Advanced Machine Learning

Bogdan Alexe,

bogdan.alexe@fmi.unibuc.ro

University of Bucharest, 2^{nd} semester, 2021-2022

Administrative

- exam date face to face: 9^{th} of June 9 - 12am
- you are allowed to have written material (whatever you want)
- you are not allowed to bring electronic devices (laptop, phone, etc)
- you will receive 3 - 4 problems of varying difficulty

Recap - AdaBoost

- construct distribution $\mathbf{D}^{(t)}$ on $\{1,..., m\}$:
  - $\mathbf{D}^{(1)}(i) = 1/m$
  - given $\mathbf{D}^{(t)}$ and h_t: $D^{(t+1)}(i) = \dfrac{D^{(t)}(i) \times e^{-w_t h_t(x_i) y_i}}{Z_{t+1}}$

where Z_{t+1} normalization factor ($\mathbf{D}^{(t+1)}$ is a distribution): $Z_{t+1} = \sum_{i=1}^{n} D^{(t)}(i) \times e^{-w_t h_t(x_i) y_i}$

w_t is a weight: $w_t = \dfrac{1}{2}\ln(\dfrac{1}{\varepsilon_t} - 1) > 0$ as the error $\varepsilon_t < 0.5$

ε_t is the error of h_t on $\mathbf{D}^{(t)}$: $\varepsilon_t = \Pr_{i \sim D^{(t)}}[h_t(x_i) \neq y_i] = \sum_{i=1}^{m} D^{(t)}(i) \times 1_{[h_t(x_i) \neq y_i]}$

If example $\mathbf{x}_i$ is correctly classified then $h_t(\mathbf{x}_i) = y_i$ so at the next iteration t+1 its importance (probability distribution) will be decreased to $D^{(t+1)}(i) = \dfrac{D^{(t)}(i) \times e^{-w_t}}{Z_{t+1}}$

If example x_i is misclassified then $h_t(x_i) \neq y_i$ so at the next iteration t+1 its importance (probability distribution) will be increased to $D^{(t+1)}(i) = \dfrac{D^{(t)}(i) \times e^{w_t}}{Z_{t+1}}$

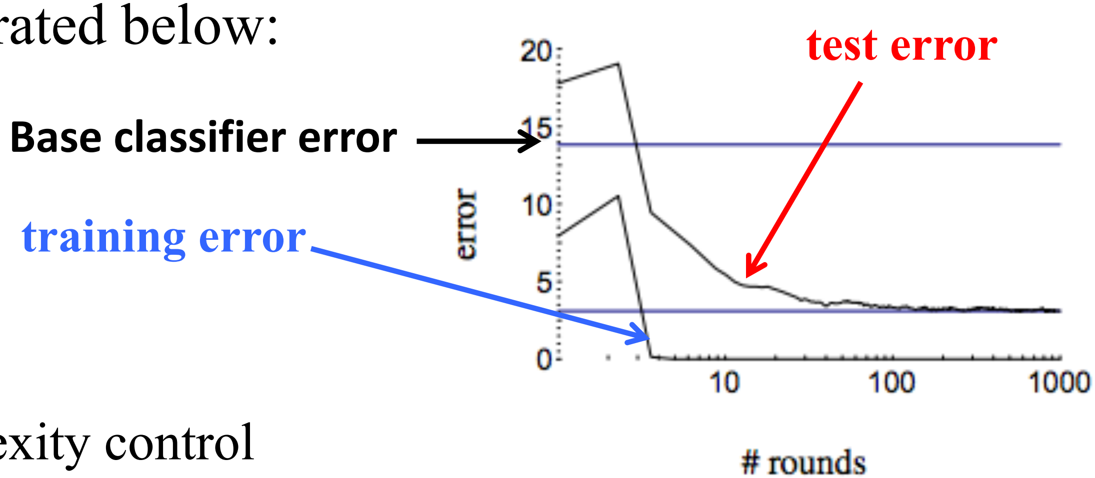
- output final/combined classifier h_{final}: $h_{final}(x) = sign(\sum_{t=1}^{T} w_t h_t(x))$

Recap - Generalization error for AdaBoost

$\mathrm{VCdim}(L(\mathcal{B},T)) \leq T \times (\mathrm{VCdim}(\mathcal{B})+1) \times (3\times\log(T \times (\mathrm{VCdim}(\mathcal{B})+1))+2)$.

The upper bound grows as $O(T\times\mathrm{VCdim}(\mathcal{B}) \times \log(T\times\mathrm{VCdim}(\mathcal{B})))$, thus, the bound suggests that AdaBoost could overfit for large values of T, and indeed this can occur.

However, in many cases, it has been observed empirically that the generalization error of AdaBoost decreases as a function of the number of rounds of boosting T, as illustrated below:



- number of rounds T is complexity control
- use validation set + “early stopping” to select T

Recap: Viola-Jones face detector

ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION 2001

Rapid Object Detection using a Boosted Cascade of Simple Features

Paul Viola
viola@merl.com
Mitsubishi Electric Research Labs
201 Broadway, 8th FL
Cambridge, MA 02139

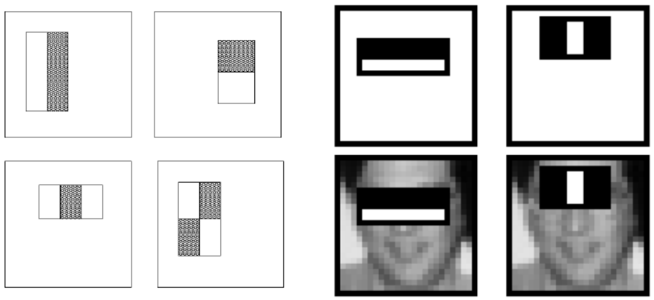
Michael Jones
mjones@crl.dec.com
Compaq CRL
One Cambridge Center
Cambridge, MA 02142

Abstract

This paper describes a machine learning approach for vi-

tected at 15 frames per second on a conventional 700 MHz Intel Pentium III. In other face detection systems, auxiliary information, such as image differences in video sequences,

Recap: Viola-Jones face detector: features



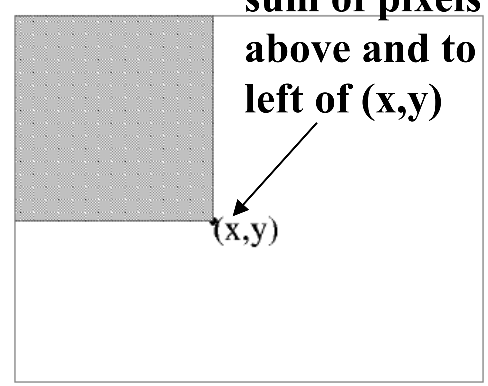
“Rectangular” filters

Feature output is difference between adjacent regions

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

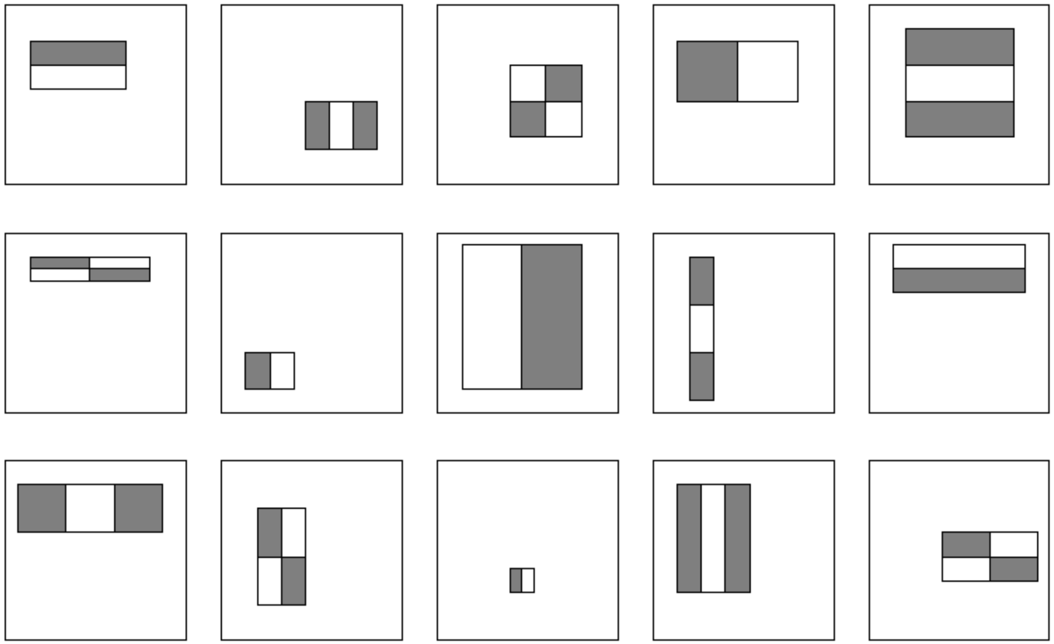
Avoid scaling images → scale features directly for same cost

Value at (x,y) is sum of pixels above and to the left of (x,y)



Integral image

Recap: Viola-Jones face detector: features



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

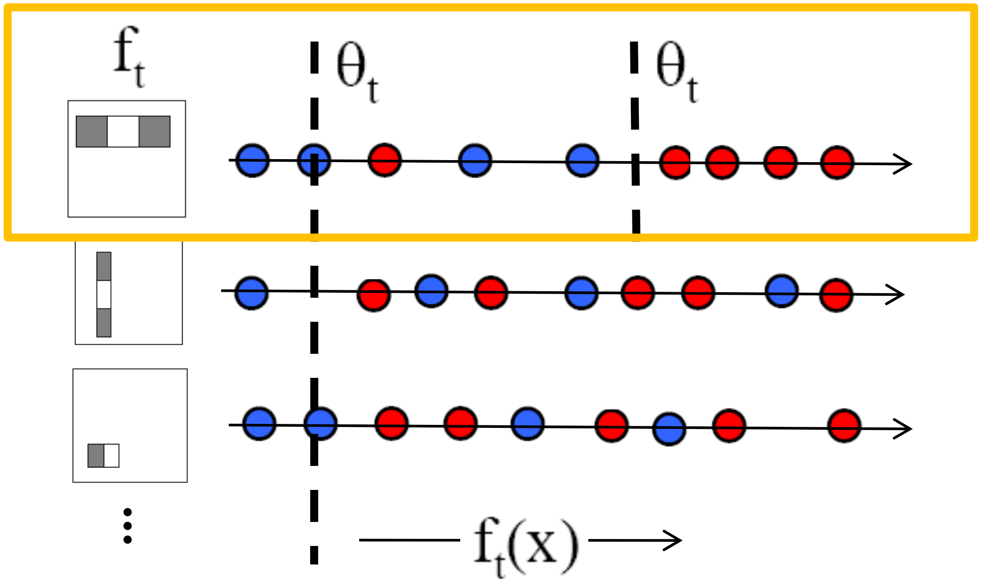
180,000+ possible features associated with each 24 x 24 window

Which subset of these features should we use to determine if a window has a face?

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

Recap: Viola-Jones detector: AdaBoost

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



Outputs of a possible rectangle feature on faces and non-faces.

Resulting weak classifier:

imum number of examples are misclassified. A weak classifier $h_j(x)$ thus consists of a feature f_j, a threshold θ_j and a parity p_j indicating the direction of the inequality sign:

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases}$$

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

The resulting weak classifier is in fact from $\mathcal{H}_{DS}{}^{d} = \{h_{i,\theta,b}\colon \mathbf{R}^d \to \{-1,1\},$
$h_{i,\theta,b}(\mathbf{x}) = \mathrm{sign}(\theta - x_i) \times b,\ 1 \le i \le d,\ \theta \in \mathbf{R},\ b \in \{-1,+1\}\}$

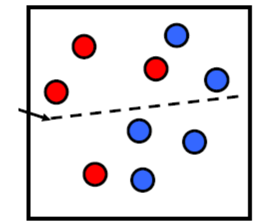
AdaBoost Algorithm

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:

  1. Normalize the weights,
  $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
  so that w_t is a probability distribution.
  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i \, |h_j(x_i) - y_i|$.
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. Update the weights:
  $$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$
  where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

- The final strong classifier is:
$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
where $\alpha_t = \log \frac{1}{\beta_t}$

Start with uniform weights on training examples



$\{\mathbf{x_1, \ldots x_n}\}$

For T rounds

Evaluate *weighted* error for each feature, pick best.

Re-weight the examples:
Incorrectly classified -> more weight
Correctly classified -> less weight

Final classifier is combination of the weak ones, weighted according to error they had.

Freund & Schapire 1995

Today's lecture: Overview

- Assignment 2

Assignment 2

Deadline: Year 1 - Sunday, 19^{th} of June, 23:59,
Year 2 - Sunday, 12^{th} of June, 23:59

Upload your solutions as a zip archive at:
`https://tinyurl.com/AML-2022-ASSIGNMENT2`

1. **(1.5 points)** Consider $\mathcal{H}$ the class of 3-piece classifiers (signed intervals):

$$\mathcal{H} = \{h_{a,b,s} : \mathbb{R} \to \{0,1\} \mid a \leq b, s \in \{-1,1\}\}, \ where \ h_{a,b,s}(x) = \begin{cases} s, & x \in [a,b] \\ -s, & x \notin [a,b] \end{cases}$$

a. Compute the shattering coefficient $\tau_H(m)$ of the growth function for $m \geq 0$ for hypothesis class $\mathcal{H}$. **(1 point)**

b. Compare your result with the general upper bound for the growth functions and show that $\tau_H(m)$ obtained at previous point a is not equal with the upper bound. **(0.25 points)**

c. Does there exist a hypothesis class $\mathcal{H}$ for which is equal to the general upper bound (over or another domain $\mathcal{X}$)? If your answer is yes please provide an example, if your answer is no please provide a justification. **(0.25 points)**

Problem 2

2. **(1.5 points)** Consider de concept class C_2 formed by the union of two closed intervals $[a, b] \cup [c, d]$, where $a, b, c, d \in \mathbb{R}, a \leq b \leq c \leq d$. Give an efficient ERM algorithm for learning the concept class C_2 and compute its complexity for each of the following cases:

a. realizable case. **(1 point)**

b. agnostic case. **(0.5 point)**

Problem 3

3. **(1.5 points)** Consider a modified version of the AdaBoost algorithm that runs for exactly three rounds as follows:

- the first two rounds run exactly as in AdaBoost (at round 1 we obtain distribution $\mathbf{D}^{(1)}$, weak classifier h_1 with error ϵ_1; at round 2 we obtain distribution $\mathbf{D}^{(2)}$, weak classifier h_2 with error ϵ_2).
- in the third round we compute for each $i = 1, 2, \ldots, m$:

$$\mathbf{D}^{(3)}(i) = \begin{cases} \frac{D^{(1)}(i)}{Z}, & if\ h_1(x) \neq h_2(x) \\ 0, & otherwise \end{cases}$$

  where Z is a normalization factor such that $\mathbf{D}^{(3)}$ is a probability distribution.

- obtain weak classifier h_3 with error ϵ_3.
- output the final classifier $h_{final}(x) = sign(h_1(x) + h_2(x) + h_3(x))$.

Assume that at each round $t = 1, 2, 3$ the weak learner returns a weak classifier h_t for which the error ϵ_t satisfies $\epsilon_t \leq \frac{1}{2} - \gamma_t, \gamma_t > 0$.

a. What is the probability that the classifier h_1 (selected at round 1) will be selected again at round 2? Justify your answer. **(0.75 points)**

b. Consider $\gamma = min\{\gamma_1, \gamma_2, \gamma_3\}$. Show that the training error of the final classifier h_{final} is at most $\frac{1}{2} - \frac{3}{2}\gamma + \gamma^2$ and show that this is strictly smaller than $\frac{1}{2} - \gamma$. **(0.75 points)**

Problem 4

4. **(1 point)** Consider H^d_{2DNF} the class of 2-term disjunctive normal form formulae consisting of hypothesis of the form $h : \{0,1\}^d \to \{0,1\}$,

$$h(x) = A_1(x) \vee A_2(x)$$

where $A_i(x)$ is a Boolean conjunction of literals H^d_{conj}.

It is known that the class H^d_{2DNF} is not efficient properly learnable but can be learned improperly considering the class H^d_{2CNF}. Give a γ-weak-learner algorithm for learning the class H^d_{2DNF} which is not a stronger PAC learning algorithm for H^d_{2DNF} (like the one considering H^d_{2CNF}). Prove that this algorithm is a γ-weak-learner algorithm for H^d_{2DNF}.

Hint: Find an algorithm that returns $h(x) = 0$ or the disjunction of 2 literals.

Ex-oficio: 0.5 points.