Machine Learning
AdaBoost

Dmitrij Schlesinger
WS2013/2014, 31.01.2014
Idea

Compose a “strong” classifier from “weak” ones
Compare with SVM – complex feature spaces, one classifier.

Given:

– a set of weak classifiers \( \mathcal{H} \). Example: linear classifiers for two classes \( h \in \mathcal{H} : \mathcal{X} \rightarrow \{-1, +1\} \)

\[
h(x) = \text{sign}\left(\langle x, w \rangle + b\right)
\]

– labeled training data \( \left((x_1, k_1), (x_2, k_2) \ldots (x_m, k_m)\right), x_i \in \mathcal{X}, k_i \in \{-1, +1\} \)

Find a strong classifier

\[
f(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)
\]

with \( h_t \in \mathcal{H}, \alpha_t \in \mathbb{R} \) that separates the training data as good as possible.
Power of the set of decision strategies

Is it possible to classify any training data without errors? If not any, which ones?

Yes, it is (if the number of used classifiers is not restricted).

Example for $x \in \mathbb{R}$, $h(x) = \text{sign}(\langle x, w \rangle + b) = \pm \text{sign}(x - b)$

Key idea: it is possible to build an “elementary” classifier for each particular data point that is “neutral” for all others.
Examples in $x \in \mathbb{R}^2$
Algorithm

Given: \( (x_1, k_1), (x_2, k_2) \ldots (x_m, k_m) \), \( x_i \in \mathcal{X}, k_i \in \{-1, +1\} \)

Initialize **Weights** for all samples as \( D^{(1)}(i) = 1/m \)

For \( t = 1, \ldots, T \)

1. Choose (learn) a weak classifier \( h_t \in \mathcal{H} \)
   taking into account the actual weights \( D^{(t)} \)

2. Choose \( \alpha_t \)

3. Update weights:

\[
D^{(t+1)}(i) = \frac{D^{(t)}(i) \cdot \exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t}
\]

with a normalizing factor \( Z_t \) so that \( \sum_i D^{(t+1)}(i) = 1 \).

The final strong classifier is:

\[
f(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)
\]
Choose (learn) a weak classifier $h_t \in \mathcal{H}$.

$$h_t = \arg \min_{h \in \mathcal{H}} \epsilon(D^{(t)}, h) = \arg \min_{h \in \mathcal{H}} \sum_i D^{(t)}(i) \cdot \delta(y_i, h(x_i))$$

i.e. choose the best one from a pre-defined family $\mathcal{H}$
– it can be SVM, Hinge Loss, whatever ...

Note: AdaBoost is a meta-algorithm, it can work practically with any classifier family $\mathcal{H}$

The specialty here – the data points are weighted with $D(i)$

**Pre-requrement**: the error $\epsilon(D^{(t)}, h) < 1/2$
– the best $h_t$ should be not worse than a random choice.

Otherwise – Stop.
Algorithm (2)

Choose $\alpha_t$.

The goal is to build $f(x)$ so that its error

$$
\epsilon(f) = \sum_i \delta(y_i, f(x_i))
$$

is minimal.

The upper bound for the overall error is $\epsilon(f) \leq \prod_{t=1}^{T} Z_t$.

→ choose $\alpha_t$ (greedy) so that the actual $Z_t$ is minimal

$$
Z_t = \sum_i D^{(t)}(i) \cdot \exp\left(-\alpha_t y_i h_t(x_i)\right) \rightarrow \min_{\alpha_t}
$$

The task is convex and differentiable → analytical solution

$$
\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \epsilon(D^{(t)}, h_t)}{\epsilon(D^{(t)}, h_t)} \right)
$$
Algorithm (3)

Update weights.

\[ D^{(t+1)}(i) \sim D^{(t)}(i) \cdot \exp\left(-\alpha_t y_i h_t(x_i)\right) \]

Note: \( \alpha_t > 0 \)

\( y_i h_t(x_i) > 0 \) (correct) \( \Rightarrow \) \( \exp\left(-\alpha_t y_i h_t(x_i)\right) < 1 \)

\( y_i h_t(x_i) < 0 \) (error) \( \Rightarrow \) \( \exp\left(-\alpha_t y_i h_t(x_i)\right) > 1 \)

The samples that are actually missclassified are supported

\( \Rightarrow \) the classifier \( h_{t+1} \) in the next round will attempt to classify properly just these.

Examples by Sochman, Tippetts, Freund
http://cseweb.ucsd.edu/~yfreund/adaboost/
Summary

History:
1990 – Boost-by-majority algorithm (Freund)
1995 – AdaBoost (Freund & Schapire)
1997 – Generalized version of AdaBoost (Schapire & Singer) (today)
2001 – AdaBoost in Face Detection (Viola & Jones)

Some interesting properties:
– AB is a simple linear combination of (linear) classifiers
– AB converges to the logarithm of the likelihood ratio
– AB has good generalization capabilities (?)
– AB is a feature selector

See also: http://www.boosting.org/
Viola & Jones’s Face Detection

*Rapid Object Detection using a Boosted Cascade of Simple Features (CVPR 2001)*

Haar features can be computed very fast

$24 \times 24$ window $\times \ldots \rightarrow 180,000$ feature values per position!!!

Weak classifier – convolution with Haar-kernel $\leq$ threshold

AdaBoost for learning – feature selector

The two best features

Best features – 0.1 to 0.3 error, the other ones – 0.4 to 0.5
Viola & Jones’s Face Detection

Database: 130 images, 507 faces

Overall error rate – about 7%