Ensemble Methods

Tutorial

Introduction to Machine Learning
Fall 2014

Alexey Gronskiy and Yatao Bian

19–21 Nov. 2014

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Outline

- **Project**
- **Homework**
- **Ensemble methods**
- **Applications**
- **Demo**

Project: Text Classification

- **Unknown/difficult language describing cities and countries**
- **Names are often misspelled and thus hard to automatically match with codes**
- **Data:**

  x : city name

  y_1 : city code

  y_2 : country code

Project: Text Classification

- **Different capitalization:**

  yrjhnjcnfy can become Yrjhnjcnfy or YFIRJHNJCNFY

- **Missing words:**

  eas cjdtncrbv u hy hedl becomes eas cjdtncrbv u hedl

- **Missing letters in a word:**

  yrjhnjcnfy becomes rjhnjcnfy

- **Wrong letters in a word:**

  yrjhnjcnfy becomes ybirjhnjcnfy

- **Different forms of the word:**

  yrjhnjcnfy becomes yrjhnjcndi

Project: Text Classification

- **Noise:** mistagged city names (only within country)
- **Grading:**
  - You are penalized by 1 for every city code that is misclassified
  - You are penalized by 0.25 for every country code that is misclassified
- **Similar to previous project:** submission website, validation/testing sets
- **Deadline:** Friday, 19 Dec. 2014 at 23:59:59

Bag of words

Representation of a Text Object

- Create dictionary: list of all words that can be found
- Each word of the dictionary corresponds to a feature
- Create feature vectors based on how many times each word appears

Bag of words
Example

News Headlines

L1: Switzerland votes against cap on executive pay

L2: 12:1 salary cap fails in Switzerland

L3: Switzerland votes not to cap the boss's pay

L4: Corporate executive pay limit rejected In Switzerland voting

L5: Switzerland votes down measure to limit executive pay

L6: How mushrooms create own micro climate

Bag of words

Example

News Headlines

L1: Switzerland votes against cap on executive pay

L2: 12:1 salary cap fails in Switzerland

L3: Switzerland votes not to cap the boss's pay

L4: Corporate Executive Pay Limit Rejected In Switzerland Voting

L5: Switzerland votes down Measure to limit Executive pay

L6: How mushrooms create own micro climate

Headline	Salary	Cap	Fails	Switzerland	Good	News	Votes	Against	Executive	Pay	Down	Measure	Limit	Corporate	Rejected	Voting	Boss	Mushrooms	Create	Micro	Climate	…
L1		1		1			1	1	1	1												
L2	1	1	1	1																		
L3		1		1			1			1							1					
L4				1					1	1			1	1	1	1						
L5				1			1		1	1		1	1									
L6																		1	1	1	1	

Ensemble Methods

- **Set of simple predictors**

- **Trained on modified versions of the training data set**

- **Combined to produce a single classifier**

Ensemble Methods

- **Set of simple predictors**

  Which predictors?
  - Decision stumps
  - Decision trees
  - (multilayer) perceptrons, etc.

- **Trained on modified versions of the training data set**

  How to train them to achieve diversity?
  - Resampling **(Bagging)**
  - Adaptive weighting **(Boosting)**

- **Combined to produce a single classifier**

  How to combine them?
  - Average **(Bagging)**
  - Weighted sum **(Boosting)**

Bagging

- **Choose a classifier class** $h_t(\mathbf{x}) \in \{-1, 1\}$

- **Diversity: train on resampled data set** Random sampling

- **Average individual outputs** $H(\mathbf{x}) = \mathrm{sign}\left(\sum_{t=0}^{T} h_t(\mathbf{x})\right)$

Intuition:

Responses of different classifiers can be regarded as independent

Errors of different classifiers will average out if they are uncorrelated

Boosting

- **Choose a classifier class** $h_t(\mathbf{x}) \in \{-1, 1\}$
- **Diversity: weight data differently** $D_1, D_2, \ldots, D_m$
- **Output: weighted combination** $$H(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=0}^{T} \alpha_t h_t(\mathbf{x})\right)$$

Boosting

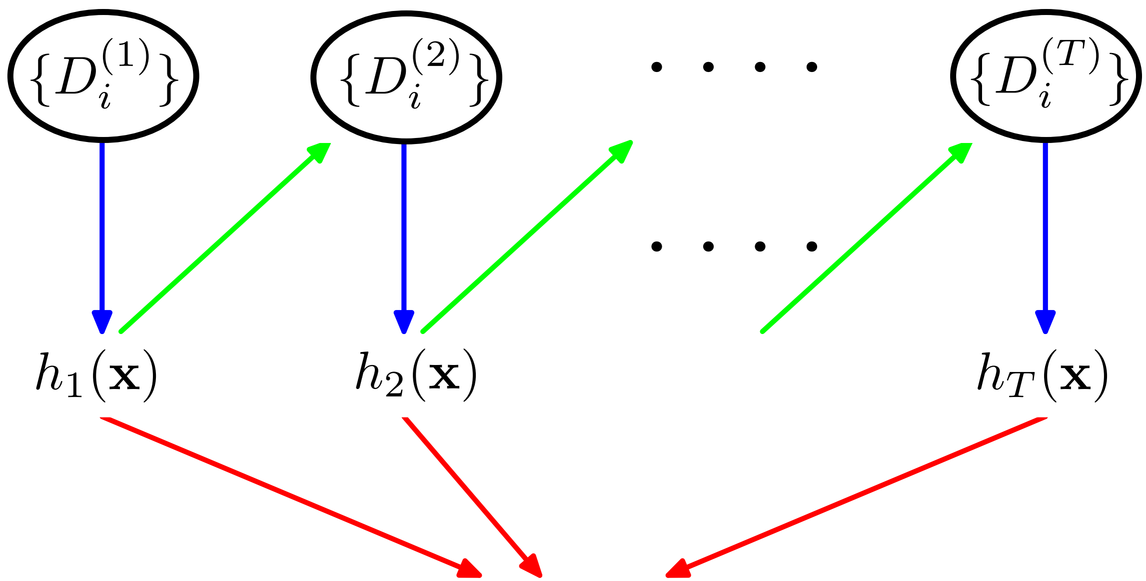
- **Choose a classifier class** $h_t(\mathbf{x}) \in \{-1, 1\}$
- **Diversity: weight data differently** $D_1, D_2, \ldots, D_m$
- **Output: weighted combination** $H(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=0}^{T} \alpha_t h_t(\mathbf{x})\right)$

Open questions:

How to set D_i

How to set α_t

AdaBoost



$$H(\mathbf{x}) = \mathrm{sign}\left(\sum_{t=0}^{T} \alpha_t h_t(\mathbf{x})\right)$$

From Pattern Recognition and Machine Learning. C. Bishop

AdaBoost: Adaptive Weighting

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error

$$\epsilon_t = \Pr_{i \sim D_t}\left[h_t(x_i) \neq y_i\right] = \sum_{i: h_t(x_i) \neq y_i} D_t(i).$$

- Choose $\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right)$.
- Update:

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \end{aligned}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

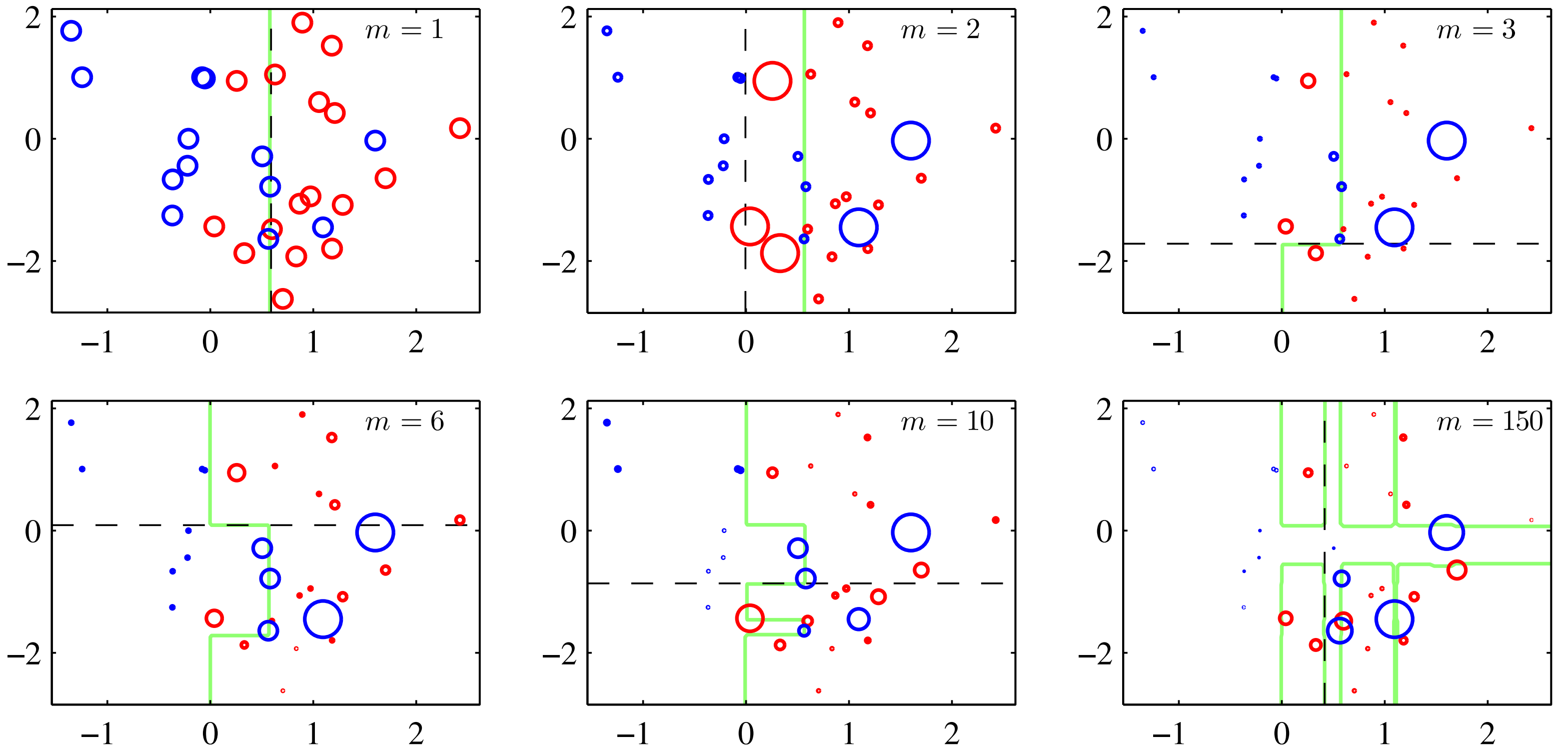
$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

From Y. Freund and R. Schaphire 1999

Observations

- $\epsilon \leq 0.5$
- $\alpha_t \geq 0$
- **The training distribution is shifted to emphasize the 'hard' cases**
- **Final result is a majority vote, weighted by accuracy**
- **The *margin* measures confidence in the prediction:** $\frac{y_i \sum_t \alpha_t h_t(\mathbf{x})}{\sum_t \alpha_t}$

AdaBoost



From Pattern Recognition and Machine Learning. C. Bishop

AdaBoost

- **Derived from minimizing exponential loss**

$$E = \sum_{n=1}^{N} \exp\{-t_n f_m(\mathbf{x}_n)\} \qquad f_m(\mathbf{x}) = \frac{1}{2}\sum_{l=1}^{m} \alpha_l y_l(\mathbf{x})$$

- **Too difficult: sequential greedy approach**

$$\begin{aligned} E &= \sum_{n=1}^{N} \exp\left\{-t_n f_{m-1}(\mathbf{x}_n) - \frac{1}{2} t_n \alpha_m y_m(\mathbf{x}_n)\right\} \\ &= \sum_{n=1}^{N} w_n^{(m)} \exp\left\{-\frac{1}{2} t_n \alpha_m y_m(\mathbf{x}_n)\right\} \end{aligned}$$

Loss Functions

Friedman et al. (2000)

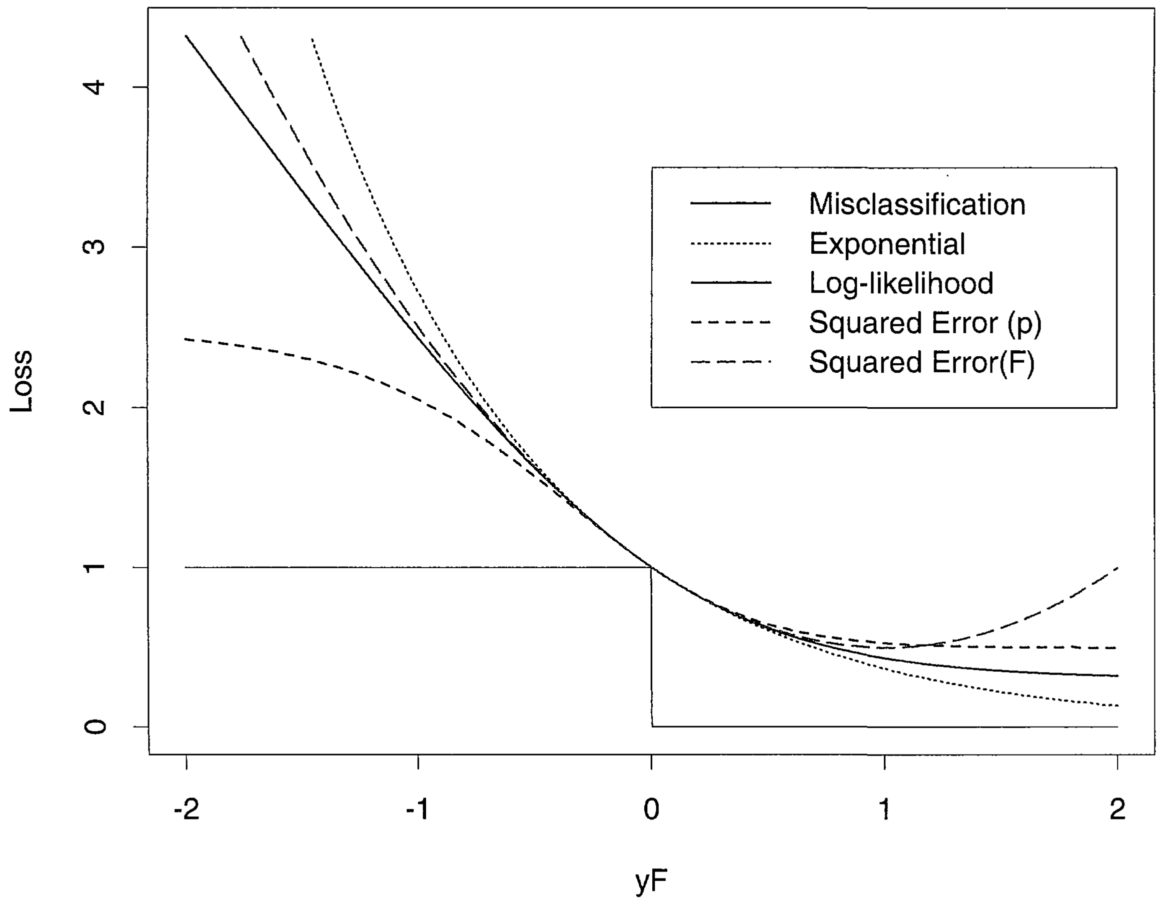


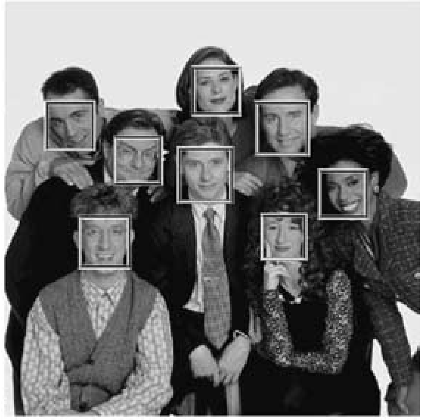
FIG. 2. *A variety of loss functions for estimating a function $F(x)$ for classification. The horizontal axis is yF, which is negative for errors and positive for correct classifications. All the loss functions are monotone in yF, and are centered and scaled to match e^{-yF} at $F = 0$. The curve labeled "Log-likelihood" is the binomial log-likelihood or cross-entropy $y^* \log p + (1 - y^*)\log(1 - p)$. The curve labeled "Squared Error(p)" is $(y^* - p)^2$. The curve labeled "Squared Error(F)" is $(y - F)^2$ and increases once yF exceeds 1, thereby increasingly penalizing classifications that are "too correct."*

Face Detection with AdaBoost

P. Voila, M. Jones, Int. J. Computer Vision 57(2), 137-154 (2004)

Goal

- Real-time (15 frames per second)
- Competitive detection rates

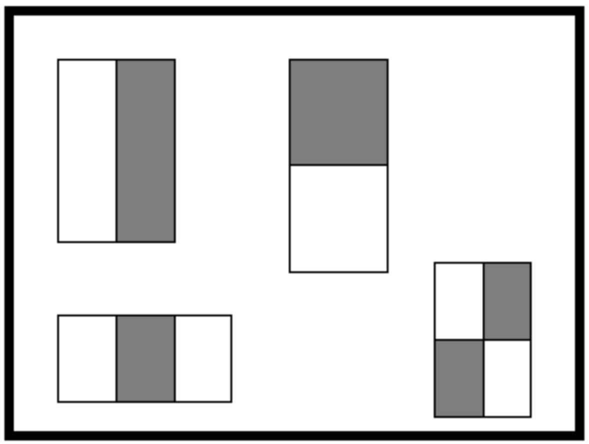


System Components

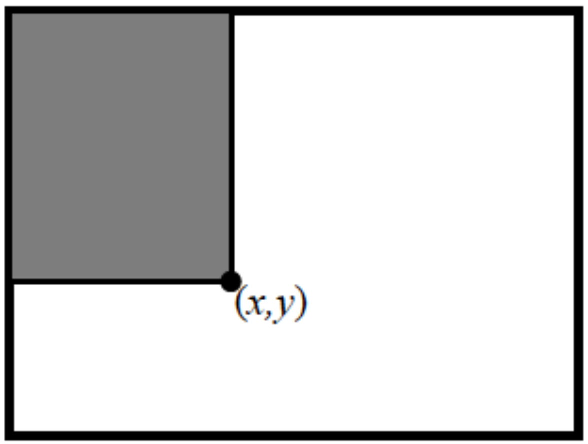
- Image representation: “Integral Images”
- Feature selection by AdaBoost
- Cascade of classifier for quick rejection

Features

Two rectangle feature:
Difference between the sum of the pixels within two rectangular regions



Integral image representation:
Sum of all pixels above and to the left



Rectangle features **rapidly calculated using** ***integral image representation***

Feature Selection by AdaBoost

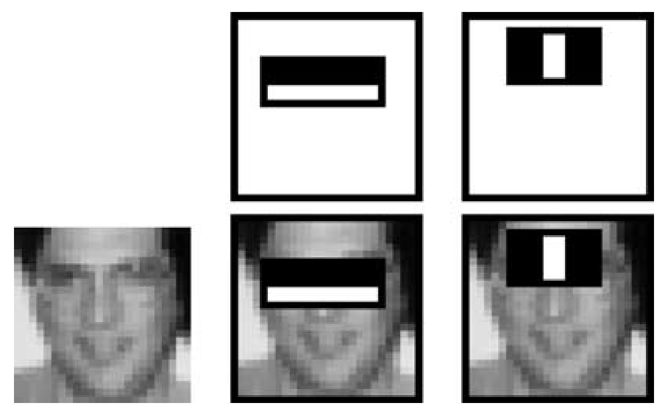
- **Very large number of features**

  160000 rectangle features in a 24x24 pixel sub-window

- **Train classifier and learn features at the same time**
  - AdaBoost: stronger learners get higher weights
  - One feature per learner
  - Take features associated with higher weights

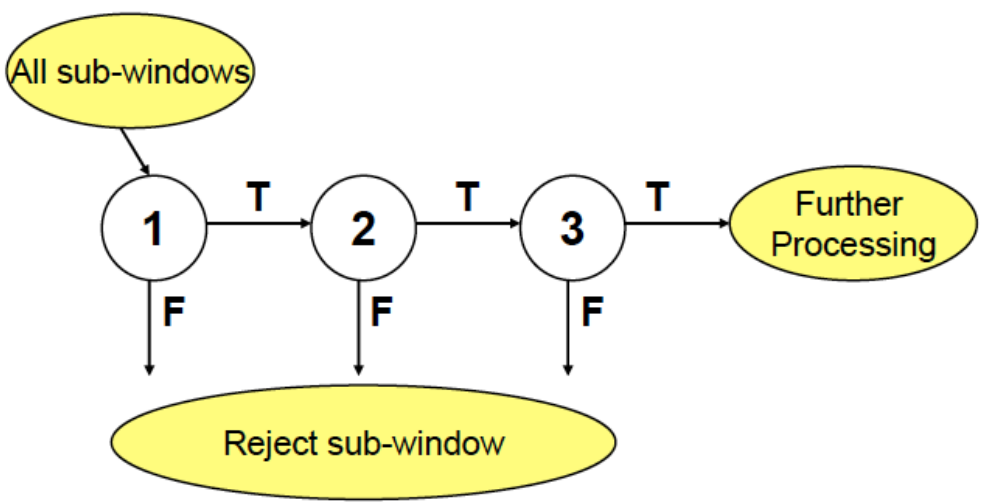
$$H(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=0}^{T} \alpha_t h_t(\mathbf{x})\right)$$

- **200 features yield reasonable results**



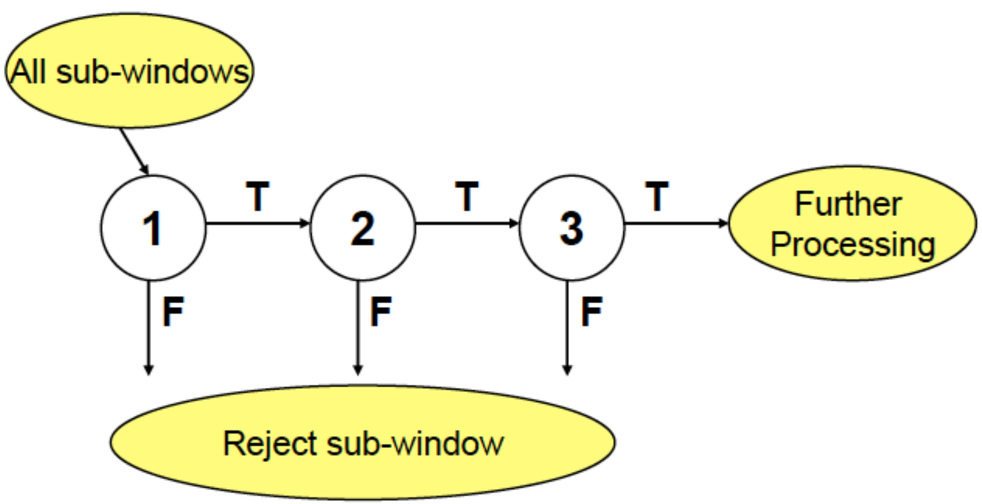
Attentional Cascade

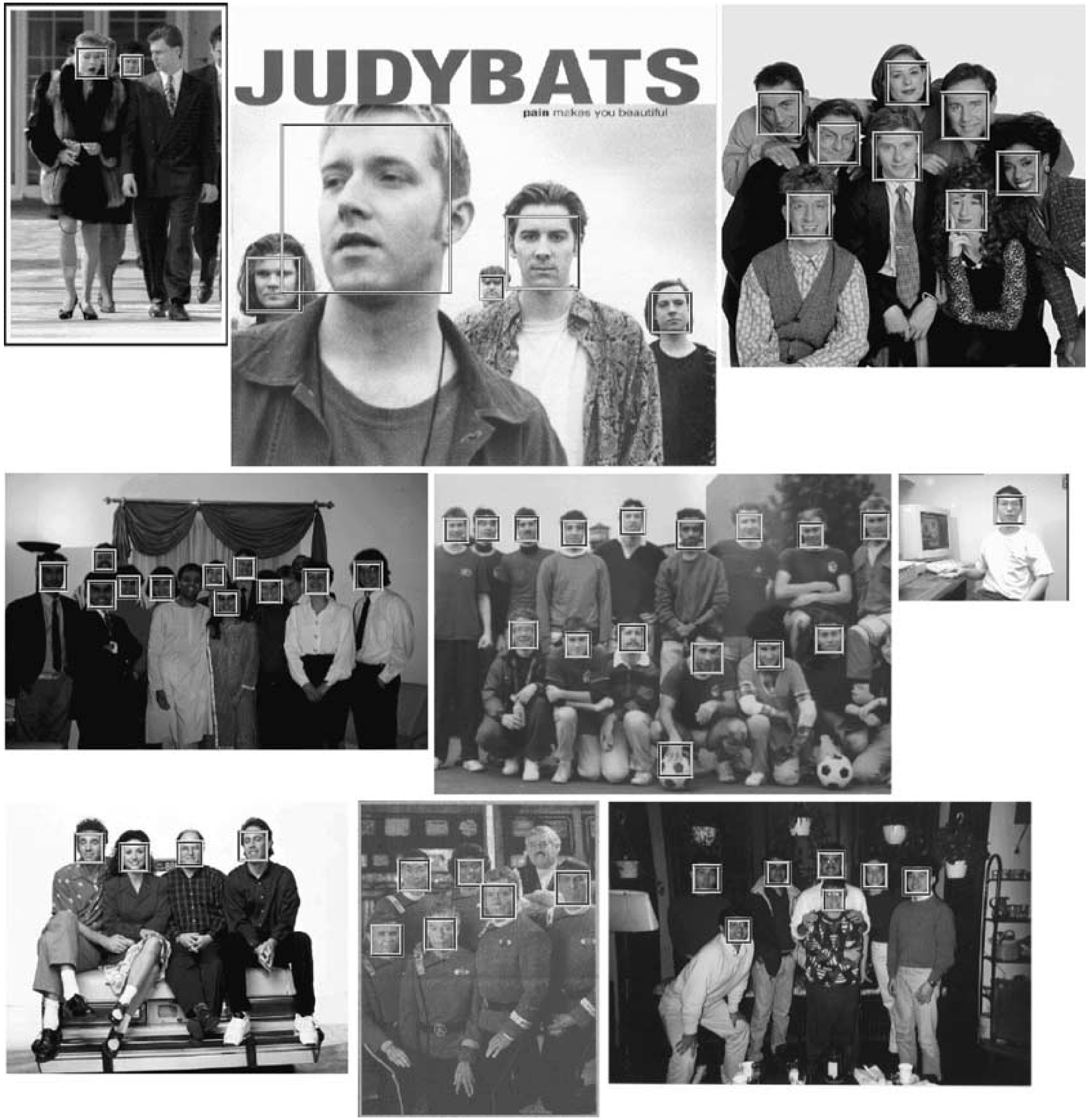
- Most sub-windows are negatively classified
- Cascade of classifiers
- *Reject as early as possible*
- Positive instance will go thought the entire cascade



Attentional Cascade

- Each node is trained with the false positives of the prior classifier
- Fast classification
- Final detector has 38 layers in the cascade, 6060 features




JUDYBATS
pain makes you beautiful

Weather Forecast using AdaBoost

Donat Perler, MS thesis, D-INFK, ETHZ, 2006

Pilot study with MeteoSwiss to predict thunderstorms

Data for Weather Forecast

- **Analysis data:**
  Output of the weather models
- **Present weather fields:**
  Weather codes based on human observations
- **Lightning data:**
  From a lightning detection system

Features for Forecast

Aversion	Description
LAT	Latitude
LONG	Longitude
HEIGHT	Height above sea level
DATE	Day of year
TIME	Day time
PMSL	Mean sea level pressure
T⋆	Temperature on the {500, 700, 850, 950} hPa model reference level
WDIR⋆	Horizontal wind direction on the {500, 700, 850, 950} hPa model reference level
WV⋆	Horizontal wind velocity on the {500, 700, 850} hPa model reference level
VERTW⋆	Vertical wind velocity on the {500, 700, 850, 950} hPa model reference level
RELHUM⋆	Relative humidity on the {500, 700, 850} hPa model reference level
THETA⋆	Equivalent potential temperature on the {500, 700, 850, 950} hPa model reference level
TD850	Dew point temperature on the 850 hPa model reference level
WSHEARL	Vertical wind shear between surface and 3 km above surface
WSHEARM	Vertical wind shear between surface and 6 km above surface

Aversion	Description
WSHEARU	Vertical wind shear between 3 km and 6 km above surface
WDIRVAR	Variance of the wind direction between 950 hPa and 500 hPa model reference level
RG	Total precipitation
RK	Convective precipitation
PDIFF	Pressure difference between cloud base and top
TTOP	Temperature at cloud top
KOI	KO-index
TT	TT-index (Total Totals Index)
SWEAT	SWEAT-index (Severe WEAther Thread index)
SHOWI	Showalter-index
LI	Surface lifted index
DCI	DCI-index (Deep Convection Index)
CAPE	Convective available potential energy
ADEDO2	Adedokun_2
MOCON	Surface moisture flux convergence
MOCONI	Surface moisture flux convergence integrated over the lowest 100 hPa
ADTHE	Equivalent potential temperature advection
HSURF	Model height of the surface over sea level

Forecast Results

- The simple decision stump classifier yield the best results.
- Sensitivity to thunderstorm detection is tuned with the initial weights in the first iteration of boosting.

Classifier	POD	FAR	FBI	CSI	HSS
DWD's expert system	18%	94%	3.12	0.05	0.08
Decision stumps	45%	68%	1.42	0.23	0.34
AdaBoost	57%	59%	1.44	0.32	0.46

POD: probability of detection, FAR: false alarm ratio, FBI: frequency bias,
CSI: critical success index, HSS: Heidke skill score

AdaBoost Demo in Matlab

http://www.mathworks.co.uk/matlabcentral/fileexchange/29245-boosting-demo