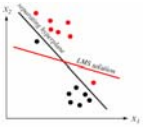
EE 6885 Statistical Pattern Recognition

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

Lecture 14 (11/02/05)

- Reading
  - DHS Chap. 5.11
  - Paper:
    Christopher J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," Data Mining and Knowledge Discovery 2, 121-167, 1998.
- Homework #6 assigned today
  - Due Nov. 16th
- Project data will be available this week
- Class schedules
  - No classes
    - 11/7 (M, Uni. Holiday), 11/9 (W), 11/14 (M)
  - Long lectures (start at 12:20pm)
    - 11/2 (today), 11/16 (W, next class), 11/21 (M)
  - Back to normal afterwards

Linear Discriminant Classifiers



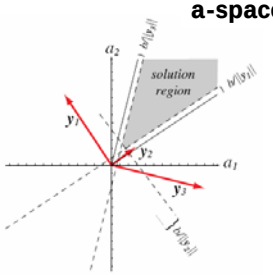
$g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + w_0 \quad \Rightarrow$ find weight $\mathbf{w}$ and bias w_o

- Augmented Vector $\mathbf{y} = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} w_0 \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix} \quad \Rightarrow g(\mathbf{x}) = g(\mathbf{y}) = \mathbf{a}^t\mathbf{y}$

map $\mathbf{y}$ to class ω_1 if $g(\mathbf{y})$>0, otherwise class ω_2

- Design Objective $\mathbf{a}^t\mathbf{y}_i$>b, $\forall\mathbf{y}_i$
  - Each y_i defines a half plane in the weight space (a).
  - Note we search weight solutions in the a-space.

a-space



Minimal Squared-Error Solution

Objective: $\mathbf{a}^t\mathbf{y}_i$=b, $\forall\mathbf{y}_i$

$\Rightarrow$ define $J_s = \sum_{i=1}^{n}(\mathbf{a}^t\mathbf{y}_i - b_i)^2$

$J_s = \|Y\mathbf{a} - \mathbf{b}\|^2 = (Y\mathbf{a} - \mathbf{b})^t(Y\mathbf{a} - \mathbf{b})$

$Y = \begin{bmatrix} \mathbf{y}_1^t \\ \mathbf{y}_2^t \\ \vdots \\ \mathbf{y}_n^t \end{bmatrix}$ **Training sample matrix**

dimension: n x (d+1)

$\nabla_{\mathbf{a}} J_s = 2Y^t(Y\mathbf{a} - \mathbf{b}) = 0$

if Y^tY is nonsingular $\Rightarrow \mathbf{a} = (Y^tY)^{-1}Y^t\mathbf{b} = Y^\dagger\mathbf{b}$ **pseudo-inverse**

- **Example**

training samples: *class* ω_1: $(1,2)^t, (2,0)^t$ *class* ω_2: $(3,1)^t, (2,3)^t$

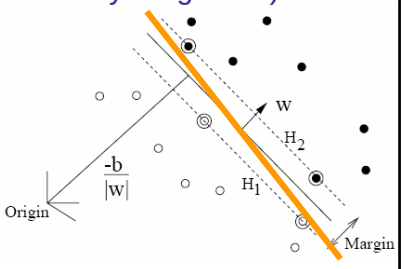
$Y = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 0 \\ -1 & -3 & -1 \\ -1 & -2 & -3 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ find $Y^\dagger$, then compute $\mathbf{a}^* = Y^\dagger\mathbf{b}$

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

- **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**

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

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

Finding the maximal margin

$$\text{minimize } \frac{1}{2}\|\mathbf{w}\|^2 \qquad \text{subject to inequality constraints}$$

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

- **Use the Lagrange multiplier technique for the constrained opt. problem**

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

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

$$\alpha_i\geq 0$$

$$\frac{dL_p}{d\mathbf{w}}=0 \ \Rightarrow\ \mathbf{w}=\sum_{i=1}^{l}\alpha_i y_i\mathbf{x}_i$$

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

maximize L_D *w.r.t.* $\mathbf{w}$ and b

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

with conditions :

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

$$\alpha_i\geq 0$$

Quadratic Programming

Primal Problem ⟷ **Dual Problem**

- **Prime and Dual have the same solutions of w and b**

KKT conditions (iff) for separable case

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

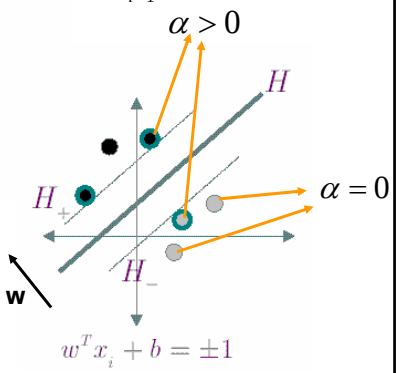
$$\frac{\partial}{\partial b} L_P = -\sum_i \alpha_i y_i = 0$$

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

$$\alpha_i \geq 0 \quad \forall i$$

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

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



if $\alpha_i > 0$, $\mathbf{x}_i$ 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: minimize

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

New objective function

Ensure positivity

KKT Conditions for non-separable Solutions

$$\frac{\partial L_P}{\partial w_\nu} = w_\nu - \sum_i \alpha_i y_i x_{i\nu} = 0 \qquad \frac{\partial L_P}{\partial \xi_i} = C - \alpha_i - \mu_i = 0$$

$$\frac{\partial L_P}{\partial b} = -\sum_i \alpha_i y_i = 0 \qquad y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i \geq 0$$

$$\xi_i \geq 0$$

$$\alpha_i \geq 0$$

$$\mu_i \geq 0$$

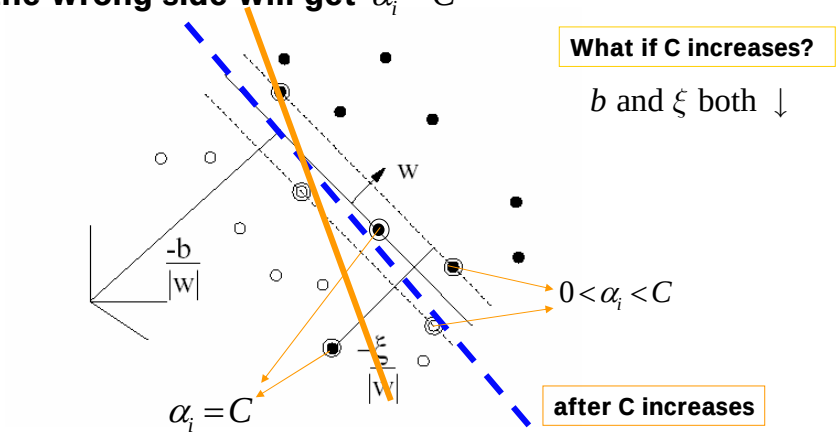
$$\alpha_i\{y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i\} = 0$$

$$\mu_i \xi_i = 0$$

If $0<\alpha_i < C$, then $\xi_i = 0 : \mathbf{x}_i$ is on H_1 or H_2

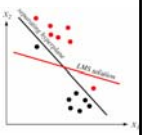
If $\alpha_i = C$,
then $\xi_i > 0 : \mathbf{x}_i$ is inside the margin region or on the wrong side
or $\xi_i = 0 : \mathbf{x}_i$ is on H_1 or H_2

- **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**

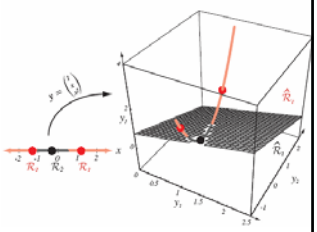
Generalized Linear Discriminant Functions



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

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

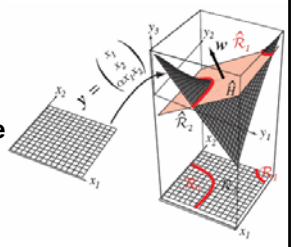
- **Shape of decision boundary**
  - **ellipsoid, hyperhyperboloid, lines etc**
- **In general** $g(\mathbf{x}) = \sum_{i=1}^{\hat{d}} a_i y_i(\mathbf{x}) = \mathbf{a}^t\mathbf{y}$



- **Example**

$$g(x) = a_1 + a_2 x + a_3 x^2 = [a_1 \quad a_2 \quad a_3][1 \quad x \quad x^2]^t$$

$$g(x) = a_1 x_1 + a_2 x_2 + a_3 x_1 x_2 = [a_1 \quad a_2 \quad a_3][1 \quad x_1 \quad x_1 x_2]^t$$



- **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)**

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

- **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**

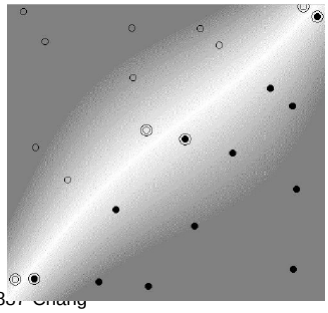
$K(\mathbf{x},\mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2}$ **Gaussian Radial Basis Function (RBF)**

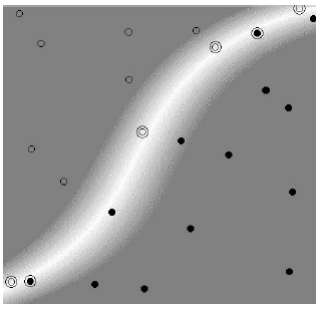
$K(\mathbf{x},\mathbf{y}) = \tanh(\kappa\mathbf{x}\cdot\mathbf{y}-\delta)$ **sigmoidal neural network**

separable

Cubic polynomial

non-separable





Error bound based on VC dimension (Vapnik '95)

- **Risk: expectation of loss**

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

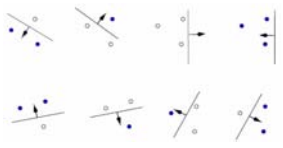
$R_{emp}(\alpha)$: empirical risk for a specific classifier over a training/test set

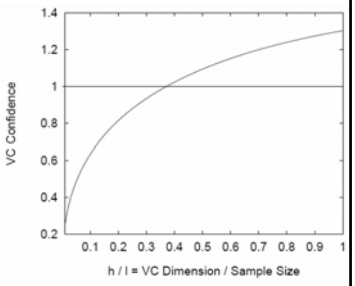
$$R(\alpha) \le R_{emp}(\alpha) + \underbrace{\sqrt{\left(\frac{h(\log(2l/h)+1)-\log(\eta/4)}{l}\right)}}_{\textbf{VC confidence}}$$

h: VC dimension, capacity

l: size of the training set

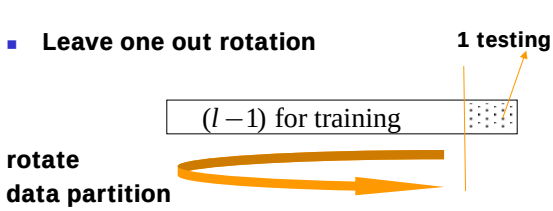
- **VC dim of hyperplane in R^n is (n+1)**

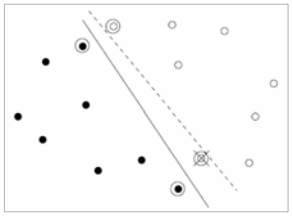




A tighter error bound of SVM

- **Leave one out rotation**





- First, train a SVM over the *l* samples
- In each rotation, re-train the SVM over the l-1 samples
- if the test sample is not SV, then SVM does not change and there is no error.

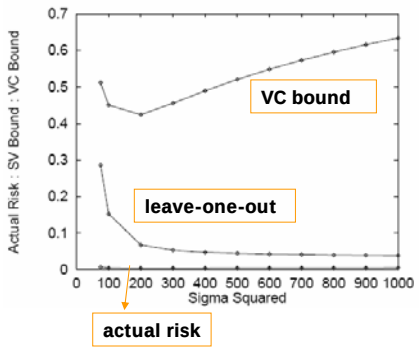
$$E[P(error)] \leq \frac{E[\#\text{ of support vectors}]}{\text{number of training samples}}$$

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

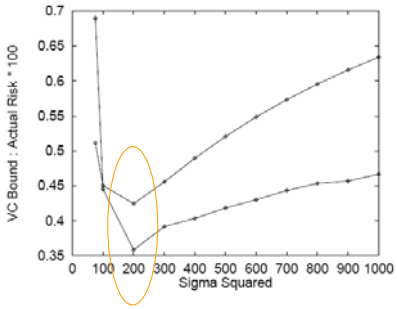
$E[P(error)]$: expected risk over all choices of training set of l-1 samples

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



Scale the actual risk by 100

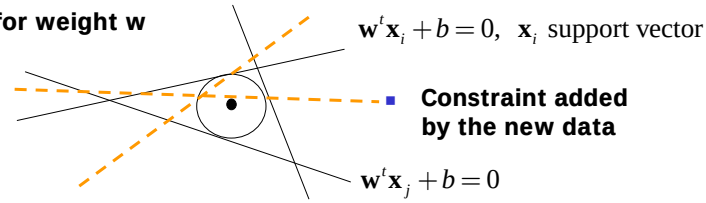


VC bound has good predictive power

- **Leave-one-out bound is tighter than VC bound in this experiment (NIST digit classification)**
- **But VC bound ahs better 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**
  - **which sample to choose?**

- **Choose the un-labeled sample that is closest to the current separation plane**