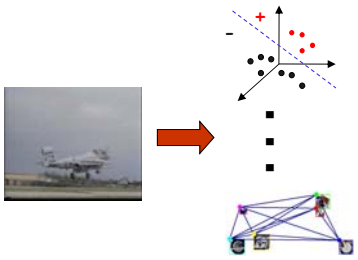
EE 6882
Visual Search Engine

Prof. Shih-Fu Chang, Feb. 20th 2012
Lecture #5

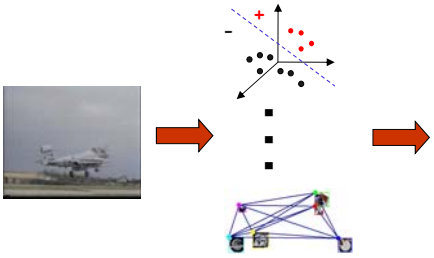
- Image Classification, Active Learning, Boosting

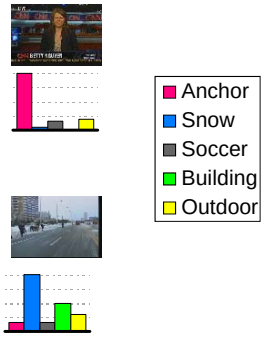


Auto Image Tagging May Help Fill the Gap

- Audio-visual features
- User social features
- Camera/location info

- Rich semantic description based on content recognition



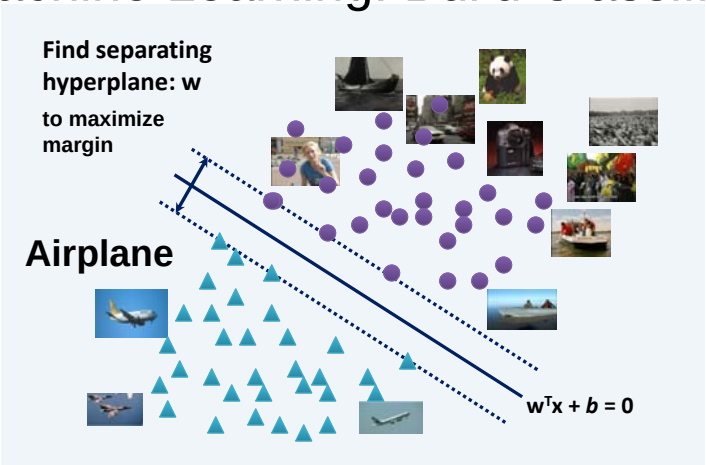


Classification

- Binary Classification
  - "indoor" vs. "outdoor"
  - "people" vs. "no people"
- Multi-class Classification
  - Assign a test sample to one of multiple possible categories, e.g., CalTech 101, 256
- Multi-label classification
  - Given an image, generate all relevant labels

3

Machine Learning: Build Classifier



Decision function: $f(x) = \text{sign}(w^T x + b)$

$w^T x_i + b > 0$ if label $y_i = +1$

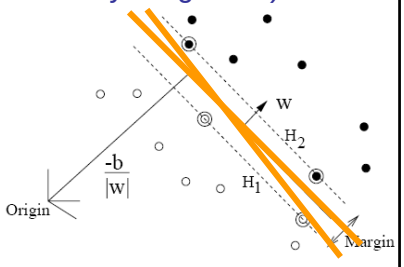
$w^T x_i + b < 0$ if label $y_i = -1$

Support Vector Machine (tutorial by Burges '98)

- **Look for separation plane with the highest margin**

  Decision boundary

  $H_0: \mathbf{w}^t\mathbf{x} + b = 0$



- **Linearly separable**

  $\mathbf{w}^T\mathbf{x}_i + b > 1$ if label $y_i = +1$

  $\mathbf{w}^T\mathbf{x}_i + b < -1$ if label $y_i = -1$

  $y_i(\mathbf{w}^T\mathbf{x}_i + b) > 1$ for all x_i

- **Two parallel hyperplanes defining the margin**

  hyperplane $H_1(H_+): \mathbf{w}^t\mathbf{x}_i + b = +1$

  hyperplane $H_2(H_-): \mathbf{w}^t\mathbf{x}_i + b = -1$

- **Margin: sum of distances of the closest points to the separation plane**

  $\text{margin} = 2/\|\mathbf{w}\|$

- **Best plane defined by w and b**

Finding the maximal margin (separable case)

$$\text{minimize } \frac{1}{2}\|\mathbf{w}\|^2 \qquad y_i(\boldsymbol{w^t x_i} + b) - 1 \geq 0, i = 1, \ldots, l$$

Primal Problem

minimize L_p *w.r.t.* $\mathbf{w}$ and b

$$L_p = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{l} \alpha_i (y_i(\boldsymbol{w^t x_i} + b) - 1)$$

$\alpha_i \geq 0$

$$\frac{dL_p}{dw} = 0 \longrightarrow w = \sum_{i=1}^{l} \alpha_i y_i \boldsymbol{x_i}$$

$$\frac{dL_p}{db} = 0 \longrightarrow \sum_{i=1}^{l} \alpha_i y_i = 0$$

Dual Problem

maximize L_D w.r.t α

$$L_D = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \boldsymbol{x_i}^T \boldsymbol{x_j}$$

$s.t.$

$\alpha_i \geq 0$

$$\sum_{i=1}^{l} \alpha_i y_i = 0$$

- **Prime and Dual have the same solutions of w and b**
- **Solved by quadratic programming**

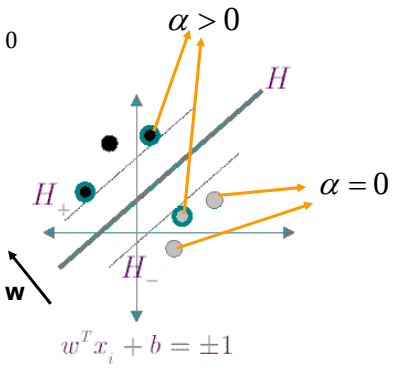
Max Margin Solution for separable case

$$w = \sum_{i=1}^{l} \alpha_i y_i \boldsymbol{x_i} \qquad \alpha_i \geq 0 \qquad \sum_{i=1}^{l} \alpha_i y_i = 0$$

$$y_i(\boldsymbol{w^t x_i} + b) - 1 \geq 0, \qquad i = 1, \ldots, l$$

$$\alpha_i(y_i(\boldsymbol{w^t x_i} + b) - 1) = 0, \qquad i = 1, \ldots, l$$

- **Weight sum from positive class = Weight sum from negative class**
- **Direction of w: roughly from negative support vectors to positive ones**



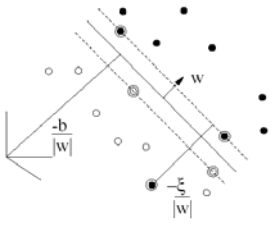
if $a_i > 0$, $\mathbf{x}_i$ is on H_+ or H_- and is a support vector

- **How to compute w and b?**
- **How to classify new data?**

Non-separable

- **Add slack variables** ξ_i

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq +1 - \xi_i \quad \text{for } y_i = +1$$
$$\mathbf{x}_i \cdot \mathbf{w} + b \leq -1 + \xi_i \quad \text{for } y_i = -1$$
$$\xi_i \geq 0 \ \forall i.$$



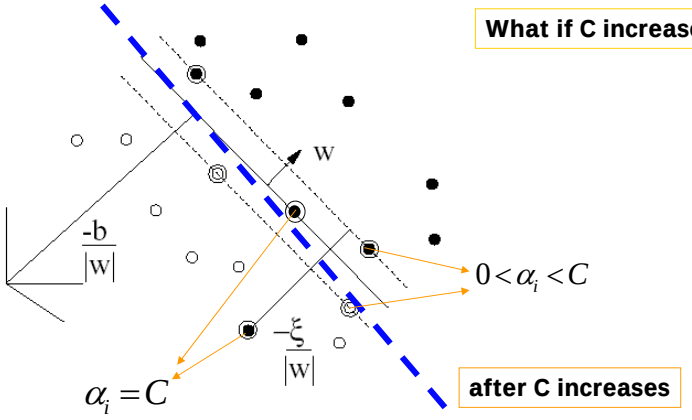
if $\xi_i > 1$, then $\mathbf{x}_i$ is misclassified (i.e. training error)

Lagrange multiplier:

$$L_P = \underbrace{\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i}_{\text{New objective function}} - \sum_i \alpha_i\{y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i\} - \sum_i \mu_i \xi_i$$

Ensure positivity

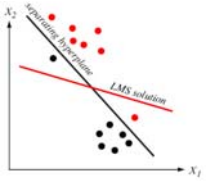
- **All the points located in the margin gap or the wrong side will get** $\alpha_i = C$



- **When C increases, samples with errors get more weights**
  - **better training accuracy, but smaller margin**
  - **less generalization performance**

Generalized Linear Discriminant Functions

- **What if linear boundaries not suitable**
- **Try to use nonlinear terms**



$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d}\sum_{j=1}^{d} w_{ij} x_i x_j$$

$$g(\mathbf{x}) = \sum_{i=1}^{d} a_i y_i(\mathbf{x})$$

- **Example** $y = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}$ $g(\mathbf{x}) = a_1 + a_2 x + a_3 x^2$

- **Data become separable in higher-dimensional space**

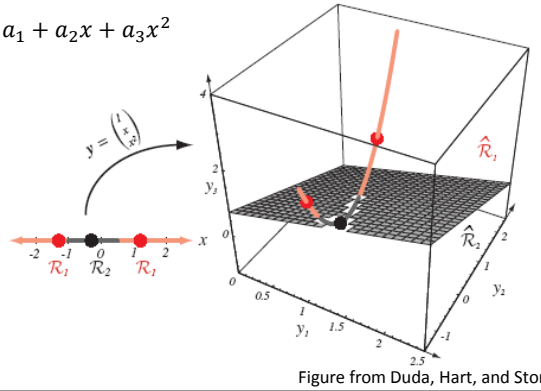
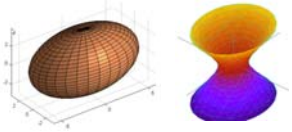


Figure from Duda, Hart, and Stork

Generalized Linear Discriminant Functions

- **Example** $\mathrm{y}=\begin{bmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{bmatrix}$ $\mathrm{g}(\mathbf{x}) = a_1 x_1 + a_2 x_2 + a_3 x_1 x_2$

- **Shape of decision boundary**
  - **ellipsoid, hyperboloid, etc.**



- **Data become separable in higher-dimensional space**
- **Learn maximal margin classifier**

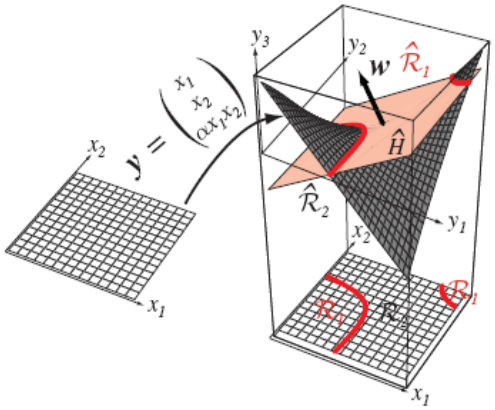


Figure from Duda, Hart, and Stork

Non-Linear Space

$\Phi : \mathbf{R}^d \mapsto \mathcal{H}.$ Map to a high dimensional space, to make the data separable $e.g., \Phi(x)=\begin{bmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{bmatrix}$

- **Find the SVM in the high-dim space (embedding space)**

$$\mathrm{g}(\mathbf{x}) = \underbrace{\sum_{i=1}^{N_s} \alpha_i y_i \Phi(\mathbf{s_i})}_{\mathbf{w}} \times \Phi(\mathbf{x}) + \mathrm{b}$$

- **Luckily, we don't have to find** $\Phi(s_i)$
- **Instead, we define kernel**

$$K(\mathbf{s_i}, \mathbf{x}) = \Phi(\mathbf{s_i}) \times \Phi(\mathrm{x})$$

$$\mathrm{g}(\mathbf{x}) = \sum_{i=1}^{N_s} \alpha_i y_i K(\mathbf{s_i}, \mathbf{x}) + \mathrm{b}$$

- **We can use the same method to maximize L_D to find** a_i

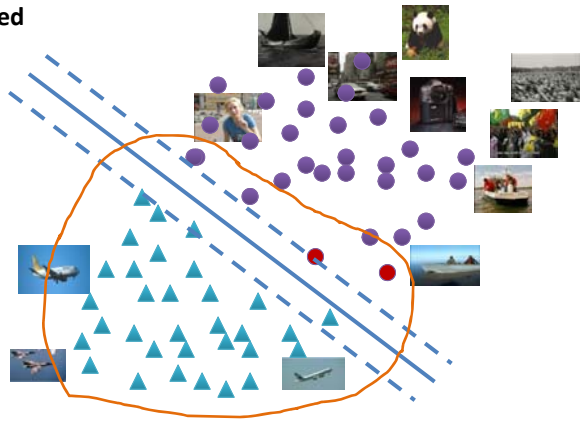
$$L_D = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \boldsymbol{K}(\boldsymbol{x_i}, \boldsymbol{x_j})$$

Some popular kernels

$k(x,y) = x^T y + c$	**Linear Kernel**
$k(x,y) = (\alpha x^T y + c)^d$	**Polynomial Kernel**
$k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$	**Gaussian Kernel**
$k(x,y) = \exp\left(-\gamma\|x-y\|^2\right)$	
$k(x,y) = \exp\left(-\frac{\|x-y\|}{2\sigma^2}\right)$	**Exponential Kernel (Radial Basis Function)**
$k(x,y) = \tanh(\alpha x^T y + c)$	**Sigmoid (Neural Net) Kernel**
$k(x,y) = \sum_{i=1}^{n} \min(x_i, y_i)$	**Histogram Intersection Kernel**
$k(x,y) = 1 - \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\frac{1}{2}(x_i + y_i)}$	**Chi Square Kernel**

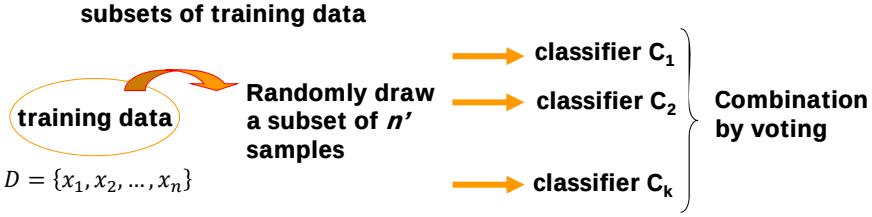
How to handle a large number of negative samples?

- **randomly subsample negative training data**
- **learn SVM (M1)**
- **apply M1 to additional training samples**
- **add "hard" samples and train SVM again**
- **repeat until model unchanged**



Bagging

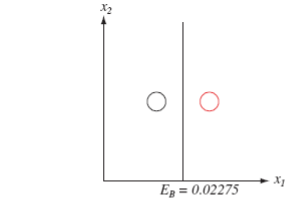
- Classifiers are affected by the choices of training set
  - train multiple component classifiers by creating multiple subsets of training data



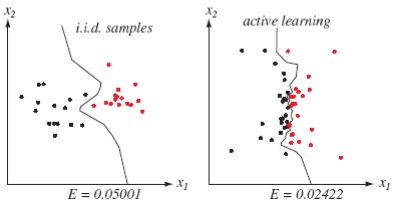
- When does this help?
- Issues
  - *n'*
  - sampling scheme
  - number of component classifiers
  - fusing method

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)

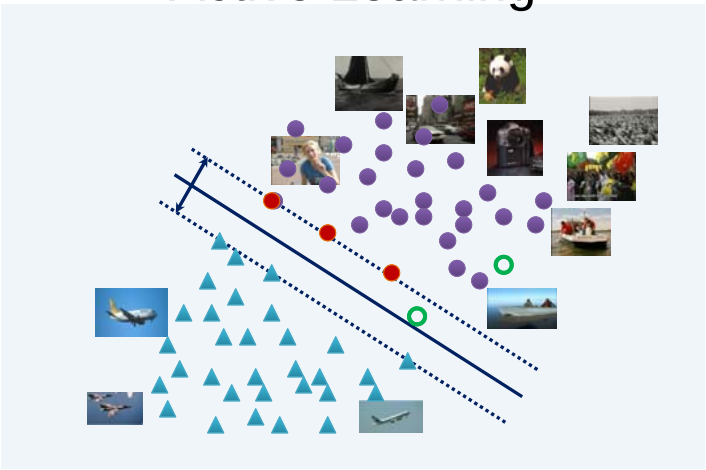


Bayesian boundary



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

Active Learning

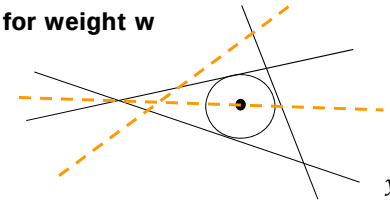


Which sample is to ask user to label?

Find sample closest to decision boundary

Applications (Active SVM)

- **Space for weight w**



$$y_i(\boldsymbol{w}^t\boldsymbol{x_i} + b) - 1 \geq 0$$

- Constraint added by the new data
- Which one is better?

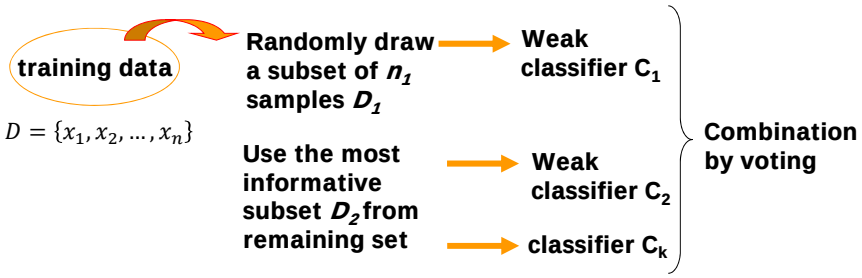
$$y_j(\boldsymbol{w}^t\boldsymbol{x_j} + b) - 1 \geq 0$$

- **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?**

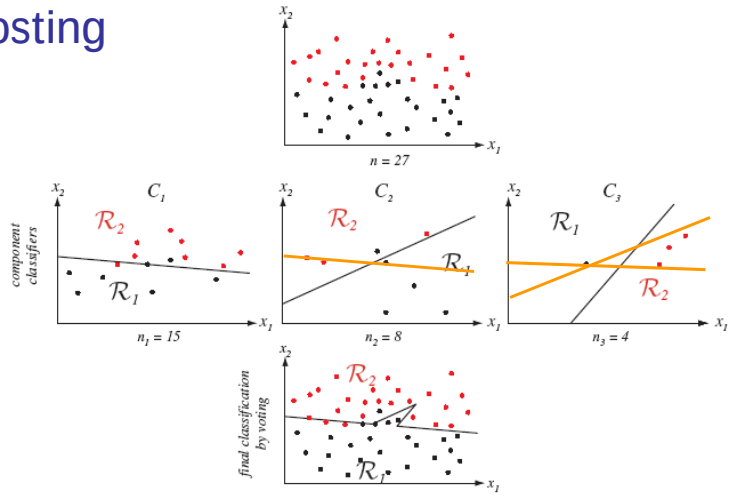
Boosting

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



  - What's the most informative subset for C_2?
    - only data that are misclassified by C_1?
    - (OR) half correctly classified and half misclassified?
  - What's the most informative subset for C_3?
    - include only points that C1 and C2 cannot agree, use C3 as tie breaker

Boosting



AdaBoost

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

$$W(i) = \frac{1}{n}, \qquad i = 1, \ldots, n$$

$$train\ weak\ classifier\ C_k\ using\ data\ weight\ W_k(i) \qquad k = 1, \ldots, k_{max}$$

$$E_k: traing\ error\ of\ C_k \qquad a_k = \frac{1}{2} ln\left[{(1 - E_k)}/{E_k}\right]$$

$$W_{k+1}(i) = \frac{W_k(i)}{Z_k} \cdot \begin{Bmatrix} e^{-a_k}, if\ x_i\ is\ correctly\ labeled \\ e^{a_k}, if\ x_i\ is\ incorrectly\ labeled \end{Bmatrix}$$

final classification

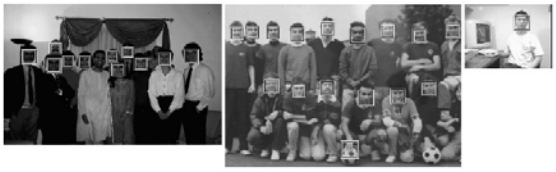
$g(\mathbf{x}) = \sum_{k=1}^{k_{\max}} a_k h_k(\mathbf{x})$, where $h_k(\mathbf{x})$ is the predicted output from C_k

AdaBoost

Combined classier training error rate $E = \prod_{k=1}^{k_{max}} \sqrt{E_k(1 - E_k)}$

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

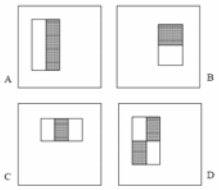
AdaBoost Face Detection (Viola and Jones, CVPR 2001)



- 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**

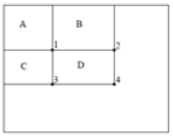


Rapid computation

- **Compute integral image in one pass**

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

- **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**

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

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

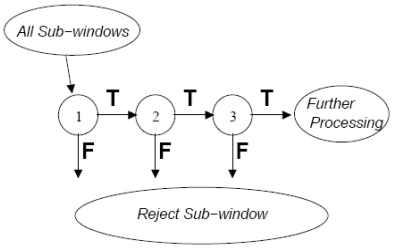
AdaBoost Learning

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

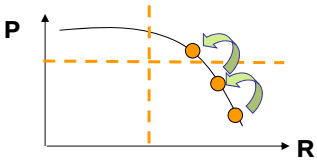
- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots,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

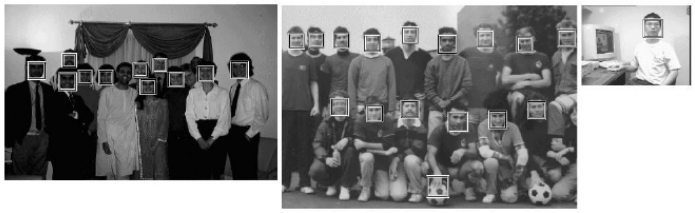
- Break a large classifier into cascade of smaller classifiers
  - E.g., 200 features to {1, 10, 25, 50, 50}
- Adjust threshold in early stage so that it rejects unlikely regions quickly
- The latter stages are more difficult. They are trained using only the images passing the early stages.
- The final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

Reading List

- Lazebnik, S., C. Schmid, and J. Ponce. *Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. in IEEE CVPR. 2006.*
- Jiang, Y., C. Ngo, and J. Yang. *Towards optimal bag-of-features for object categorization and semantic video retrieval. in ACM CIVR. 2007.*
- Chang, S., et al. *Columbia University/VIREO-CityU/IRIT TRECVID2008 high-level feature extraction and interactive video search. in NIST TRECVID Workshop. 2008.*
- Jiang, Y., et al. *Columbia-UCF TRECVID2010 Multimedia Event Detection: Combining Multiple Modalities, Contextual Concepts, and Temporal Matching. in TRECVID Workshop. 2010.*
- *Pattern Classification*, 2nd ed., Richard O. Duda, Peter E. Hart, and David G. Stork ISBN: 0-471-05669-3, 2000, Wiley
- Viola, P. and M. Jones. *Rapid Object Detection using a Boosted Cascade of Simple Features. in Proceedings IEEE Conf. on Computer Vision and Pattern Recognition. 2001.*
- Yan, R., J. Yang, and A.G. Hauptmann. *Learning Query-Class Dependent Weights in Automatic Video Retrieval. in ACM Multimedia. 2004. New York.*