EE 6885 Statistical Pattern Recognition

Fall 2005
Prof. Shih-Fu Chang
http://www.ee.columbia.edu/~sfchang

Lecture 17 (11/23/05)

- Today's lecture
  - Application of AdaBoost in Face Detection
    - DHS Chap. 9.5
    - Paul Viola and Michael Jones, "Rapid object detection using a boosted cascade of simple features," CVPR, 2001.
  - Classifier Combination
    - DHS Chap. 9.7
    - R. Yan, J. Yang, and A. Hauptmann, "Learning Class-Dependent Weights in Automatic Video Retrieval," ACM Multimedia 2004
- Homework #7 due Nov. 30th
- Final Exam
  - Dec. 16th Friday 1:10-3 pm, Mudd Rm 644

Bias vs. variance for estimator

Assume F is a quantity whose value is to be estimated

data source → **Randomly draw n samples** $D = \{x_1, x_2 \ldots, x_n\}$ → **Learn** g_D **to estimate F**

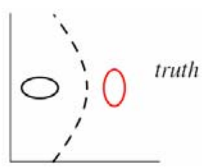
Repeat multiple times

$$\text{expected estimation error:} \quad E_D\left[\left|g_D - F\right|^2\right]$$

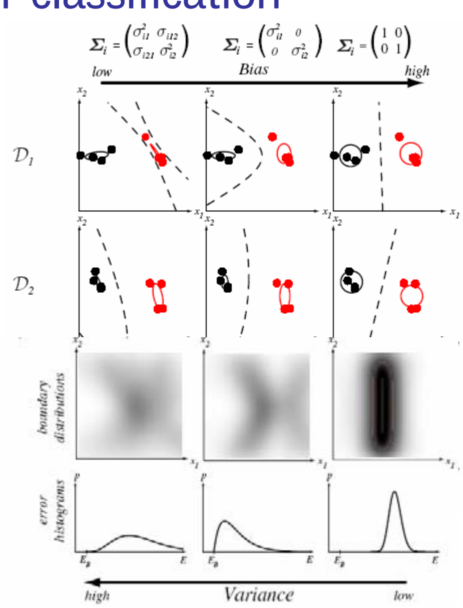
$$= \underbrace{\left[E_D(g_D) - F\right]^2}_{\textbf{Bias}^2} + \underbrace{E_D\left[\left|g_D - E_D(g_D)\right|^2\right]}_{\textbf{Variance}}$$

Bias vs. variance for classification

- **Ground truth: 2D Gaussian**

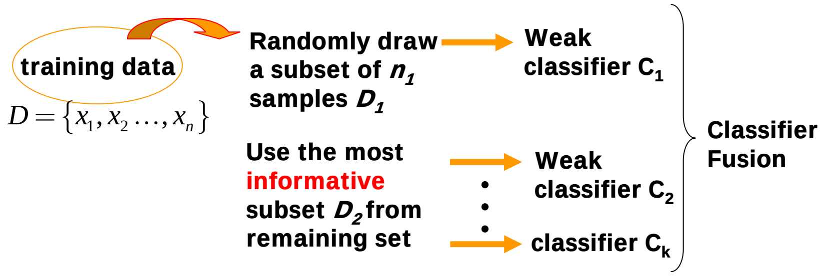


- **Complex models have smaller biases, more variances than simple models**
- **Increasing training pool size helps reduce the variance**
- **Occam's Razor principle**



Boosting

- **For each component classifier, use the subset of data that is most informative given the current set of component classifiers**



AdaBoost (Freund and Shapire)

- **Add weak classifiers until low training error has been achieved**
- **Each training pattern receives a weight determining its chance of being selected for subsequent learning steps.**
- **If a pattern is correctly classified, then the weight is decreased.**

begin with $W(i) = 1/n, \quad i = 1, \ldots, n$

$k = 1, \ldots, k_{\max}$

train weak classifier C_k using D training patterns with weights $W_k(i)$

$E_k \leftarrow$ training error of C_k measured on D using weights $W_k(i)$

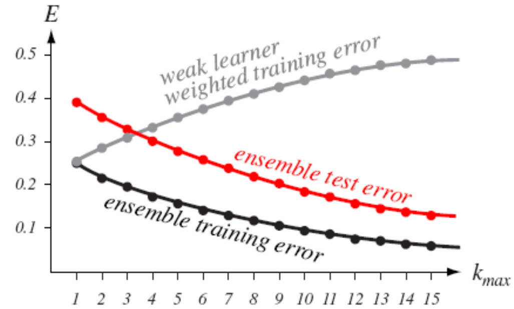
$$\alpha_k \leftarrow \frac{1}{2} \ln[(1 - E_k)/E_k]$$

$$W_{k+1}(i) \leftarrow \frac{W_k(i)}{Z_k} \times \begin{cases} e^{-\alpha_k} & \text{,if } x_i \text{ is correctly classified} \\ e^{\alpha_k} & \text{,if } x_i \text{ is incorrectly classified} \end{cases}$$

final classification rule

$$g(\mathbf{x}) = \sum_{k=1}^{k_{\max}} \alpha_k h_k(\mathbf{x}) > 0, \quad \text{where } h_k(\mathbf{x}) \text{ is the predicted output } \{\pm 1\} \text{ from } C_k$$

AdaBoost

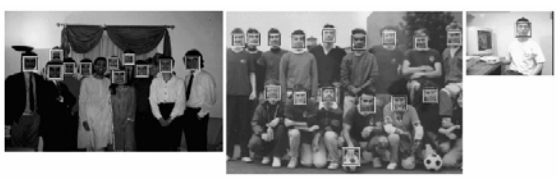


ensemble training error rate

$$E = \prod_{k=1}^{k_{max}} \left[2\sqrt{E_k(1-E_k)} \right]$$

- **It can shown that AdaBoost can maximize "margin" rapidly in iterations and thus has good generalization performance over test data.**

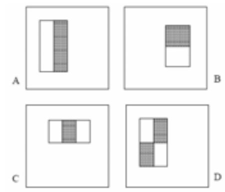
AdaBoost Face Detection (Viola and Jones, CVPR 2001)



- Rapid face detection for security and HCI applications
  - 2001 performance:
    - 384x288 images 15 frames per second
    - 2 frames per second on iPaq (200MIPS)
- Main contributions
  - New image representation: integral image
    - Allow rapid computation of Harr like filter responses
  - AdaBoost learning for feature selection
    - In each iteration, choose one weak classifier based on one feature only
  - Combine complex classifiers in a cascade way to discard non-interesting regions quickly

Harr filter like features

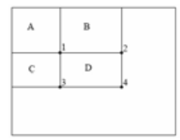
- **Pros and cons?**
- **Very simple rectangle difference features**
- **Sum of the pixels in the white area is subtracted from the sum in the grey area**
- **Number of rectangles can be increased as needed**



Rapid computation

$$ii(x,y) = \sum_{x' \le x, y' \le y} i(x', y')$$

- **Compute integral image in one pass**
- **Rectangle sum can be quickly computed**



- **A very large number of features:**
  - For each 24x24 detection region, there are more than 180,000 features
- **Each feature as a weak classifier**

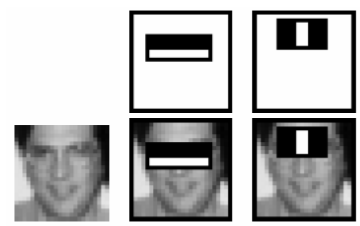
$$h_j(x) = \begin{cases} 1 & , \text{ if } f_j(x) > \text{ or } < \theta_j \\ 0 & otherwise \end{cases}$$

X is a 24x24 subimage, $f_j(x)$ is feature

- Image processing
  - Each subimage is variance normalized to avoid lighting variation
- Training:
  - 4916 face images scaled and aligned to 24x24 pixels, plus their vertical mirror images
  - 10,000 subimages from 9544 non-face images
- Detect faces at multiple scales, with a factor of 1.25 apart, and multiple overlapped scanning locations

AdaBoost Learning

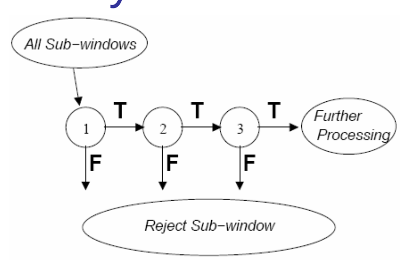
- **The first two features after feature selection**



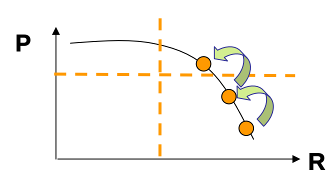
- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots,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

Cascade classifier for efficiency

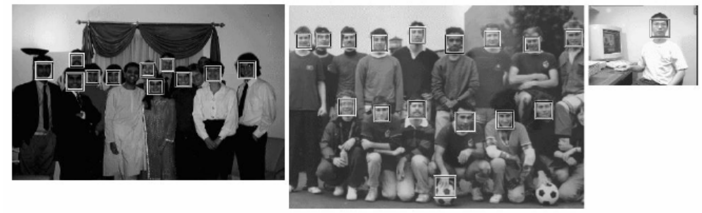
- **Break a large classifier into cascade of smaller classifiers**
  - **E.g., 200 features to {1, 10, 25, 50, 50}**
- **Adjust threshold in early stage so that it rejects unlikely regions quickly**
- **The latter stages are more difficult. They are trained using only the images passing the early stages.**
- **The final detector has 38 stages over 6000 features**
- **On average each sunimage uses 10 features**



- **Design tradeoffs**
  - **Number of features in each classifier**
  - **Threshold uses in each classifier**
  - **Number of classifiers**
- **Add stages until objective in P-R is met**



Performance over MIT-CMU data set

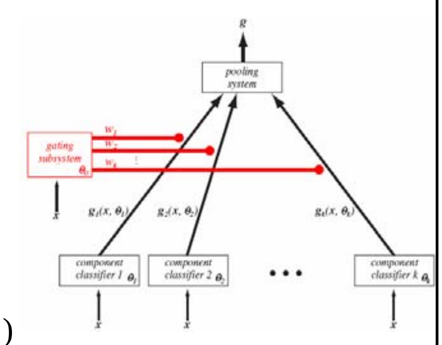


Detector \ False detections	10	31	50	65	78	95	167
Viola-Jones	76.1%	88.4%	91.4%	92.0%	92.1%	92.9%	93.9%
Viola-Jones (voting)	81.1%	89.7%	92.1%	93.1%	93.1%	93.2 %	93.7%
Rowley-Baluja-Kanade	83.2%	86.0%	-	-	-	89.2%	90.1%
Schneiderman-Kanade	-	-	-	94.4%	-	-	-
Roth-Yang-Ahuja	-	-	-	-	(94.8%)	-	-

- **Voting by multiple classifiers (learned from the same method) helps slightly**

Mixture of Experts

- Each component classifier is treated as an expert
- The predictions from each expert are pooled and fused by a gating subsystem



$$P(\mathbf{y}\,|\,\mathbf{x},\Theta) = \sum_{r=1}^{k} P(r\,|\,\mathbf{x},\theta_0)P(\mathbf{y}\,|\,\mathbf{x},\theta_r)$$

where $\mathbf{x}$ is the input pattern, $\mathbf{y}$ is the output

θ_0 controls the gating system; θ_r is parameter for component classifier r

- **How to determine** $P(r\,|\,\mathbf{x},\theta_0)$ **, i.e., mixture priors?**
- **Maximize data likelihood** $l(D,\Theta) = \sum_i \ln \sum_{r=1}^{k} P(r\,|\,\mathbf{x}^i,\theta_0)P(\mathbf{y}^i\,|\,\mathbf{x}^i,\theta_r)$
  - **gradient decent or EM**

Converting output from component classifiers

- **Convert various output formats to Prob(detecton or relevance)**
- **Rank order** $g_r = 1 - rank / N$
- **Binary label {1,0}**

$$g_r = \begin{cases} 1-\varepsilon & \text{if label} = 1 \\ \varepsilon & \text{if label} = 0 \end{cases}$$

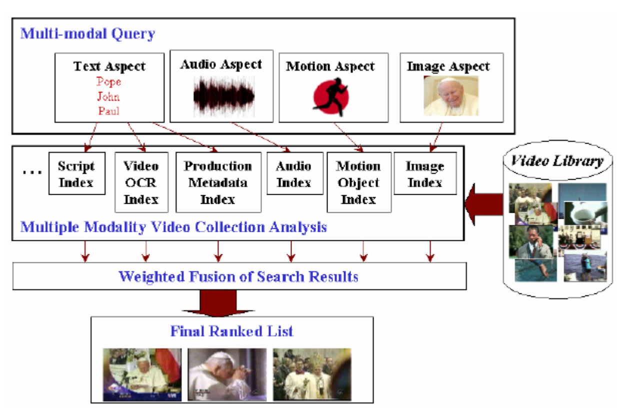
- **Multi-category discriminant values to detect a specific category**

$$\text{softmax} \quad g_r = e^{g_{r,j}} / \sum_{j=1}^{C} e^{g_{r,j}}$$

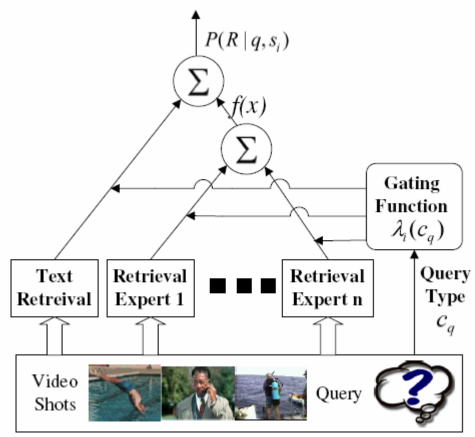
Mixture of Experts for Video Retrieval

(Yan, Yang, & Hauptmann 2004)

- Need to fuse retrieval results from tools using different modalities (text, image, concept etc)



- **A two-level mixture model**
  - Text-based search dominates most of times
  - Use audio-visual tools to refine text-based results
  - Group non-text tools under one level to avoid performance deterioration



EE6887-Chang

- **Training data:**
  - **a set of queries {q}, and the relevance labels of each document {s_i}**
- **First level likelihood** $P(R|q,s_i) = \lambda_1^u(c_q)P_1^u(R|q,s_i) + \lambda_2^u(c_q)P_2^u(R|q,s_i)$
- **Second level likelihood** $P_2^u(R|q,s_i) = f\left(\sum_{k=1}^{m} \lambda_k^l(c_q)P_k^l(R|q,s_i)\right)$
- **Total data likelihood**

$$l(\lambda^u;X) \quad = \quad \sum_i \log \sum_{t=1,2} \lambda_t^u P_t^u(R|q,s_i)$$

- **Use E-M to estimate** λ_t^u and λ_k^l

$$Q(\lambda^u;\lambda_j^u) = \sum_{t=1,2}\sum_i h_{it}\left(\log \lambda_t^u + \log P_t^u(R|q,s_i)\right)$$

h_{it} is the posterior porb. that document s_i is generated by classifier t

EE6887-Chang

E-M for Mixture of Expert for retrieval

Input: $P_t^u(R|q, s_i)$, t=1,2, and $y_i \in \{-1, +1\}$
Output: $\sum_{t=1}^{2} \lambda_t P_t^u(R|q, s_i)$ which optimizes $l(\lambda; X)$.
Algorithm:
Initialize $\lambda_i^{(0)}$ such that $\forall i, 0 < \lambda_i^{(0)} < 1, \sum_i \lambda_i^{(0)} = 1$
For $j = 1, 2, \ldots$

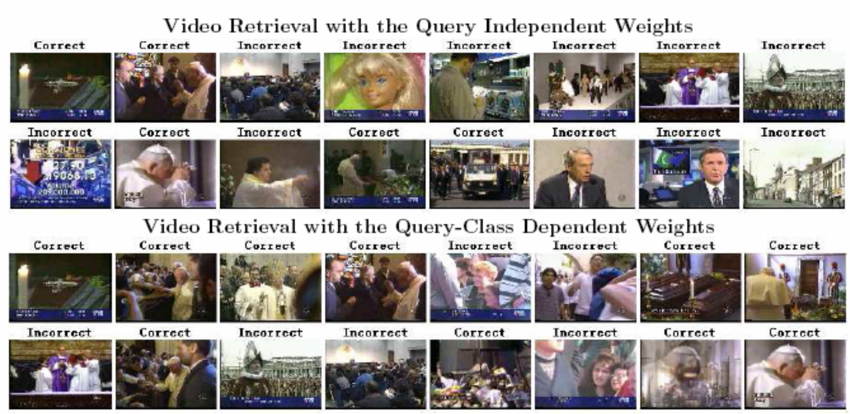
1. E-step: Compute expectation
$$h_{it}^{(j)} = \frac{\lambda_t^{(j)} P_t(R|q, s_i)}{\sum_t \lambda_t^{(j)} P_t(R|q, s_i)}$$
2. M-step: Update parameter $\lambda_t^{(j+1)} = \frac{1}{n} \sum_i h_{it}^{(j)}$
3. M-step: Maximize the weighted log-likelihood in (7)
4. Check convergence if $|l(\lambda^{(j+1)}; X) - l(\lambda^{(j)}; X) < \epsilon|$

	P_1	P_2	h
s_1	0.3	0.1	1
s_2	0.2	0.4	1
s_3	0.6	0.5	2
s_4	0.5	0.8	2

- **h is the hidden variable indicating the responsible expert**
- **E-step: compute h**
- **M-step: find** λ

EE6887-Chang 17-19

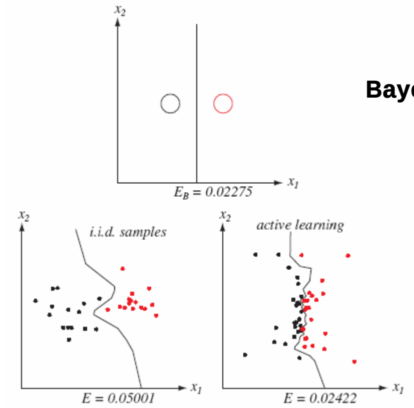
Example of EM learning of weights



EE6887-Chang 17-20

Active Learning (Learning with Queries)

- Actively select the next training data that are most informative
  - one that is closest to the decision boundary
  - (or) one that has the most ambiguous confidence scores (e.g., similar discriminant values from two classes)

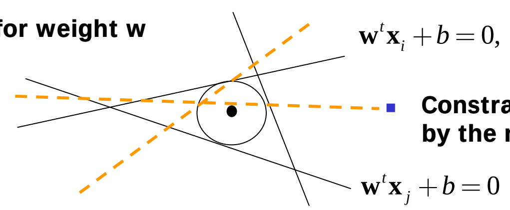


Bayesian boundary

Active selection of training data
Nearest neighbor classifiers
Start with the far points in the space

Applications (Active SVM)

- Space for weight w



- Constraint added by the new data

- In image retrieval
  - first train a SVM from labeled data
  - now in interactive retrieval
  - select a new sample and present it to user
  - user label the new data
  - use the new label to re-train the weight w
  - which sample to choose?

- Choose the un-labeled sample that is closest to the current separation plane. Why?