Kernel Machines

CptS 570 – Machine Learning
School of EECS
Washington State University
Kernel Machines

- Or, support vector machine (SVM)
- Discriminant-based method
  - Learn class boundaries
- Support vector consists of examples closest to boundary
- Kernel computes similarity between examples
  - Maps instance space to a higher-dimensional space where (hopefully) linear models suffice
- Choosing the right kernel is crucial
- Kernel machines among best-performing learners
Kernel Machines

- Likely to underfit using only hyperplanes
- But we can map the data to a nonlinear space and use hyperplanes there
  - $\Phi: \mathbb{R}^d \rightarrow \mathcal{F}$
  - $\mathbf{x} \rightarrow \Phi(\mathbf{x})$
Optimal Separating Hyperplane

\[ \mathcal{X} = \{x^t, r^t\} \text{ where } r^t = \begin{cases} +1 & \text{if } x^t \in C_1 \\ -1 & \text{if } x^t \in C_2 \end{cases} \]

find \( w \) and \( w_0 \) such that
\[ w^T x^t + w_0 \geq +1 \text{ for } r^t = +1 \]
\[ w^T x^t + w_0 \leq -1 \text{ for } r^t = -1 \]
which can be rewritten as
\[ r^t(w^T x^t + w_0) \geq +1 \]

- Note we want \( \geq +1 \), not \( \geq 0 \)
- Want instances some distance from hyperplane
Margin

- Distance from instance $x^t$ to hyperplane $w^T x^t + w_0$
  \[
  \frac{|w^T x^t + w_0|}{\|w\|} \quad \text{or} \quad \frac{r^t(w^T x^t + w_0)}{\|w\|}
  \]

- Distance from hyperplane to closest instances is the margin
Optimal separating hyperplane is the one maximizing the margin

We want to choose $w$ maximizing $\rho$ such that

$$\frac{r^t(w^T x^t + w_0)}{\|w\|} \geq \rho, \forall t$$

Infinite number of solutions by scaling $w$

Thus, we choose solution minimizing $\|w\|

$$\min \frac{1}{2} \|w\|^2 \text{ subject to } r^t(w^T x^t + w_0) \geq +1, \forall t$$
Optimal Separating Hyperplane

\[
\min \frac{1}{2} \|w\|^2 \text{ subject to } r^t \left( w^T x^t + w_0 \right) \geq +1, \forall t
\]

- Quadratic optimization problem with complexity polynomial in \(d\) (#features)
- Kernel will eventually map \(d\)-dimensional space to higher-dimensional space
- Prefer complexity not based on \#dimensions
Optimal Separating Hyperplane

- Convert optimization problem to depend on number of training examples \( N \) (not \( d \))
  - Still polynomial in \( N \)
- But optimization will depend only on closest examples (support vector)
  - Typically \( \ll N \)
Lagrange Multipliers

- Rewrite quadratic optimization problem using Lagrange multipliers $\alpha^t$, $1 \leq t \leq N$

$$\min \frac{1}{2} \|w\|^2 \text{ subject to } r^t (w^T x^t + w_0) \geq +1, \forall t$$

$$L_p = \frac{1}{2} \|w\|^2 - \sum_{t=1}^{N} \alpha^t \left[r^t (w^T x^t + w_0) - 1\right]$$

$$= \frac{1}{2} \|w\|^2 - \sum_{t=1}^{N} \alpha^t r^t (w^T x^t + w_0) + \sum_{t=1}^{N} \alpha^t$$

- Minimize $L_p$
Equivalelently, we can maximize $L_p$ subject to the constraints:

$$\frac{\partial L_p}{\partial w} = 0 \Rightarrow w = \sum_{t=1}^{N} \alpha_t r^t x^t$$

$$\frac{\partial L_p}{\partial w_0} = 0 \Rightarrow \sum_{t=1}^{N} \alpha_t r^t = 0$$

Plugging these into $L_p$ ...
Lagrange Multipliers

\[ L_d = \frac{1}{2} (w^T w) - w^T \sum_t \alpha^t r^t x^t - w_0 \sum_t \alpha^t r^t + \sum_t \alpha^t \]

\[ = -\frac{1}{2} (w^T w) + \sum_t \alpha^t \]

\[ = -\frac{1}{2} \sum_t \sum_s \alpha^t \alpha^s r^t r^s (x^t)^T x^s + \sum_t \alpha^t \]

subject to \( \sum_t \alpha^t r^t = 0 \) and \( \alpha^t \geq 0, \forall t \)

- Maximize \( L_d \) with respect to \( \alpha^t \) only
- Complexity \( O(N^3) \)
Support Vector Machine

- Most $\alpha^t = 0$
  - I.e., $r^t(w^T x^t + w_0) > 1$ ($x^t$ lie outside margin)

- Support vectors: $x^t$ such that $\alpha^t > 0$
  - I.e., $r^t(w^T x^t + w_0) = 1$ ($x^t$ lie on margin)

- $w = \sum_t \alpha^t r^t x^t$

- $w_0 = r^t - w^T x^t$ for any support vector $x^t$
  - Typically average over all support vectors

- Resulting discriminant is called the support vector machine (SVM)
Support Vector Machine

O = support vectors

margin
Soft Margin Hyperplane

- Data not linearly separable
- Find hyperplane with least error
- Define slack variables $\xi^t \geq 0$ storing deviation from the margin

\[ r^t (w^T x^t + w_0) \geq 1 - \xi^t \]
(a) Correctly classified example far from margin ($\xi^t = 0$)
(b) Correctly classified example on the margin ($\xi^t = 0$)
(c) Correctly classified example, but inside the margin ($0 < \xi^t < 1$)
(d) Incorrectly classified example ($\xi^t \geq 1$)

Soft error = $\sum_{t}^{\xi^t}$
Soft Error

O = support vectors

margin
Soft Margin Hyperplane

- Lagrangian equation with slack variables

\[ L_p = \frac{1}{2} \|w\|^2 + C \sum_t \xi^t - \sum_t \alpha^t \left[ r^t (w^T x^t + w_0) - 1 + \xi^t \right] - \sum_t \mu^t \xi^t \]

- C is penalty factor
- \( \mu^t \geq 0 \), new set of Lagrange multipliers
- Want to minimize \( L_p \)
Minimize $L_p$ by setting derivatives to zero

$$\frac{\partial L_p}{\partial w} = 0 \Rightarrow w = \sum_{t=1}^{N} \alpha^t r^t x^t$$

$$\frac{\partial L_p}{\partial w_0} = 0 \Rightarrow \sum_{t=1}^{N} \alpha^t r^t = 0$$

$$\frac{\partial L_p}{\partial \xi^t} = 0 \Rightarrow C - \alpha^t - \mu^t = 0$$

Plugging these into $L_p$ yields dual $L_d$

Maximize $L_d$ with respect to $\alpha^t$
Soft Margin Hyperplane

\[ L_d = -\frac{1}{2} \sum_t \sum_s \alpha^t \alpha^s r^t r^s (x^t)^T x^s + \sum_t \alpha^t \]

subject to \( \sum_t \alpha^t r^t = 0 \) and \( 0 \leq \alpha^t \leq C, \forall t \)

- Quadratic optimization problem
- Support vectors have \( \alpha^t > 0 \)
  - Examples on margin: \( \alpha^t < C \)
  - Examples inside margin or misclassified: \( \alpha^t = C \)
C is a regularization parameter

- High C → high penalty for non-separable examples (overfit)
- Low C → less penalty (underfit)
- Determine using validation set (C=1 typical)

\[ L_d = -\frac{1}{2} \sum_t \sum_s \alpha_t \alpha_s r_t r_s (x^t)^T x^s + \sum_t \alpha_t \]

subject to \( \sum_t \alpha_t r_t = 0 \) and \( 0 \leq \alpha_t \leq C, \forall t \)
Kernels

- To use previous approaches, data must be near linearly separable
- If not, perhaps a transformation $\phi(x)$ will help
- $\phi(x)$ are basis functions
Kernels

- Transform $d$-dimensional $\mathbf{x}$ space to $k$-dimensional $\mathbf{z}$ space using basis functions $\phi(\mathbf{x})$

\[ \mathbf{z} = \phi(\mathbf{x}) \quad \text{where} \quad z_j = \phi_j(\mathbf{x}) \quad j=1,\ldots,k \]

\[ g(\mathbf{z}) = \mathbf{w}^T \mathbf{z} \]

\[ g(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) = \sum_{j=1}^{k} w_j \phi_j(\mathbf{x}) \]

- Instead of $w_0$, assume $z_1 = \phi_1(\mathbf{x}) \equiv 1$
Soft Margin with Kernels

\[ L_p = \frac{1}{2} \|w\|^2 + C \sum_t \xi_t - \sum_t \alpha_t [r^T w^T \phi(x^t) - 1 + \xi_t] - \sum_t \mu_t \xi_t \]

\[ L_d = -\frac{1}{2} \sum_t \sum_s \alpha_t \alpha_s r^t r^s \phi(x^t)^T \phi(x^s) + \sum_t \alpha_t \]

subject to \( \sum_t \alpha_t r^t = 0 \) and \( 0 \leq \alpha_t \leq C, \forall t \)

- Replace inner product of basis functions \( \phi(x^t)^T \phi(x^s) \) with kernel function \( K(x^t, x^s) \)

\[ L_d = -\frac{1}{2} \sum_t \sum_s \alpha_t \alpha_s r^t r^s K(x^t, x^s) + \sum_t \alpha_t \]
Kernel Functions

- Kernel $K(x^t, x^s)$ computes $z$-space product $\phi(x^t)^T \phi(x^s)$ in $x$-space

\[ w = \sum_t \alpha^t r^t z^t = \sum_t \alpha^t r^t \phi(x^t) \]

\[ g(x) = w^T \phi(x) = \sum_t \alpha^t r^t \phi(x^t)^T \phi(x) \]

\[ g(x) = \sum_t \alpha^t r^t K(x^t, x) \]

- Matrix of kernel values $K$, where $K_{ts} = K(x^t, x^s)$, called the Gram matrix
- $K$ should be symmetric and positive semidefinite
Kernel Functions

- Polynomial kernel of degree \( q \)
  \[
  K(x^t, x) = (x^T x^t + 1)^q
  \]
- If \( q = 1 \), then use original features
- For example, when \( q = 2 \) and \( d = 2 \)
  \[
  K(x, y) = (x^T y + 1)^2
  = (x_1 y_1 + x_2 y_2 + 1)^2
  = 1 + 2x_1 y_1 + 2x_2 y_2 + 2x_1 x_2 y_1 y_2 + x_1^2 y_1^2 + x_2^2 y_2^2
  \]
  \[
  \phi(x) = [1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2, x_1^2, x_2^2]^T
  \]
Kernel Functions

- Polynomial kernel of degree 2

O = support vectors

margin
Kernel Functions

- Radial basis functions (Gaussian kernel)

\[ K(x^t, x) = \exp \left[ -\frac{||x^t - x||^2}{2s^2} \right] \]

- $x^t$ is the center
- $s$ is the radius
- Larger $s$ implies smoother boundaries
Kernel Functions: Radial Basis Functions

(a) $s^2 = 2$

(b) $s^2 = 0.5$

(c) $s^2 = 0.25$

(d) $s^2 = 0.1$
Kernel Functions

- Sigmoidal functions

\[ K(x^t, x) = \tanh(2x^T x^t + 1) \]
Defining Kernels

- Kernel $K(x,y)$ increases with similarity between $x$ and $y$
- Prior knowledge can be included in the kernel function
  - E.g., training examples are documents
    - $K(D_1,D_2) = \# \text{ shared words}$
  - E.g., training examples are strings (e.g., DNA)
    - $K(S_1,S_2) = \frac{1}{\text{edit distance between } S_1 \text{ and } S_2}$
    - Edit distance is the number of insertions, deletions and/or substitutions to transform $S_1$ into $S_2$
Defining Kernels

- E.g., training examples are nodes in a graph (e.g., social network)
- \( K(N_1, N_2) = \frac{1}{\text{length of shortest path connecting nodes}} \)
- \( K(N_1, N_2) = \# \text{paths connecting nodes} \)
- Diffusion kernel
Defining Kernels

- Training examples are graphs, not feature vectors
  - E.g., carcinogenic vs. non-carcinogenic chemical structures
- Compare substructures of graphs
  - E.g., walks, paths, cycles, trees, subgraphs
- \( K(G_1, G_2) = \) number of identical random walks in both graphs
- \( K(G_1, G_2) = \) number of subgraphs shared by both graphs
Combining Kernels

- Training data from multiple modalities (e.g., biometrics, social network, audio/visual)
- Construct new kernels by combining simpler kernels
- If \( K_1(x,y) \) and \( K_2(x,y) \) are valid kernels, and \( c \) is a constant, then

\[
K(x,y) = \begin{cases} 
  cK(x,y) \\
  K_1(x,y) + K_2(x,y) \\
  K_1(x,y)K_2(x,y)
\end{cases}
\]

- are valid kernels
Combining Kernels

- Adaptive kernel combination

\[ K(x, y) = \sum_{i=1}^{m} \eta_i K_i(x, y) \]

\[ L_d = \sum_{t} \alpha^t - \frac{1}{2} \sum_{t} \sum_{s} \alpha^t \alpha^s r^t r^s \sum_{i} \eta_i K_i(x^t, x^s) \]

\[ g(x) = \sum_{t} \alpha^t r^t \sum_{i} \eta_i K_i(x^t, x) \]

- Learn both \( \alpha^t \)s and \( \eta_i \)s
K > 2 Classes

- Learn K different kernel machines $g_i(x)$
  - Each uses one class as positive, remaining classes as negative
  - Choose class $i$ such that $i = \text{argmax}_j g_j(x)$
  - Best approach in practice
K>2 Classes

- Learn $K(K-1)/2$ kernel machines
  - Each uses one class as positive and another class as negative
  - Easier (faster) learning per kernel machine
$K>2$ Classes

- Learn all margins at once

$$\min \frac{1}{2} \sum_{i=1}^{K} \|w_i\|^2 + C \sum_i \sum_t \xi_i^t$$

subject to

$$w_{z_t^t}^T x_t + w_{z_t^0} \geq w_i^T x_t + w_{i0} + 2 - \xi_i^t, \forall i \neq z_t, \xi_i^t \geq 0$$

- $z_t^t$ is the class index of $x_t$

- $K^*N$ variables to optimize (expensive)
SVM Regression

- Normally, we would use squared error
  \[ e(r^t, f(x^t)) = [r^t - f(x^t)]^2 \]
  \[ f(x) = w^T x + w_0 \]

- For SVM, we use \( \varepsilon \)-sensitive loss
  \[ e_\varepsilon(r^t, f(x^t)) = \begin{cases} 
  0 & \text{if } |r^t - f(x^t)| < \varepsilon \\
  |r^t - f(x^t)| - \varepsilon & \text{otherwise}
\end{cases} \]
  - Tolerate errors up to \( \varepsilon \)
  - Errors beyond \( \varepsilon \) have only linear effect
SVM Regression

- Use slack variables to account for deviations beyond $\varepsilon$
  - $\xi^+_t$ for positive deviations
  - $\xi^-_t$ for negative deviations

$$\min \frac{1}{2} \|w\|^2 + C \sum_t (\xi^+_t + \xi^-_t)$$

Subject to

$$r^t - \left( w^T x + w_0 \right) \leq \varepsilon + \xi^+_t$$

$$\left( w^T x + w_0 \right) - r^t \leq \varepsilon + \xi^-_t$$

$$\xi^+_t, \xi^-_t \geq 0$$
SVM Regression

\[
L_d = -\frac{1}{2} \sum_t \sum_s \left( \alpha_+^t - \alpha_-^t \right) \left( \alpha_+^s - \alpha_-^s \right) (x^t)^T x^s
\]

\[-\varepsilon \sum_t (\alpha_+^t + \alpha_-^t) - \sum_t r^t (\alpha_+^t - \alpha_-^t)\]

subject to

\[0 \leq \alpha_+^t \leq C, \quad 0 \leq \alpha_-^t \leq C, \quad \sum_t (\alpha_+^t - \alpha_-^t) = 0\]

- Non-support vectors (inside margin): \( \alpha_+^t = \alpha_-^t = 0 \)
- Support vectors
  - \( \otimes \) on the margin: \( 0 < \alpha_+^t < C \) or \( 0 < \alpha_-^t < C \)
  - \( \blacklozenge \) outside margin (outlier): \( \alpha_+^t = C \) or \( \alpha_-^t = C \)
SVM Regression
SVM Regression

- Fitted line
- \( f(x) \) is weighted sum of support vectors

\[
 f(x) = w^T x + w_0 = \sum_t (\alpha_+^t - \alpha_-^t)(x^t)^T x + w_0
\]

\[
 w = \sum_t (\alpha_+^t - \alpha_-^t)x^t
\]

- Average \( w_0 \) over:

\[
 r^t = w^T x^t + w_0 + \varepsilon, \quad \text{if} \ 0 < \alpha_+^t < C
\]

\[
 r^t = w^T x^t + w_0 - \varepsilon, \quad \text{if} \ 0 < \alpha_-^t < C
\]
Kernel Regression

\[ L_d = -\frac{1}{2} \sum_t \sum_s (\alpha_+^t - \alpha_-^t)(\alpha_+^s - \alpha_-^s)K(x^t, x^s) \]

\[ -\varepsilon \sum_t (\alpha_+^t + \alpha_-^t) - \sum_t r^t(\alpha_+^t - \alpha_-^t) \]

subject to

\[ 0 \leq \alpha_+^t \leq C, \quad 0 \leq \alpha_-^t \leq C, \quad \sum_t (\alpha_+^t - \alpha_-^t) = 0 \]

\[ f(x) = w^T x + w_0 = \sum_t (\alpha_+^t - \alpha_-^t)K(x^t, x) + w_0 \]
Kernel Regression

- Polynomial (quadratic) kernel
- Gaussian kernel
WEKA Kernel Machines

- Classification: SMO
- Regression: SMOreg
- Sequential Minimal Optimization (SMO)
- Kernels
  - Polynomial
  - RBF
  - String
Summary: Kernel Machines

- Seek optimal separating hyperplane
- Support vector machine (SVM) finds hyperplane using only closest examples
- Kernel function allows SVM to operate in higher dimensions
- Kernel regression
- Choosing correct kernel is crucial
- Kernel machines among best–performing learners