Machine Learning

AdaBoost - Lecture VIII

Videos for This Lecture

- Repetition Video (Part 0)
- AdaBoost (Part 1)
- Error Function Analysis (Part 2)
- Application (Part 3)

Course Outline

Basic Concepts

- Parametric Method,
- Bayesian Learning and Nonparametrics Methods
- Clustering and Mixture of Gaussians

Classification Approaches

- Linear Discriminants
- Ensemble Methods and Boosting
- Randomized Trees, Forest

Reinforcement Learning

- Classical Reinforcement Learning
- Deep Reinforcement Learning

Deep Learning

- Foundations
- Optimization

Recap: Stacking

Idea

Learn L classifiers (based on the training data)

Find a meta-classifier that takes as input the output of the L first-level classifiers.

Example

Learn L classifiers with leave-one-out cross-validation

Interpret the prediction of the L classifiers as L-dimensional feature vector

Learn ‘level-2’ classifier based on the examples generated this way

Slide source: B. Leibe Slides ML, 2015

Recap: Bayesian Model Averaging

Model Averaging

Suppose we have H different models $h=1,\ldots,H$ with prior probabilities $p(h)$

Construct the marginal distribution over the data set

$$p(X) = \sum_{h=1}^{H} p(X|h)p(h)$$

Average error of committee

$$\mathbb{E}_{COM} = \frac{1}{M}\mathbb{E}_{AV}$$

This suggests that the average error of a model can be reduced by a factor of M simply by averaging M versions of the model!

Unfortunately, this result depends on the assumption that the errors are all uncorrelated.

Slide source: B. Leibe Slides ML, 2015

Part 1, Video AdaBoost_p1

AdaBoost

- Intuition
- Algorithm
- Analysis
- Visualisation

Today's Topics

AdaBoost

- Intuition
- Algorithm
- Analysis
- Extensions

Error Function Analysis

- Exponential error

Application

- Face Detection
- Feature extraction

Slide source: B. Leibe Slides ML, 2015

AdaBoost – “Adaptive Boosting”

Main idea

Instead of resampling, re-weight misclassified training examples.

Increase the chance of being selected in a sampled training set.

Or increase the misclassification cost when training on the full set.

Components

$h_m(x)$: “weak” or base classifier

– Condition: <50% training error over any distribution

$H(x)$: “strong” or final classifier

AdaBoost:

Construct a strong classifier as a thresholded linear combination of the weighted weak classifiers:

$$H(\mathbf{x}) = sign\left(\sum_{m=1}^{M} \alpha_m h_m(\mathbf{x})\right)$$

Slide source: B. Leibe Slides ML, 2015

AdaBoost - Algorithm

1. Initialization: Set $w_n^{(1)} = \frac{1}{N}$ for $n = 1, ...N$

2. For $m = 1, ...M$ iterations

a) Train a new weak classifier $h_m(x)$ using the current weighting coefficients $W^{(m)}$ by minimizing the weighted error function

$$J_m = \sum_{n=1}^{N} w_n^{(m)} I(h_m(\mathbf{x}) \neq t_n)$$

$$I(A) = \begin{cases} 1 & \text{if } A \text{ is true} \\ 0 & \text{else} \end{cases}$$

b) Estimate the weighted error of this classifier on X:

$$\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(h_m(\mathbf{x}) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$$

c) Calculate a weighting coefficient for $h_m(x)$: $a_m = ?$

d) Update the weighting coefficients: $w_n^{(m+1)} = ?$

How should we do this exactly?

Slide source: B. Leibe Slides ML, 2015

AdaBoost – Minimizing Exponential Error

Exponential error function

$$E = \sum_{n=1}^{N} \exp\{-t_n f_m(\mathbf{x}_n)\}$$

where $f_m(x)$ is a classifier defined as a linear combination of base classifiers $h_l(x)$:

$$f_m(\mathbf{x}) = \frac{1}{2} \sum_{l=1}^{m} \alpha_l h_l(\mathbf{x})$$

Goal

Minimize E with respect to both the weighting coefficients α_l and the parameters of the base classifiers $h_l(x)$.

Slide source: B. Leibe Slides ML, 2015

AdaBoost – Minimizing Exponential Error

Sequential Minimization

Suppose that the base classifiers $h_1(x), \ldots, h_{m-1}(x)$ and their coefficients $\alpha_1, \ldots, \alpha_{m-1}$ are fixed.

=> Only minimize with respect to α_m and $h_m(x)$.

$$E = \sum_{n=1}^{N} \exp\left\{-t_n f_m(\mathbf{x}_n)\right\} \quad \textbf{with} \quad f_m(\mathbf{x}) = \frac{1}{2}\sum_{l=1}^{m} \alpha_l h_l(\mathbf{x})$$

$$= \sum_{n=1}^{N} \exp\left\{\underbrace{-t_n f_{m-1}(\mathbf{x}_n)}_{= \textit{const.}} - \frac{1}{2} t_n \alpha_m h_m(\mathbf{x}_n)\right\}$$

$$= \sum_{n=1}^{N} w_n^{(m)} \exp\left\{-\frac{1}{2} t_n \alpha_m h_m(\mathbf{x}_n)\right\}$$

Slide source: B. Leibe Slides ML, 2015

AdaBoost – Minimizing Exponential Error

$$E = \sum_{n=1}^{N} w_n^{(m)} \exp\left\{-\frac{1}{2} t_n \alpha_m h_m(\mathbf{x}_n)\right\}$$

Observation:

- **Correctly classified points:** $t_n h_m(\mathbf{x}_n) = +1$ $\Rightarrow$ **collect in** $\mathcal{T}_m$
- **Misclassified points:** $t_n h_m(\mathbf{x}_n) = -1$ $\Rightarrow$ **collect in** $\mathcal{F}_m$

Rewrite the error function as

$$E = e^{-\alpha_m/2} \sum_{n \in \mathcal{T}_m} w_n^{(m)} + e^{\alpha_m/2} \sum_{n \in \mathcal{F}_m} w_n^{(m)}$$

$$= \left(e^{\alpha_m/2} \qquad\qquad\right) \sum_{n=1}^{N} w_n^{(m)} I(h_m(\mathbf{x}_n) \neq t_n)$$

Slide source: B. Leibe Slides ML, 2015

AdaBoost – Minimizing Exponential Error

$$E = \sum_{n=1}^{N} w_n^{(m)} \exp\left\{-\frac{1}{2} t_n \alpha_m h_m(\mathbf{x}_n)\right\}$$

Observation:

- **Correctly classified points:** $t_n h_m(\mathbf{x}_n) = +1$ $\Rightarrow$ **collect in** $\mathcal{T}_m$
- **Misclassified points:** $t_n h_m(\mathbf{x}_n) = -1$ $\Rightarrow$ **collect in** $\mathcal{F}_m$

Rewrite the error function as

$$E = e^{-\alpha_m/2} \sum_{n \in \mathcal{T}_m} w_n^{(m)} + e^{\alpha_m/2} \sum_{n \in \mathcal{F}_m} w_n^{(m)}$$

$$= \left(e^{\alpha_m/2} - e^{-\alpha_m/2}\right) \sum_{n=1}^{N} w_n^{(m)} I(h_m(\mathbf{x}_n) \neq t_n) + e^{-\alpha_m/2} \sum_{n=1}^{N} w_n^{(m)}$$

Slide source: B. Leibe Slides ML, 2015

AdaBoost – Minimizing Exponential Error

Minimize respect to $h_m(\mathbf{x})$: $\frac{\partial E}{\partial h_m(\mathbf{x}_n)} \stackrel{!}{=} 0$

$$E = \underbrace{\left(e^{\alpha_m/2} - e^{-\alpha_m/2}\right)}_{= const.} \sum_{n=1}^{N} w_n^{(m)} I(h_m(\mathbf{x}_n) \neq t_n) + \underbrace{e^{-\alpha_m/2} \sum_{n=1}^{N} w_n^{(m)}}_{= const.}$$

=> This is equivalent to minimizing

$$J_m = \sum_{n=1}^{N} w_n^{(m)} I(h_m(\mathbf{x}) \neq t_n)$$

(our weighted error function from step 2a) of the algorithm)

=> *We're on the right track. Let's continue…*

Slide source: B. Leibe Slides ML, 2015

AdaBoost – Minimizing Exponential Error

Minimize respect to α_m: $\dfrac{\partial E}{\partial \alpha_m} \stackrel{!}{=} 0$

$$E = \left(e^{\alpha_m/2} - e^{-\alpha_m/2}\right) \sum_{n=1}^{N} w_n^{(m)} I(h_m(\mathbf{x}_n) \neq t_n) + e^{-\alpha_m/2} \sum_{n=1}^{N} w_n^{(m)}$$

$$\left(\cancel{\frac{1}{2}} e^{\alpha_m/2} + \cancel{\frac{1}{2}} e^{-\alpha_m/2}\right) \sum_{n=1}^{N} w_n^{(m)} I(h_m(\mathbf{x}_n) \neq t_n) \stackrel{!}{=} \cancel{\frac{1}{2}} e^{-\alpha_m/2} \sum_{n=1}^{N} w_n^{(m)}$$

weighted error $\epsilon_m := \dfrac{\sum_{n=1}^{N} w_n^{(m)} I(h_m(\mathbf{x}_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}} = \dfrac{e^{-\alpha_m/2}}{e^{\alpha_m/2} + e^{-\alpha_m/2}}$

$$\epsilon_m = \frac{1}{e^{\alpha_m} + 1}$$

$\Rightarrow$ ***Update for the α coefficients:*** $\alpha_m = \ln\left\{\dfrac{1 - \epsilon_m}{\epsilon_m}\right\}$

Slide source: B. Leibe Slides ML, 2015

AdaBoost – Minimizing Exponential Error

Remaining step: update the weights

Recall that

$$E = \sum_{n=1}^{N} \underbrace{w_n^{(m)} \exp\left\{-\frac{1}{2} t_n \alpha_m h_m(\mathbf{x}_n)\right\}}_{\text{This becomes } w_n^{(m+1)} \text{ in the next iteration.}}$$

Therefore

$$\begin{aligned} w_n^{(m+1)} &= w_n^{(m)} \exp\left\{-\frac{1}{2} t_n \alpha_m h_m(\mathbf{x}_n)\right\} \\ &= \ldots \\ &= w_n^{(m)} \exp\left\{\alpha_m I(h_m(\mathbf{x}_n) \neq t_n)\right\} \end{aligned}$$

=> *Update for the weight coefficients*

Slide source: B. Leibe Slides ML, 2015

AdaBoost – Final Algorithm

1. Initialization: Set $w_n^{(1)} = \frac{1}{N}$ for $n = 1,\ldots,N$
2. For $m = 1,\ldots,M$ iterations
   a) Train a new weak classifier $h_m(\mathbf{x})$ using the current weighting coefficients $\mathbf{W}^{(m)}$ by minimizing the weighted error function

$$J_m = \sum_{n=1}^{N} w_n^{(m)} I(h_m(\mathbf{x}) \neq t_n)$$

   b) Estimate the weighted error of this classifier on $\mathbf{X}$:

$$\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(h_m(\mathbf{x}) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$$

   c) Calculate a weighting coefficient for $h_m(\mathbf{x})$: $\alpha_m = \ln\left\{\frac{1-\epsilon_m}{\epsilon_m}\right\}$
   d) Update the weighting coefficients:

$$w_n^{(m+1)} = w_n^{(m)} \exp\{\alpha_m I(h_m(\mathbf{x}_n) \neq t_n)\}$$

Slide source: B. Leibe Slides ML, 2015

AdaBoost Visualisation

AdaBoost Visualisation

Part 2, Video AdaBoost_p2

Error Function Analysis

- Exponential error

Today's Topics

AdaBoost

- Intuition
- Algorithm
- Analysis
- Extensions

Error Function Analysis

- Exponential error

Application

- Face Detection
- Feature extraction

Slide source: B. Leibe Slides ML, 2015

Recap: Error Functions

$t_n \in \{-1, 1\}$

Ideal misclassification error function (black)

- This is what we want to approximate,
- Unfortunately, it is not differentiable.
- The gradient is zero for misclassified points.

=> We cannot minimize it by gradient descent.

Slide source: B. Leibe Slides ML, 2015

Recap: Error Functions

Squared error used in Least-Squares Classification

- Very popular, leads to closed-form solutions.
- However, sensitive to outliers due to squared penalty.
- Penalizes "too correct" data points

=> Generally does not lead to good classifiers.

Slide source: B. Leibe Slides ML, 2015

Recap: Error Functions

"Hinge error" used in SVMs

- Zero error for points outside the margin ($z_n > 1$) => sparsity.
- Linear penalty for misclassified points ($z_n < 1$) => robustness.
- Not differentiable around $z_n = 1$ => Cannot be optimized directly

Slide source: B. Leibe Slides ML, 2015

Discussion: AdaBoost Error Functions

Exponential error used in AdaBoost

- Zero error for points outside the margin ($z_n > 1$) => sparsity.
- Sequential minimization leads to simple AdaBoost scheme
- Properties?

Slide source: B. Leibe Slides ML, 2015

Discussion: AdaBoost Error Functions

Exponential error used in AdaBoost

- No penalty for too correct data points, fast convergence.
- Disadvantage: exponential penalty for large negative values!

=> Less robust to outliers or misclassified data points!?

Slide source: B. Leibe Slides ML, 2015

Discussion: Other Possible Error Functions

$$E = -\sum \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}$$

'Cross-entropy error' used in Logistic Regression

- Similar to exponential error for z>0.
- Only grows linearly with large negative values of z

=> Make AdaBoost more robust by switching to this error function.

=> 'GentleBoost'

Slide source: B. Leibe Slides ML, 2015

Summary: AdaBoost

Properties

- Simple combination of multiple classifiers.
- Easy to implement
- Can be used with many different types of classifiers
  - None of them needs to be too good on its own
  - In fact, they only have to be slightly better than chance
- Commonly used in many areas
- Empirically good generalization capabilities

Limitations

- Original AdaBoost sensitive to misclassified training data points.
  - Because of exponential error function
  - Improvement by GentleBoost
- Two-class classifier
  - Multiclass extensions available

Slide source: B. Leibe Slides ML, 2015

Part 3, Video AdaBoost_p3

Application

- Face Detection
- Feature extraction

Today's Topics

AdaBoost

- Intuition
- Algorithm
- Analysis
- Extensions

Error Function Analysis

- Exponential error

Application

- Face Detection
- Feature extraction

Slide source: B. Leibe Slides ML, 2015

Example Application: Face Detection

Frontal faces are a good example of a class where global appearance models + a sliding window detection approach fit well:

- Regular 2D structure
- Center of face almost shaped like a “patch”/windowt

Now we’ll take AdaBoost and see how the ViolaJones face detector works

Slide source: B. Leibe Slides ML, 2015

Feature extraction

“Rectangular” filters

Feature output is difference between adjacent regions

Efficiently computable with integral image: any sum can be computed in constant time

Avoid scaling images -> scale features directly for same cost

Integral image

D = 1+4-(2+3)=
= A+(A+B+C+D)-(A+C+A+B) = D

Slide source: B. Leibe Slides ML, 2015

Large Library of Filters

180,000+ possible features associated with each 24 x 24 window

Use AdaBoost both to select the informative features and to form the classifier

Considering all possible filter parameters: position, scale, and type:

Slide source: B. Leibe Slides ML, 2015

AdaBoost for Feature+Classifier Selection

Want to select the single rectangle feature and threshold that best separates positive (faces) and negative (nonfaces) training examples, in terms of *weighted* error

Outputs of a possible rectangle feature on a face and non-face

Resulting weak classifier:

$$h_t(x) = \begin{cases} +1 & \text{if } f_t(x) > \theta_t \\ -1 & \text{otherwise} \end{cases}$$

For next round, reweight the examples according to errors, choose another filter/ threshold combo.

Slide source: B. Leibe Slides ML, 2015

AdaBoost for Efficient Feature Selection

Image features = weak classifiers

For each round of boosting:

- Evaluate each rectangle filter on each example
- Sort examples by filter values
- Select best threshold for each filter (min error)
  - Sorted list can be quickly scanned for the optimal threshold
- Select best filter/threshold combination
- Weight on this features is a simple function of error rate
- Reweight examples

P. Viola, M. Jones, Robust Real-Time Face Detection, IJCV, Vol. 57(2), 2004.(first version appeared at CVPR 2001)

Slide source: B. Leibe Slides ML, 2015

Viola-Jones Face Detector: Results

Slide source: B. Leibe Slides ML, 2015

Viola-Jones Face Detector: Results

Slide source: B. Leibe Slides ML, 2015

Viola-Jones Face Detector: Results

Slide source: B. Leibe Slides ML, 2015

Readings

Bishop's book

Chapter 14.1 - 14.3 9 (More information on Classifier Combination and Boosting)

Additional information

A more in-depth discussion of the statistical interpretation of AdaBoost is available in the following paper:

J. Friedman, T. Hastie, R. Tibshirani, Additive Logistic Regression: a Statistical View of Boosting, The Annals of Statistics, Vol. 38(2), pages 337-374, 2000

Slide is taken from B. Leibe, ML, 2017

You should be able to answer …

- AdaBoost: Main idea, components, intuition
- AdaBoost algorithm be able to write the formulas as pseudo code
- Which error function AdaBoost is minimising, which properties has this error function?
- Properties and limitations of AdaBoost.
- The idea behind feature extraction application.