

Machine Learning- Exercise 3

AdaBoost

Paul Voigtlaender & Sabarinath Mahadevan
`<lastname>@vision.rwth-aachen.de`

RWTH Aachen University - Computer Vision Group
<http://www.vision.rwth-aachen.de/>

2018-12-05



Content

1. AdaBoost

Content

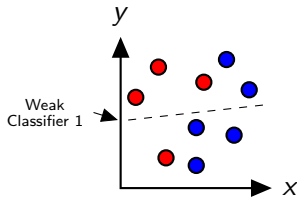
1. AdaBoost

Adaboost[Freund & Schapire, 1996]-Recap

- ▶ Main idea
 - ▶ Instead of resampling, reweight 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
 - ▶ $c_k(\mathbf{x})$: “weak” or base classifier
 - ▶ Condition: $< 50\%$ training error over any distribution
 - ▶ $C(\mathbf{x})$: “strong or final classifier
- ▶ Adaboost:
 - ▶ Construct a strong classifier as a thresholded linear combination of the weighted classifiers:

$$C(\mathbf{x}) = \text{sign} \left(\sum_{k=1}^K \alpha_k c_k(\mathbf{x}) \right)$$

Adaboost - Recap

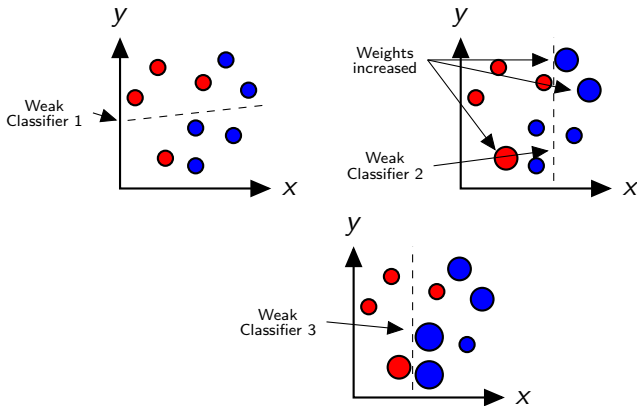


Consider a 2D feature space with **positive** and **negative** examples.

Each weak classifier splits the training examples with at least 50% accuracy.

Examples misclassified by a previous weak learner are given more emphasis at future rounds.

Adaboost - Recap



- Final classifier is combination of the weak classifiers.

Adaboost - Algorithm

- ▶ Initialization: Set $w_n^{(1)} = \frac{1}{N}$ for $n = 1, \dots, N$.
- ▶ For $k = 1, \dots, k$ iterations
 - ▶ Train a new weak classifier $c_k(\mathbf{X})$ using current weights $\mathbf{W}^{(k)}$ by minimizing the weighted error function
 - ▶ estimate the weighted error of this classifier on \mathbf{X} :

$$\epsilon_k = \frac{\sum_{n=1}^N w_n^{(k)} I(c_k(\mathbf{X}) \neq y_n)}{\sum_{n=1}^N w_n^{(k)}}$$

- ▶ Calculate a weighting coefficient for $c_k(\mathbf{X})$:

$$\alpha_k = \ln \left\{ \frac{1 - \epsilon_k}{\epsilon_k} \right\}$$

- ▶ Update the weighting coefficients:

$$w_n^{(k+1)} = w_n^{(k)} \exp\{\alpha_k I(c_k(\mathbf{X}) \neq y_n)\}$$