Support Vector Machines: Kernels

CS4780/5780 – Machine Learning
Fall 2012
Thorsten Joachims
Cornell University

Reading: Schölkopf/Smola Chapter 7.4, 7.6, 7.8
Cristianini/Shawe-Taylor 3.1, 3.2, 3.3.2, 3.4

Outline

- Transform a linear learner into a non-linear learner
- Kernels can make high-dimensional spaces tractable
- Kernels can make non-vectorial data tractable

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

Learn linear rule in feature space.

Example:

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

Example

• Input Space: \( \tilde{x} = (x_1, x_2) \) (2 attributes)
• Feature Space: \( \Phi(\tilde{x}) = (x_1^2, x_2^2, x_1 x_2, x_1, x_2, 1) \) (6 attributes)

Dual SVM Optimization Problem

- Primal Optimization Problem

\[
\begin{align*}
\text{minimize} & : & P(w, b, \xi) = \frac{1}{2} |w|^2 + C \sum_{i=1}^{n} \xi_i \\
\text{subject to} & : & y_i(w^T \tilde{x}_i + b) \geq 1 - \xi_i \\
& & \sum_{i=1}^{n} \xi_i = \xi
\end{align*}
\]

- Dual Optimization Problem

\[
\begin{align*}
\text{maximize} & : & D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \tilde{x}_i^T \tilde{x}_j \\
\text{subject to} & : & \sum_{i=1}^{n} y_i \alpha_i = 0 \\
& & \sum_{i=1}^{n} \alpha_i \leq C \\
& & \alpha_i \geq 0
\end{align*}
\]

Theorem: If \( w^* \) is the solution of the Primal and \( \alpha^* \) is the solution of the Dual, then

\[
\tilde{w}^* = \sum_{i=1}^{n} y_i \alpha_i \tilde{x}_i
\]
Kernels

• Problem:
  – Very many Parameters! 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 $K(\hat{a}, \hat{b}) = \Phi(\hat{a}) \cdot \Phi(\hat{b})$

• Example:
  – For $\Phi\(\hat{x}\) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_3, x_4)$ calculating $K(\hat{a}, \hat{b}) = \hat{a} \cdot \hat{b} + 1$ computes inner product in feature space.

$\Rightarrow$ no need to represent feature space explicitly.

Examples of Kernels

Polynomial
$K(\hat{a}, \hat{b}) = [\hat{a} \cdot \hat{b} + 1]^2$

Radial Basis Function
$K(\hat{a}, \hat{b}) = \exp(-\gamma |\hat{a} - \hat{b}|^2)$

What is a Valid Kernel?

Definition: 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

$\forall \alpha: \alpha^T G \alpha \geq 0$

How to Construct Valid Kernels

Theorem: Let $K_1$ and $K_2$ be valid Kernels over $X \times X$, $\alpha \geq 0$, $0 \leq \gamma \leq 1$, $f$ 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

$K(x, z) = \lambda \cdot K_3(\phi(x), \phi(z))$

$K(x, z) = x^T K z$

SVM with Kernel

• Training:
  
maximize: $L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K(x_i, x_j)$

subject to: $\sum_{i=1}^{n} y_i \alpha_i = 0$

$\forall i: 0 \leq \alpha_i \leq C$

• Classification:

$h(x) = \text{sign} \left( \sum_{i=1}^{n} y_i \alpha_i \Phi(x_i) \cdot \Phi(x) + b \right) = \text{sign} \left( \sum_{i=1}^{n} y_i \alpha_i K(x_i, x) + b \right)$

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

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 $0 \leq \gamma \leq 1$ consider the following features space

$K(\text{cat}, \text{cat}) = \lambda^2$, 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)
  ➔ 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