Kernels and the Kernel Trick

Martin Hofmann

Reading Club "Support Vector Machines"
Optimization Problem

- maximize:

\[ W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_i \cdot x_j \rangle \]

subject to \( \alpha_i \geq 0, i = 1, \ldots, m \) and \( \sum_{i=1}^{m} \alpha_i y_i = 0 \)

- data not linear separable in input space

\[ \rightarrow \text{map into some feature space where data is linear separable} \]
Mapping Example

- map data points into feature space with some function $\phi$
- e.g.:
  - $\phi : \mathbb{R}^2 \to \mathbb{R}^2$
  - $(x_2, x_2) \to (z_1, z_2, z_3) := (x_1^2, \sqrt{2}x_1x_2, x_2^2)$

hyperplane $\langle w \cdot z \rangle = 0$, as a function of $x$:

$$w_1x_1^2 + w_2\sqrt{2}x_1x_2 + w_3x_2^2 = 0$$
Kernel Trick

- solve maximisation problem using mapped data points

\[ W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_j \alpha_i y_i y_j \langle \phi(x_i) \cdot \phi(x_j) \rangle \]

- Dual Representation of Hyperplane (☉ primal Lagrangian):

\[ f(x) = \langle w \cdot x \rangle + b = \sum \alpha_i y_i \langle x_i \cdot x \rangle \text{ with } w = \sum \alpha_i y_i x_i \]

- weight vector represented only by data points
- only inner product of data points necessary, no coordinates
- kernel function \( K(x_1, x_2) = \langle \phi(x_i) \cdot \phi(x_j) \rangle \)
  \[ \rightarrow \phi \text{ not necessary any more} \]
  \[ \rightarrow \text{possible to operate in any n-dimensional } FS \]
  \[ \rightarrow \text{complexity independent of } FS \]
Example Kernel Trick

\[ \vec{x} = (x_1, x_2) \]
\[ \vec{z} = (z_1, z_2) \]
\[ K(x, z) = \langle \vec{x} \cdot \vec{z} \rangle^2 \]

\[
K(x, z) = \langle \vec{x} \cdot \vec{z} \rangle^2 \\
= (x_1z_1 + x_2z_2)^2 \\
= (x_1^2z_1^2 + 2x_1z_1x_2z_2 + x_2^2z_2^2) \\
= \left\langle (x_1^2, \sqrt{2}x_1x_2, x_2^2) \cdot (z_1^2, \sqrt{2}z_1z_2, z_2^2) \right\rangle \\
= \langle \phi(\vec{x}) \cdot \phi(\vec{z}) \rangle
\]

mapping function \( \phi \) fused in \( K \)

\[ \rightarrow \text{implicit } \phi(\vec{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \]
Typical Kernels

- **Polynomial Kernel**

  \[ K(x, z) = (\langle x \cdot z \rangle + \theta)^d, \quad \text{for} \ d \geq 0 \]

- **Radial Basis Function** (Gaussian Kernel)

  \[ K(x, z) = e^{-\frac{\|x - z\|^2}{2\sigma^2}} \quad \|x\| := \sqrt{\langle x \cdot x \rangle} \]

- **(Sigmoid Kernel)**

  \[ K(x, z) = \tanh(\eta \langle x \cdot z \rangle + \theta) \]

- **Inverse multi-quadric**

  \[ K(x, z) = \frac{1}{\sqrt{\|x - z\|^2 2\sigma^2 + c^2}} \]
Typical Kernels Cont.

- Kernels for Sets - $\chi, \chi'$

$$K - s(\chi, \chi') = \sum_{i=1}^{N_x} \sum_{j=1}^{N_{\chi'}} k(x_i, x'_j)$$

where $k(x_i, x'_j)$ is a kernel on elements in $\chi, \chi'$

- Kernels for strings (Spectral Kernels) and trees

  $\rightarrow$ no one-fits-all kernel
  $\rightarrow$ model search and cross-validation in practice
  $\rightarrow$ low polynomial or RBF a good initial try
Kernel Properties

- **Symmetry**

\[
K(x, z) = \langle \phi(x) \cdot \phi(z) \rangle = \langle \phi(z) \cdot \phi(x) \rangle = K(z, x)
\]

- **Cauchy-Schwarz Inequality**

\[
K(x, z)^2 = \langle \phi(x) \cdot \phi(z) \rangle^2 \leq \| \phi(x) \|^2 \| \phi(z) \|^2
\]

\[
= \langle \phi(x) \cdot \phi(x) \rangle \langle \phi(z) \cdot \phi(z) \rangle
\]

\[
= K(x, x)K(z, z)
\]
Making Kernels from Kernels

- create complex Kernels by combining simpler ones
- Closure Properties:

\[
\begin{align*}
K(x, z) &= c \cdot K_1(x, z) \\
K(x, z) &= c + K_1(x, z) \\
K(x, z) &= K_1(x, z) + K_2(x, z) \\
K(x, z) &= K_1(x, z) \cdot K_2(x, z) \\
K(x, z) &= f(x) \cdot f(z)
\end{align*}
\]

if $K_1$ and $K_2$ are kernels, $\forall f : X \to \mathbb{R}$, and $c > 0$
Gram Matrix

- Kernel function as similarity measure between input objects
- Gram Matrix (Similarity/Kernel Matrix) represents similarities between input vectors
- let $V = \vec{v}_1, \ldots, \vec{v}_n$ a set of input vectors, then the Gram Matrix $K$ is defined as:

$$K = \begin{pmatrix}
\langle \phi(\vec{v}_1) \cdot \phi(\vec{v}_1) \rangle & \ldots & \langle \phi(\vec{v}_1) \cdot \phi(\vec{v}_n) \rangle \\
\langle \phi(\vec{v}_2) \cdot \phi(\vec{v}_1) \rangle & \ddots & \vdots \\
\vdots & \ddots & \\
\langle \phi(\vec{v}_n) \cdot \phi(\vec{v}_1) \rangle & \ldots & \langle \phi(\vec{v}_n) \cdot \phi(\vec{v}_n) \rangle
\end{pmatrix}$$

- $K$ is symmetric and positive semis-definite (positive eigenvalues)
Mercer’s Theorem

• assume:
  • finite input space \( X = \{x_1, \ldots, x_n\} \)
  • symmetric function \( K(x, z) \) on \( X \)
  • Gram Matrix \( K = (K(x_i, x_j))_{i,j=1}^n \)
  • since \( K \) is symmetric there exists an orthogonal matrix \( V \) s.t. \( K = V \Lambda V' \)
  • diagonal \( \Lambda \) containing eigenvalues \( \lambda_t \) of \( K \)
  • and eigenvectors \( v_t = (v_{ti})_{i=1}^n \) as columns of \( V \)
  • all eigenvalues are non-negative and let feature mapping be

\[
\phi : x_i \mapsto (\sqrt{\lambda_i}v_{ti})_{t=1}^n \in \mathbb{R}^n, \ i = 1, \ldots, n.
\]

• then

\[
\langle \phi(x_i) \cdot \phi(x_j) \rangle = \sum_{t=1}^n \lambda_t v_{ti} v_{tj} = (V \Lambda V')_{ij} = K_{ij} = K(x_i, x_j)
\]
Mercer’s Theorem Cont.

- every Gram Matrix is symmetric and positive semi-definite
- every spsd matrix can be regarded as a Kernel Matrix, i.e. as an inner product matrix in some space
- diagonal matrix satisfies Mercer’s criteria, but not good as Gram Matrix
  - self-similarity dominates between-sample similarity
  - represents orthogonal samples
- generalization for infinite input space
  - eigenvectors of the data in can be used to detect directions of maximum variance
  - kernel principal components analysis
Summary

- Kernel calculates dot product of mapped data points without mapping function \( \phi \)
- represented by symmetric, positive semi-definite Gram Matrix
  - fuses information about data and kernel
- standard kernels (cross validation)
- every similarity matrix can be used as kernel (satisfying Mercer’s criteria)
- ongoing research to estimate Kernel Matrix from available data