Text classification: Schapire and Singer - Used stumps with normalized term frequency and multi-class encoding
  - OCR: Schwenk and Bengio (neural networks)
  - Natural language Processing: Collins; Haruno, Shirai and Ooyama
  - Image retrieval: Thieu and Viola
  - Medical diagnosis: Merle et al.
  - Fraud Detection: Rätsch & Müller 2001
  - Drug Discovery: Rätsch, Demiriz, Bennett 2002

AdaBoost: Information

- Internet [http://www.boosting.org](http://www.boosting.org)
- People List available at [http://www.boosting.org](http://www.boosting.org)
- Software Only few implementations (algorithms 'too simple') (cf. [http://www.boosting.org](http://www.boosting.org))