Support Vector Machines

Chapter 9, Learning from Data

Introduction (1)

- Universal constructive learning procedure
  - Based on statistical learning theory (Vapnik, 1995)
  - Used to learn a variety of representations
    - neural nets, radial basis functions, splines, polynomial estimators
  - Provides a new form of parameterization of functions.
  - Provides a meaningful characterization of the function's complexity that is independent of the problem's dimensionality.

Introduction (2)

- Motivation
  - For nonlinear models
    - VC-dimension cannot be accurately estimated.
    - Implementation of structural risk minimization leads to nonlinear optimization.
  - For linear models of large multivariate problems
    - The curse of dimensionality
Introduction (3)

- SVM overcomes two problems
  1. Conceptual problem
     - How to control the complexity of the set of approximating functions in a high-dimensional space in order to provide good generalization ability.
     - Using penalized linear estimators with a large number of basis functions.
  2. Computational problem
     - How to perform numerical optimization in a high-dimensional space.
     - Using the dual kernel representation of linear functions.

Introduction (4)

- SVM combines four distinct concepts
  1. New implementation of the SRM inductive principle:
     - SVM can analytically estimate the VC-dim:
       - Minimize the VC-dim, keeping the value of the empirical risk nearly zero.
     - Ordinary SRM implementation.
       - About each $V_{C_1} < V_{C_2} < ... < V_{C_n}$ models,
       - Minimize each empirical risk.
       - Choose the best model of which guaranteed risk is small.

Introduction (5)

2. Input samples mapped onto a very high-dimensional space using a set of nonlinear basis functions defined a priori:
   - In ordinary learning problem, feature space is usually made for the purpose of reduction of complexity.

Introduction (6)

3. Linear functions with constraints on complexity used to approximate or discriminate the input samples in the high-dimensional space:
   - Accurate estimates for model complexity can be obtained for linear estimators.
   - The drawbacks of nonlinear estimators
     - Lack of complexity measures
     - Lack of optimization approaches
4. *Duality theory of optimization used to make estimation of model parameters in a high-dimensional feature space computationally tractable.*

- In SVM, a quadratic programming is used for optimization.
- In original problem, large number of parameter must be estimated, which makes the problem intractable.
- The size of dual problem scales in size with the number of training samples.
- The solution of dual problem becomes the support vectors’ weights

### 9.1. Optimal Separating Hyperplane

- **Separating hyperplane**
  - A linear function that is capable of separating the training data
    
    \[
    D(x) = (w \cdot x) + w_0
    \]
    
    \[
    y_i[(w \cdot x_i) + w_0] \geq 1, \quad i = 1, ..., n \quad (9.3)
    \]
  - Note that when linearly separable case, \( w, w_0 \) can be scaled so that next condition holds.
    
    \[
    (w \cdot x) + w_0 \geq 1 \quad \text{if} \quad y_i = +1
    \]
    
    \[
    (w \cdot x) + w_0 \leq -1 \quad \text{if} \quad y_i = -1, \quad i = 1, ..., n
    \]

- **Margin:** \( \tau \)
  - Minimal distance from the separating hyperplane to the closest data

- **Optimal separating hyperplane (s.h.)**
  - When the margin is the maximum size.

- **Distance between s.h. and a sample \( x' \)**
  
  \[
  |D(x')|/||w||
  \]

- **All patterns obey the inequality**
  
  \[
  \frac{y_k D(x_k)}{||w||} \geq \tau, \quad k = 1, ..., n
  \]
9.1. Optimal Separating Hyperplane

- Maximizing the margin = Minimizing $||w||$
  $$\tau = \frac{1}{||w||}$$

- Support Vector
  - The data that exist at the margin (when the equality condition of (9.3) is satisfied).
  - Dimensionality independent generalization error bound
    $$E_n[\text{Error rate}] \leq E_n[\text{Number of support vectors}] \frac{n}{n}$$
  - Number of SVs is much smaller than number of patterns in most cases.

- The VC-dim of hyperplane of (9.3) satisfying $c \geq ||w||^2$
  $$h \leq \min(r^2c,d) + 1$$

- SRM implementation
  $$R(w) \leq R_{\text{emp}}(w) + \Phi$$
  - S. t. always has zero empirical risk
  - $\Phi$ is minimized by minimizing the VC-dim $h$, which corresponds to minimizing $||w||^2$

- Quadratic optimization problem
  $$\min_w \eta(w) = \frac{1}{2} ||w||^2$$
  subject to
  $$y_i[(w \cdot x_i) + w_0] \geq 1, \quad i = 1, \ldots, n$$
  - Minimizing quadratic function with linear constraints.
  - The solution consists of $d+1$ parameters.
### 9.1. Optimal Separating Hyperplane

#### (8)

- **Dual problem**
  - The solution consists of $n$ parameters.
  - Convertible if cost and constraint are convex.
- **Step 1 of conversion**
  - Construct Lagrangian function
    \[
    Q(w, w_0, \alpha) = \frac{1}{2} (w \cdot w) - \sum_{i=1}^{n} \alpha_i \{ y_i (w \cdot x_i) + w_0 \} - 1
    \]
- **Step 2 of conversion**
  - Using the optimal condition

#### (9)

- Dual problem
  - The solution consists of $n$ parameters.
  - Convertible if cost and constraint are convex.
- **Step 1 of conversion**
  - Construct Lagrangian function
    \[
    Q(w, w_0, \alpha) = \frac{1}{2} (w \cdot w) - \sum_{i=1}^{n} \alpha_i \{ y_i (w \cdot x_i) + w_0 \} - 1
    \]
- **Step 2 of conversion**
  - Using the optimal condition

#### (10)

- Dual problem
  - The solution consists of $n$ parameters.
  - Convertible if cost and constraint are convex.
- **Step 1 of conversion**
  - Construct Lagrangian function
    \[
    Q(\alpha) = -\frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^{n} \alpha_i
    \]
- **Step 2 of conversion**
  - Using the optimal condition

#### (11)

- The resulting equation s.h.
  - The data corresponding nonzero $\alpha_i$ are support vectors.
  - Using the optimal condition

\[
\alpha_i^* \{ y_i (w^* \cdot x_i) + w_0^* \} - 1 = 0, \quad i = 1, ..., n
\]
§1. Optimal Separating Hyperplane

(12)

- Nonseparable problem
  - Certain data point where doesn’t satisfy (9.3) exists.
- Introducing positive slack variables $\xi_i$
  $$y_i[(w \cdot x_i) + w_0] \geq 1 - \xi_i$$  \hspace{2cm} (9.25)
- Optimization problem
  $$Q(w) = \sum_{i=1}^{n} I(\xi_i > 0)$$  \hspace{2cm} (9.26)
  - (9.26) is combinatorial optimization and very difficult because of the nonlinearity.

(13)

- Dual Problem
  $$\text{maximize} \ Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$
  subject to
  $$\sum_{i=1}^{n} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq \frac{C}{n}, \quad i = 1, \ldots, n$$
- Resulting equation of s.h.
  $$D(x) = \sum_{i=1}^{n} \alpha_i^* y_i (x \cdot x_i) + w_0^*$$

(14)

- Approximation of (9.26) is used
  $$Q(\xi) = \sum_{i=1}^{n} \xi_i^p$$  \hspace{2cm} (9.27)
- QP (when $p=1$)
  $$\text{minimize} \quad \frac{C}{n} \sum_{i=1}^{n} \xi_i + \frac{1}{2} \|w\|^2$$
  subject to
  $$y_i[(w \cdot x_i) + w_0] \geq 1 - \xi_i$$

(15)
§ 2. High-Dimensional Mapping and Inner-Product Kernels (1)

- Complexity of optimal hyperplanes are dimensionality independent.
- Dual problem only needs the inner product between vectors in feature space.
- Nonlinear transformation function \( g(\mathbf{x}) = [g_1(\mathbf{x}), \ldots, g_m(\mathbf{x})] \).
  - Even for a small problem the feature space can be very large.

§ 2. High-Dimensional Mapping and Inner-Product Kernels (2)

- Example
  - \( g_j(x), j=1, \ldots, m \) are polynomial terms of \( \mathbf{x} \) up to 3rd-order
  - Feature space has 16 dimension.

\[
\begin{align*}
g_1(x_1, x_2) = 1 & \quad g_2(x_1, x_2) = x_1 \\
g_3(x_1, x_2) = x_1^2 & \quad g_4(x_1, x_2) = x_1^2 x_2 \\
g_5(x_1, x_2) = x_2 & \quad g_6(x_1, x_2) = x_1 x_2 \\
g_7(x_1, x_2) = x_1 x_2^2 & \quad g_8(x_1, x_2) = x_1^3 \\
g_9(x_1, x_2) = x_2^2 & \quad g_{10}(x_1, x_2) = x_1^2 x_2^2 \\
g_{11}(x_1, x_2) = x_1^3 & \quad g_{12}(x_1, x_2) = x_1^3 x_2 \end{align*}
\]

§ 2. High-Dimensional Mapping and Inner-Product Kernels (3)

- Decision function
  \[
  D(\mathbf{x}) = \sum_{j=0}^{\infty} w_j g_j(\mathbf{x})
  \]
- Dual form of decision function
  \[
  D(\mathbf{x}) = \sum_{i=1}^{N} y_i H(\mathbf{x}, \mathbf{x}_i)
  \]
  - where
    \[
    H(\mathbf{x}, \mathbf{x}') = g(\mathbf{x})^T g(\mathbf{x}') = \sum_{j=1}^{m} g_j(\mathbf{x}) g_j(\mathbf{x}')
    \]

§ 2. High-Dimensional Mapping and Inner-Product Kernels (4)

- Any symmetric function \( H(\mathbf{x}, \mathbf{x}') \) satisfying the Mercer's condition can be used as a inner product.
  \[
  \int \int H(\mathbf{x}, \mathbf{x}') \varphi(\mathbf{x}) \varphi(\mathbf{x}') d\mathbf{x} d\mathbf{x}' > 0 \quad \text{for all } \varphi \neq 0, \int \varphi^2(\mathbf{x}) d\mathbf{x} < \infty
  \]
  - Polynomials of degree \( q \):
    \[
    H(\mathbf{x}, \mathbf{x}') = [(\mathbf{x} \cdot \mathbf{x}') + 1]^q
    \]
  - RBF with width \( \sigma \):
    \[
    H(\mathbf{x}, \mathbf{x}') = \exp \left\{ - \frac{||\mathbf{x} - \mathbf{x}'||^2}{\sigma^2} \right\}
    \]
9.2. High-Dimensional Mapping and Inner Product Kernels (4)

- Neural network with parameters \( v, a \) satisfying the Mercer’s theorem:
  \[
  H(x, x') = \tanh(v(x \cdot x') + a)
  \]

Example 9.1

- The exclusive–or (XOR) problem
- The inner product kernel for polynomial
  \[
  H(x, x') = [(x \cdot x')]^2
  \]
- The set of basis function
  \[
  \varphi(x) = [1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2]^T
  \]
- Solve the dual problem when \( C = \infty \)

9.3. Support Vector Machine for Classification (1)

- Decision function for nonseparable data
  \[
  D(x) = \sum_{i=1}^{n} \alpha_i y_i H(x_i, x)
  \]
- Dual problem
  \[
  \max_{\alpha} Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j H(x_i \cdot x_j)
  \]
  subject to
  \[
  \sum_{i=1}^{n} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq \frac{C}{n}, \quad i = 1, \ldots, n
  \]

Example 9.1

<table>
<thead>
<tr>
<th>Index</th>
<th>( x )</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1, 1)</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>(0, -1)</td>
<td>-1</td>
</tr>
<tr>
<td>3</td>
<td>(-1, -1)</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>(-1, 1)</td>
<td>-1</td>
</tr>
</tbody>
</table>

![Figure 9.1](image)
Example 9.1

maximize \( Q(\alpha) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2} \sum_{i,j=1}^{4} \alpha_i \alpha_j y_i y_j h_{ij} \)

subject to
\[
\sum_{i=1}^{4} y_i \alpha_i = \alpha_1 - \alpha_2 + \alpha_3 - \alpha_4 = 0
\]
\[
0 \leq \alpha_1
\]
\[
0 \leq \alpha_2
\]
\[
0 \leq \alpha_3
\]
\[
0 \leq \alpha_4
\]

Example 9.1

- Inner product model
\[
H = \begin{bmatrix}
9 & 1 & 1 & 1 \\
1 & 9 & 1 & 1 \\
1 & 1 & 9 & 1 \\
1 & 1 & 1 & 9
\end{bmatrix}
\]
- The solution to this optimization problem
\[
\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 0.125
\]
- The decision function in the inner product representation
\[
D(x) = \sum_{i=1}^{4} \alpha_i y_i H(x, x) = (0.125) \sum_{i=1}^{4} y_i [(x_i \cdot x) + 1]^2
\]

9.4. Support Vector Machine for Regression (1)

- A function linear in parameters is used to approximate the regression in the feature space.
\[
f(x, w) = \sum_{j=1}^{m} w_j g_f(x)
\]
- A special loss function (Vapnik’s loss function)
\[
L_\epsilon(y, f(x, w)) = \begin{cases} 
|y - f(x, w)| & \text{if } |y - f(x, w)| \leq \epsilon \\
\epsilon & \text{otherwise}
\end{cases}
\]
- More relaxed assumption about noise than \( L_2 \) loss function.
- \( \epsilon \) controls the width of the insensitive zone.
§.4. Support Vector Machine for Regression (2)

- **Quadratic Problem**
  
  \[
  \text{minimize} \quad \frac{C}{n} \left( \sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \xi_i^* \right) + \frac{1}{2} (w^T \cdot w) \\
  \text{subject to} \quad y_i - \sum_{j=1}^{n} w_j g_j(x_i) \leq \epsilon + \xi_i \\
  \sum_{j=1}^{n} w_j g_j(x_i) - y_i \leq \epsilon + \xi_i^* \\
  \xi_i^* \geq 0 \\
  \xi_i \geq 0
  \]

§.4. Support Vector Machine for Regression (3)

- **Dual Problem**
  
  \[
  \text{maximize} \quad Q(\alpha, \beta) = -\epsilon \sum_{i=1}^{n} (\alpha_i + \beta_i) - \sum_{i=1}^{n} y_i (\alpha_i - \beta_i) \\
  \quad - \frac{1}{2} \sum_{i,j=1}^{n} (\alpha_i - \beta_i)(\alpha_j - \beta_j) H(x_i, x_j) \\
  \text{subject to} \quad \sum_{i=1}^{n} \alpha_i = \sum_{i=1}^{n} \beta_i, \quad 0 \leq \alpha_i \leq \frac{C}{n}, \quad 0 \leq \beta_i \leq \frac{C}{n}, \quad i = 1, ..., n
  \]

- **The resulting regression function**
  
  \[f(x) = \sum_{i=1}^{n} (\alpha_i^* - \beta_i^*) H(x_i, x)\]

§.5. Summary

- SVM's four principles.
  - Direct solution rather than indirect via density estimation
  - Dimension independent complexity control
  - Nonlinear feature selection
    - Directly incorporated in parameter optimization.
  - Implementation of an inductive principle