Today’s lecture
- SVM with kernel, error bounds

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
  
  \[ H_0: \ w^\top x + b = 0 \]

- Linearly separable
  
  \[ w^\top x_i + b \geq 1 \quad \forall x_i \text{ in class } \omega_1 \quad \text{i.e.} \quad y_i = +1 \]
  
  \[ w^\top x_i + b \leq -1 \quad \forall x_i \text{ in class } \omega_2 \quad \text{i.e.} \quad y_i = -1 \]

  Inequality constraints: \[ y_i(w^\top x_i + b) - 1 \geq 0 \quad \forall i \]

- Two parallel hyperplanes defining the margin
  
  hyperplane \( H_i(H_+): w^\top x_i + b = +1 \)
  
  hyperplane \( H_i(H_-): w^\top x_i + b = -1 \)

- Margin: sum of distances of the closest points to the separation plane
  
  \[ \text{margin} = \frac{2}{\|w\|} \]

  - Best plane defined by \( w \) and \( b \)

KKT conditions (iff) for separable case

\[
\frac{\partial}{\partial w} L_P = w - \sum_{i} \alpha_i y_i x_i = 0 \\
\frac{\partial}{\partial b} L_P = -\sum_{i} \alpha_i y_i = 0 \\
y_i(w^\top x_i + b) - 1 \geq 0 \quad i = 1, \ldots, l \]

\[ \alpha_i \geq 0 \quad \forall i \]

\[ \alpha_i (y_i(w^\top x_i + b) - 1) = 0 \quad \forall i \]

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

  if \( \alpha_i > 0 \), \( x_i \) 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 $\xi_i$

  \[ x_i \cdot w + b \geq +1 - \xi_i \text{ for } y_i = +1 \]
  \[ x_i \cdot w + b \leq -1 + \xi_i \text{ for } y_i = -1 \]
  \[ \xi_i \geq 0 \text{ for all } i. \]

  if $\xi_i > 1$, then $x_i$ is misclassified (i.e. training error)

Lagrange multiplier: minimize

\[ L_P = \frac{1}{2} \|w\|^2 + C \sum_i \xi_i - \sum_i \alpha_i \{y_i(x_i \cdot w + b) - 1 + \xi_i\} - \sum \mu_i \xi_i \]

New objective function

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 $\xi$ both decrease

When $C$ increases, samples with errors get more weights

- better training accuracy, but smaller margin
- less generalization performance

EE6887-Chang
Generalized Linear Discriminant Functions

- Include more than just the linear terms
  \[ g(x) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d} \sum_{j=1}^{d} w_{ij} x_i x_j = w_0 + w^T x + x^T W x \]

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

- In general \( g(x) = \sum_{i=1}^{d} a_i y_i (x) = a^T y \)

Example

\[
g(x) = a_1 + a_2 x + a_3 x^2 \quad g(x) = a_1 x_1 + a_2 x_2 + a_3 x_3 \\
= [a_1 \quad a_2 \quad a_3] [1 \quad x \quad x^2]^T = [a_1 \quad a_2 \quad a_3] [x_1 \quad x_2 \quad x_3]^T
\]

- Data become separable in higher-dimensional space
  - learning parameters in high dimension is hard (curse of dim.)
  - instead, try to maximize margins \( \rightarrow \text{SVM} \)

Non-Linear Space

\( \Phi : \mathbb{R}^d \mapsto \mathcal{H} \). Map to a high dimensional space, \( \Phi(x) = \left( \begin{array}{c} x_1^2 \\ \sqrt{2} x_1 x_2 \\ x_2^2 \end{array} \right) \)

- Find the SVM in the high-dim space (embedding space)
  \[ g(x) = \sum_{i=1}^{N_t} \alpha_i y_i \Phi(s_i) \cdot \Phi(x) + b \]

- Luckily, we don’t have to find \( \Phi(s_i) \) nor \( \sum_{i=1}^{N_t} \alpha_i y_i \Phi(s_i) \)

- Instead, we define kernel \( K(s_i, x) = \Phi(s_i) \cdot \Phi(x) \)

  \[ g(x) = \sum_{i=1}^{N_t} \alpha_i y_i K(s_i, x) + b \]

- We can use the same method to maximize \( L_D \) to find \( \alpha_i \)

\[
L_D = \sum_{i=1}^{N_t} \alpha_i - \frac{1}{2} \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} \alpha_i \alpha_j y_i y_j \Phi(x_i) \cdot \Phi(x_j) \\
= \sum_{i=1}^{N_t} \alpha_i - \frac{1}{2} \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} \alpha_i \alpha_j y_i y_j K(x_i, x_j)
\]
Some popular kernels

- $K(x, y) = (x \cdot y + 1)^p$  \textbf{polynomial}
- $K(x, y) = e^{-||x-y||^2/2\sigma^2}$ \textbf{Gaussian Radial Basis Function (RBF)}
- $K(x, y) = \tanh(\kappa x \cdot y - \delta)$ \textbf{sigmoidal neural network}

Separable Cubic polynomial Non-separable

Error bound based on VC dimension (Vapnik ’95)

- \textbf{Risk: expectation of loss}
  \[ R(\alpha) = E\left[\frac{1}{2}\left|y_i - f(x_i, \alpha)\right|\right], \text{ where } \alpha \text{ is the classifier model parameter} \]
- $R_{\text{emp}}(\alpha)$: empirical risk for a specific classifier over a training/test set
  \[ R(\alpha) \leq R_{\text{emp}}(\alpha) + \sqrt{\frac{h(\log(2l/h) + 1) - \log(\eta/4)}{l}} \quad \text{with probability } (1-\eta) \]
  \[ h: \text{ VC dimension, capacity} \]
  \[ l: \text{ size of the training set} \]

- \textbf{VC dim of hyperplane in } \mathbb{R}^n \text{ is } (n+1)
A tighter error bound of SVM

- **Leave one out rotation**
  - 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.

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

\(P(error)\): 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
\(E[\# \text{ of s.v.}]\): expected \# of s.v. over all choices of training set of size \(l\)

Potential use and issue of the error bound

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

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

- Space for weight $w$
  - $w^T x_j + b = 0$, $x_j$ support vector
  - Constraint added by the new data
  - $w^T x_j + b = 0$

- 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?