Support Vector Machines: Kernels

CS6780 – Advanced Machine Learning
Spring 2015
Thorsten Joachims
Cornell University

Reading: Murphy 14.1, 14.2, 14.4
Schoelkopf/Smola Chapter 7.4, 7.6, 7.8

Support Vector Machines: Kernels

Non-Linear Problems

Problem:
• some tasks have non-linear structure
• no hyperplane is sufficiently accurate
How can SVMs learn non-linear classification rules?

Extending the Hypothesis Space

Idea: add more features

Example:

The separating hyperplane in feature space is degree two polynomial in input space.

Dual SVM Optimization Problem

• Primal Optimization Problem
  minimize: $P(w, b, \xi) = \frac{1}{2} w \cdot w + C \sum_{i=1}^{n} \xi_i$
  subject to: $\sum_{i=1}^{n} y_i (w \cdot x_i + b) \geq 1 - \xi_i \quad \forall i \in [1, n]$

• Dual Optimization Problem
  maximize: $D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j (x_i \cdot x_j)$
  subject to: $\sum_{i=1}^{n} y_i \alpha_i = 0 \quad \forall \alpha_i \leq C$
  $\Phi(\alpha)$

• Theorem: If $w^*$ is the solution of the Primal and $\alpha^*$ is the solution of the Dual, then
  $w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$

Kernels

• Problem:
  – Very many Parameters!
  – Example: Polynomials of degree p over N attributes in input space lead to $O(N^p)$ attributes in feature space!

• Solution:
  – The dual OP depends only on inner products
  $\rightarrow$ Kernel Functions $K(\tilde{a}, \tilde{b}) = \Phi(\tilde{a}) \cdot \Phi(\tilde{b})$

• Example:
  – For $\Phi(\tilde{x}) = (x_1^2, x_2^2, \sqrt{2} x_1, \sqrt{2} x_2, \sqrt{2} x_1 x_2, 1)$ calculating
    $K(\tilde{a}, \tilde{b}) = [\tilde{a} \cdot \tilde{b} + 1]^2$ computes inner product in feature space.

$\rightarrow$ no need to represent feature space explicitly.
SVM with Kernel

- Training:
  \[ \text{maximize: } D(a) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \]
  \[ \text{subject to: } \sum_{i=1}^{n} \alpha_i y_i = 0 \]
  \[ \alpha_i \geq 0 \]

- Classification:
  \[ k(f) = \text{sign} \left( \sum_{i=1}^{n} \alpha_i y_i \Phi(x_i) \cdot \Phi(x) + b \right) \]

- New hypotheses spaces through new Kernels:
  - Linear: \[ K(a, b) = a \cdot b \]
  - Polynomial: \[ K(a, b) = (a \cdot b + 1)^d \]
  - Radial Basis Function: \[ K(a, b) = \exp(-\gamma |a - b|^2) \]
  - Sigmoid: \[ K(a, b) = \tanh(\gamma |a - b| + c) \]

Examples of Kernels

<table>
<thead>
<tr>
<th>Polynomial</th>
<th>Radial Basis Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ K(a, b) = (a \cdot b + 1)^d ]</td>
<td>[ K(a, b) = \exp(-\gamma</td>
</tr>
</tbody>
</table>

What is a Valid Kernel?

Definition [simplified]: Let \( X \) be a nonempty set. A function is a valid kernel in \( X \) if for all \( n \) and all \( x_1, \ldots, x_n \in X \) it produces a Gram matrix

\[ G_{ij} = K(x_i, x_j) \]

that is symmetric

\[ G = G^T \]

and positive semi-definite

\[ \forall a: a^T G a \geq 0 \]

How to Construct Valid Kernels

Theorem: Let \( K_1 \) and \( K_2 \) be valid kernels over \( X \times X, \alpha \geq 0, 0 \leq \lambda \leq 1 \), \( f \) a real-valued function on \( X, \Phi: X \rightarrow \mathbb{R}^m \) with a kernel \( K_3 \) over \( \mathbb{R}^m \times \mathbb{R}^m \), and \( K \) a symmetric positive semi-definite matrix. Then the following functions are valid kernels

\[ K(x, z) = \lambda \ K_1(x, z) + (1-\lambda) \ K_2(x, z) \]
\[ K(x, z) = \alpha K_1(x, z) \]
\[ K(x, z) = K_1(x, z) \ K_2(x, z) \]
\[ K(x, z) = f(x) f(z) \]
\[ K(x, z) = K_3(\Phi(x), \Phi(z)) \]
\[ K(x, z) = x^T K z \]

Kernels for Discrete and Structured Data

Kernels for Sequences: Two sequences are similar, if they have many common and consecutive subsequences.

Example [Lodhi et al., 2000]: For \( 0 \leq \lambda \leq 1 \) consider the following features space

<table>
<thead>
<tr>
<th>( a(cat) )</th>
<th>( a(car) )</th>
<th>( a(bar) )</th>
<th>( b(a) )</th>
<th>( b(t) )</th>
<th>( c-r )</th>
<th>( a-r )</th>
<th>( b-r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ K(\text{cat, cat}) = \lambda^4 \text{, efficient computation via dynamic programming} \]

Kernels for Non-Vectorial Data

- Applications with Non-Vectorial Input Data
  - classify non-vectorial objects
  - Protein classification (\( x \) is string of amino acids)
  - Drug activity prediction (\( x \) is molecule structure)
  - Information extraction (\( x \) is sentence of words)
  - Etc.

- Applications with Non-Vectorial Output Data
  - predict non-vectorial objects
  - Natural Language Parsing (\( y \) is parse tree)
  - Noun-Phrase Co-reference Resolution (\( y \) is clustering)
  - Search engines (\( y \) is ranking)

\[ \Rightarrow \text{Kernels can compute inner products efficiently!} \]
Properties of SVMs with Kernels

- **Expressiveness**
  - SVMs with Kernel can represent any boolean function (for appropriate choice of kernel)
  - SVMs with Kernel can represent any sufficiently “smooth” function to arbitrary accuracy (for appropriate choice of kernel)

- **Computational**
  - Objective function has no local optima (only one global)
  - Independent of dimensionality of feature space

- **Design decisions**
  - Kernel type and parameters
  - Value of C

SVMs for other Problems

- **Multi-class Classification**
  - [Schoelkopf/Smola Book, Section 7.6]

- **Regression**
  - [Schoelkopf/Smola Book, Section 1.6]

- **Outlier Detection**

- **Structured Output Prediction**