EE 6885 Statistical Pattern Recognition

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

Lecture 15 (11/16/05)

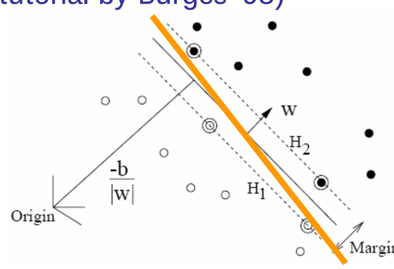
- Today's lecture
  - SVM with kernel, error bounds
  - Paper:
    Christopher J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," Data Mining and Knowledge Discovery 2, 121-167, 1998.
- Guest Lecture: Application of SVM in video stream concept detection
- Project data and guideline announced. Due Dec 12.
- Next Lecture: Nov. 21st Monday, Long lecture, starts at 12:20pm
- Final Exam
  - Dec. 16th Friday 1:10-4pm, Mudd Rm 644

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**

$\mathbf{w}^t\mathbf{x}_i + b \geq +1 \quad \forall \mathbf{x}_i$ in class ω_1 i.e. $y_i = +1$

$\mathbf{w}^t\mathbf{x}_i + b \leq -1 \quad \forall \mathbf{x}_i$ in class ω_2 i.e. $y_i = -1$

Inequality constraints : $y_i(\mathbf{w}^t\mathbf{x}_i + b) - 1 \geq 0 \ , \ \forall \ i$

- **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**

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

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

KKT conditions (iff) for separable case

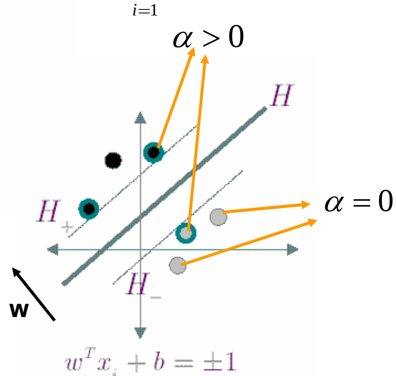
$$\frac{\partial}{\partial w_\nu} L_P = w_\nu - \sum_i \alpha_i y_i x_{i\nu} = 0 \qquad \nu = 1, \cdots, d \quad \longrightarrow \quad \mathbf{w}^* = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial}{\partial b} L_P = -\sum_i \alpha_i y_i = 0$$

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0 \qquad i = 1, \cdots, l$$

$$\alpha_i \geq 0 \qquad \forall i$$

$$\alpha_i(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1) = 0 \qquad \forall i$$



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

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

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

Non-separable

- **Add slack variables** ξ_i

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

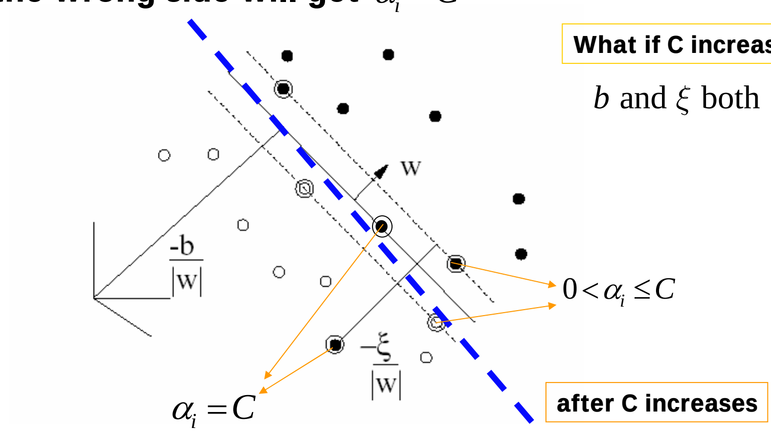
if $\xi_i > 1$, then $\mathbf{x}_i$ is misclassified (i.e. training error)

Lagrange multiplier: minimize

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

Ensure positivity

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

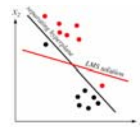


What if C increases?

b and ξ both $\downarrow$

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

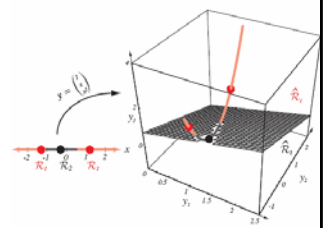
Generalized Linear Discriminant Functions



- **Include more than just the linear terms**

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

- **Shape of decision boundary**
  - **ellipsoid, hyperhyperboloid, lines etc**

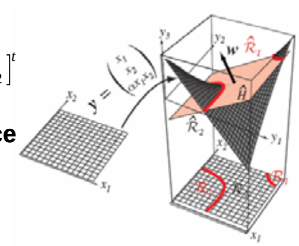


- **In general** $g(\mathbf{x}) = \sum_{i=1}^{\hat{d}} a_i y_i(\mathbf{x}) = \mathbf{a}^t \mathbf{y}$

- **Example**

$$g(x) = a_1 + a_2 x + a_3 x^2 \qquad g(x) = a_1 x_1 + a_2 x_2 + a_3 x_1 x_2$$

$$= [a_1 \quad a_2 \quad a_3][1 \quad x \quad x^2]^t \qquad = [a_1 \quad a_2 \quad a_3][x_1 \quad x_2 \quad x_1 x_2]^t$$



- **Data become separable in higher-dimensional space**
  - **learning parameters in high dimension is hard (curse of dim.)**
  - **instead, try to maximize margins → SVM**

Non-Linear Space

$\Phi : \mathbf{R}^d \mapsto \mathcal{H}$. Map to a high dimensional space, to make the data separable

$$\Phi(\mathbf{x}) = \begin{pmatrix} x_1^2 \\ \sqrt{2}\, x_1 x_2 \\ x_2^2 \end{pmatrix}$$

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

$$g(\mathbf{x}) = \underbrace{\sum_{i=1}^{N_s} \alpha_i y_i \Phi(\mathbf{s}_i)}_{\mathbf{w}} \cdot \Phi(\mathbf{x}) + b$$

- **Luckily, we don't have to find** $\Phi(\mathbf{s}_i)$ nor $\sum_{i=1}^{l} \alpha_i y_i \Phi(\mathbf{s}_i)$

- **Instead, we define kernel** $K(\mathbf{s}_i, \mathbf{x}) = \Phi(\mathbf{s}_i) \cdot \Phi(\mathbf{x})$

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

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

$$L_D = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$$

$$= \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

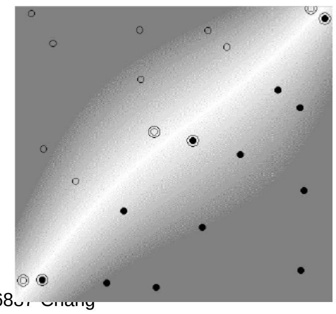
Some popular kernels

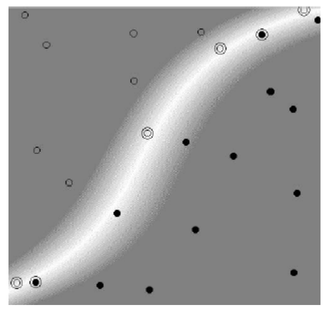
$K(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y}+1)^p$ **polynomial**

$K(\mathbf{x},\mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2}$ **Gaussian Radial Basis Function (RBF)**

$K(\mathbf{x},\mathbf{y}) = \tanh(\kappa\mathbf{x}\cdot\mathbf{y}-\delta)$ **sigmoidal neural network**

separable Cubic polynomial non-separable





Error bound based on VC dimension (Vapnik '95)

- **Risk: expectation of loss**

$$R(\alpha) = E\left[\frac{1}{2}\underbrace{|y_i - f(\mathbf{x}_i,\alpha)|}_{\text{loss}}\right],$$ where α is the classifier model parameter

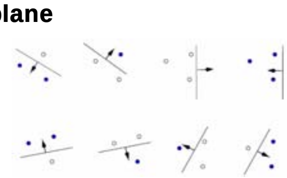
$R_{emp}(\alpha)$: empirical risk for a specific classifier over a training/test set

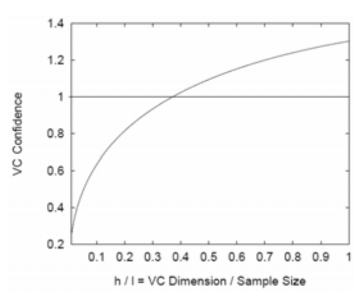
$$R(\alpha) \le R_{emp}(\alpha) + \underbrace{\sqrt{\left(\frac{h(\log(2l/h)+1)-\log(\eta/4)}{l}\right)}}_{\textbf{VC confidence}}$$ with probability $(1-\eta)$

h: VC dimension, capacity

l: size of the training set

- **VC dim of hyperplane in R^n is $(n+1)$**





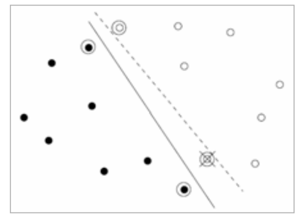
A tighter error bound of SVM

- **Leave one out rotation**

1 testing data

$(l-1)$ for training

rotate
data partition



- First, train a SVM over l samples
- In each rotation, re-train the SVM over the l-1 samples, test on the remaining data
- if the test sample is not SV, then SVM does not change and there is no error. Otherwise, there might be an error.

$$\boxed{E[P(error)] \le \frac{E[\text{\# of support vectors}]}{\text{number of training samples}} = \frac{E[\text{\# of support vectors}]}{(l\text{-}1)}}$$

$P(error)$: risk (expected test error) for a learned classifier trained on l-1 samples

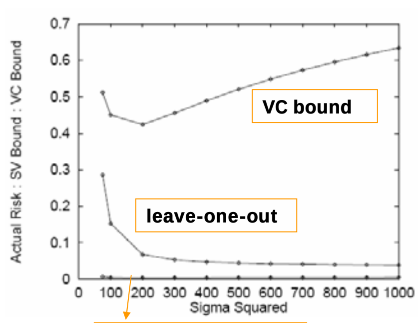
$E[P(error)]$: expected risk over all choices of training set of l-1 samples

E[# of s.v.]: expected # of s.v. over all choices of training set of size l

Potential use and issue of the error bound

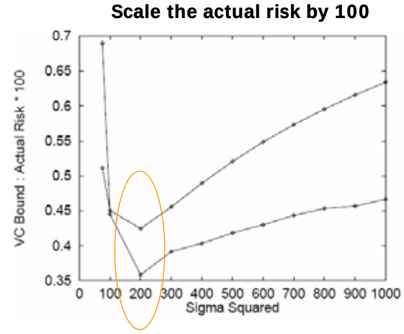
$K(\mathbf{x},\mathbf{y}) = e^{-(\mathbf{x}\text{-}\mathbf{y})^2/2\sigma^2}$

- **How to determine the best sigma value?**



Measured test error (actual risk)

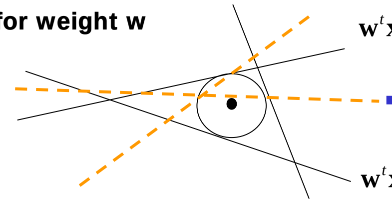
Scale the actual risk by 100



- **Leave-one-out bound is tighter than VC bound in this experiment (NIST digit classification)**
- **But VC bound ahs better predictive power for selecting a good classifier (machine)**

Applications (Active SVM)

- **Space for weight w**



- **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?**