Chapter 5: Kernel Methods

Mohammed J. Zaki\textsuperscript{1}  Wagner Meira Jr.\textsuperscript{2}

\textsuperscript{1}Department of Computer Science  Rensselaer Polytechnic Institute, Troy, NY, USA

\textsuperscript{2}Department of Computer Science  Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

$k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$
For mining and analysis, it is important to find a suitable data representation. For example, for complex data such as text, sequences, images, and so on, we must typically extract or construct a set of attributes or features, so that we can represent the data instances as multivariate vectors.

Given a data instance \( x \) (e.g., a sequence), we need to find a mapping \( \phi \), so that \( \phi(x) \) is the vector representation of \( x \).

Even when the input data is a numeric data matrix a nonlinear mapping \( \phi \) may be used to discover nonlinear relationships.

The term input space refers to the data space for the input data \( x \) and feature space refers to the space of mapped vectors \( \phi(x) \).
A full kernel matrix is represented by a square matrix of size $n \times n$, where $n$ is the number of samples. The computational complexity for operations involving this matrix is $O(n^2)$. The matrix is denoted as $K$ and includes terms $K_{ij}$ for $i, j = 1, 2, ..., n$. The diagonal terms correspond to self-kernel values for each sample, and the off-diagonal terms represent kernel values between different samples.
Sequence-based Features

Consider a dataset of DNA sequences over the alphabet $\Sigma = \{A, C, G, T\}$. One simple feature space is to represent each sequence in terms of the probability distribution over symbols in $\Sigma$. That is, given a sequence $x$ with length $|x| = m$, the mapping into feature space is given as

$$\phi(x) = \{P(A), P(C), P(G), P(T)\}$$

where $P(s) = \frac{n_s}{m}$ is the probability of observing symbol $s \in \Sigma$, and $n_s$ is the number of times $s$ appears in sequence $x$.

For example, if $x = ACAGCAGTA$, with $m = |x| = 9$, since $A$ occurs four times, $C$ and $G$ occur twice, and $T$ occurs once, we have

$$\phi(x) = \left(\frac{4}{9}, \frac{2}{9}, \frac{2}{9}, \frac{1}{9}\right) = (0.44, 0.22, 0.22, 0.11)$$

We can compute larger feature spaces by considering, for example, the probability distribution over all substrings or words of size up to $k$ over the alphabet $\Sigma$. 
Consider the mapping $\phi$ that takes as input a vector $\mathbf{x} = (x_1, x_2)^T \in \mathbb{R}^2$ and maps it to a “quadratic” feature space via the nonlinear mapping

$$
\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)^T \in \mathbb{R}^3
$$

For example, the point $\mathbf{x} = (5.9, 3)^T$ is mapped to the vector

$$
\phi(\mathbf{x}) = (5.9^2, 3^2, \sqrt{2} \cdot 5.9 \cdot 3)^T = (34.81, 9, 25.03)^T
$$

We can then apply well-known linear analysis methods in the feature space.
Kernel Method

Let $\mathcal{I}$ denote the input space, which can comprise any arbitrary set of objects, and let $\mathbf{D} = \{x_i\}_{i=1}^n \subset \mathcal{I}$ be a dataset comprising $n$ objects in the input space. Let $\phi: \mathcal{I} \rightarrow \mathcal{F}$ be a mapping from the input space $\mathcal{I}$ to the feature space $\mathcal{F}$.

Kernel methods avoid explicitly transforming each point $x$ in the input space into the mapped point $\phi(x)$ in the feature space. Instead, the input objects are represented via their pairwise similarity values comprising the $n \times n$ kernel matrix, defined as

$$
K = \begin{pmatrix}
K(x_1, x_1) & K(x_1, x_2) & \cdots & K(x_1, x_n) \\
K(x_2, x_1) & K(x_2, x_2) & \cdots & K(x_2, x_n) \\
\vdots & \vdots & \ddots & \vdots \\
K(x_n, x_1) & K(x_n, x_2) & \cdots & K(x_n, x_n)
\end{pmatrix}
$$

$K: \mathcal{I} \times \mathcal{I} \rightarrow \mathbb{R}$ is a kernel function on any two points in input space, which should satisfy the condition

$$
K(x_i, x_j) = \phi(x_i)^T \phi(x_j)
$$

Intuitively, we need to be able to compute the value of the dot product using the original input representation $x$, without having recourse to the mapping $\phi(x)$.
Let $\phi(x) \rightarrow x$ be the *identity kernel*. This leads to the *linear kernel*, which is simply the dot product between two input vectors:

$$\phi(x)^T \phi(y) = x^T y = K(x, y)$$

For example, if $x_1 = (5.9 \ 3)^T$ and $x_2 = (6.9 \ 3.1)^T$, then we have

$$K(x_1, x_2) = x_1^T x_2 = 5.9 \times 6.9 + 3 \times 3.1 = 40.71 + 9.3 = 50.01$$
Kernel Trick

Many data mining methods can be *kernelized* that is, instead of mapping the input points into feature space, the data can be represented via the $n \times n$ kernel matrix $K$, and all relevant analysis can be performed over $K$.

This is done via the *kernel trick*, that is, show that the analysis task requires only dot products $\phi(x_i)^T \phi(x_j)$ in feature space, which can be replaced by the corresponding kernel $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ that can be computed efficiently in input space.

Once the kernel matrix has been computed, we no longer even need the input points $x_i$, as all operations involving only dot products in the feature space can be performed over the $n \times n$ kernel matrix $K$. 
A function $K$ is called a **positive semidefinite kernel** if and only if it is symmetric:

$$K(x_i, x_j) = K(x_j, x_i)$$

and the corresponding kernel matrix $K$ for any subset $D \subset \mathcal{I}$ is positive semidefinite, that is,

$$a^T K a \geq 0, \text{ for all vectors } a \in \mathbb{R}^n$$

which implies that

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j K(x_i, x_j) \geq 0, \text{ for all } a_i \in \mathbb{R}, i \in [1, n]$$
If \( K(x_i, x_j) \) represents the dot product \( \phi(x_i)^T \phi(x_j) \) in some feature space, then \( K \) is a positive semidefinite kernel.

First, \( K \) is symmetric since the dot product is symmetric, which also implies that \( K \) is symmetric.

Second, \( K \) is positive semidefinite because

\[
\begin{align*}
    a^T K a &= \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j K(x_i, x_j) \\
    &= \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j \phi(x_i)^T \phi(x_j) \\
    &= \left( \sum_{i=1}^{n} a_i \phi(x_i) \right)^T \left( \sum_{j=1}^{n} a_j \phi(x_j) \right) \\
    &= \left\| \sum_{i=1}^{n} a_i \phi(x_i) \right\|^2 \geq 0
\end{align*}
\]
Empirical Kernel Map

We now show that if we are given a positive semidefinite kernel \( K: \mathcal{I} \times \mathcal{I} \to \mathbb{R} \), then it corresponds to a dot product in some feature space \( \mathcal{F} \).

Define the map \( \phi \) as follows:

\[
\phi(x) = \left( (K(x_1, x), K(x_2, x), \ldots, K(x_n, x)) \right)^T \in \mathbb{R}^n
\]

The empirical kernel map is defined as

\[
\phi(x) = K^{-1/2} \cdot \left( (K(x_1, x), K(x_2, x), \ldots, K(x_n, x)) \right)^T \in \mathbb{R}^n
\]

so that the dot product yields

\[
\phi(x_i)^T \phi(x_j) = \left( K^{-1/2} K_i \right)^T \left( K^{-1/2} K_j \right)
\]

\[
= K_i^T (K^{-1/2}K^{-1/2}) K_j
\]

\[
= K_i^T K^{-1} K_j
\]

where \( K_i \) is the \( i \)\(^{th} \) column of \( K \).

Over all pairs of mapped points, we have

\[
\left\{K_i^T K^{-1} K_j\right\}_{i,j=1}^n = K K^{-1} K = K
\]
The Mercer kernel map also corresponds to a dot product in feature space. Since \( K \) is a symmetric positive semidefinite matrix, it has real and non-negative eigenvalues. It can be decomposed as follows:

\[
K = U \Lambda U^T
\]

where \( U \) is the orthonormal matrix of eigenvectors \( u_i = (u_{i1}, u_{i2}, \ldots, u_{in})^T \in \mathbb{R}^n \) (for \( i = 1, \ldots, n \)), and \( \Lambda \) is the diagonal matrix of eigenvalues, with both arranged in non-increasing order of the eigenvalues \( \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0 \):

The Mercer map \( \phi \) is given as

\[
\phi(x_i) = \sqrt{\Lambda} U_i
\]

where \( U_i \) is the \( i \)th row of \( U \).

The kernel value is simply the dot product between scaled rows of \( U \):

\[
\phi(x_i)^T \phi(x_j) = \left( \sqrt{\Lambda} U_i \right)^T \left( \sqrt{\Lambda} U_j \right) = U_i^T \Lambda U_j
\]
Polynomial Kernel

Polynomial kernels are of two types: homogeneous or inhomogeneous. Let \( \mathbf{x}, \mathbf{y} \in \mathbb{R}^d \). The (inhomogeneous) polynomial kernel is defined as

\[
K_q(\mathbf{x}, \mathbf{y}) = (\phi(\mathbf{x})^T \phi(\mathbf{y}))^q = (c + \mathbf{x}^T \mathbf{y})^q
\]

where \( q \) is the degree of the polynomial, and \( c \geq 0 \) is some constant. When \( c = 0 \) we obtain the homogeneous kernel, comprising only degree \( q \) terms. When \( c > 0 \), the feature space is spanned by all products of at most \( q \) attributes. This can be seen from the binomial expansion

\[
K_q(\mathbf{x}, \mathbf{y}) = (c + \mathbf{x}^T \mathbf{y})^q = \sum_{k=1}^{q} \binom{q}{k} c^{q-k} (\mathbf{x}^T \mathbf{y})^k
\]

The most typical cases are the linear (with \( q = 1 \)) and quadratic (with \( q = 2 \)) kernels, given as

\[
K_1(\mathbf{x}, \mathbf{y}) = c + \mathbf{x}^T \mathbf{y} \\
K_2(\mathbf{x}, \mathbf{y}) = (c + \mathbf{x}^T \mathbf{y})^2
\]
\[ d = 3 \]

\[
\begin{array}{cccc}
X_1 & X_2 & X_3 \\
\hline
0 & 1 & 0.5 & 2 \\
1 & 2 & 3 \\
\end{array}
\]

\[ c = 1 \]

\[ K(x, x') = \left( 1 + \frac{x_1^T x_2}{\alpha} \right)^2 \]

\[ = (1 + 7.1)^2 = (8.1)^2 \leq 67. \\
\]

\[ q = 0 \quad q = 1 \quad q = 2 \]

\[
\begin{array}{cccccc}
1 & x_1 & x_2 & x_3 & x_1^2 & x_2^2 & x_3^2 & x_1x_2 & x_1x_3 & x_2x_3 \\
\hline
1 & 1 & 2 & 3 & 1 & 4 & 9 & 2 & 3 & 6 \\
\end{array}
\]

\[ \phi(x_1) \phi(x_2) \]
The Gaussian kernel, also called the Gaussian radial basis function (RBF) kernel, is defined as

$$K(x, y) = \exp\left\{ -\frac{\|x - y\|^2}{2\sigma^2} \right\}$$

where $\sigma > 0$ is the spread parameter that plays the same role as the standard deviation in a normal density function.

Note that $K(x, x) = 1$, and further that the kernel value is inversely related to the distance between the two points $x$ and $y$.

A feature space for the Gaussian kernel has infinite dimensionality.
Basic data analysis tasks that can be performed solely via kernels, without instantiating $\phi(x)$.

**Norm of a Point:** We can compute the norm of a point $\phi(x)$ in feature space as follows:

$$\|\phi(x)\|^2 = \phi(x)^T \phi(x) = K(x, x)$$

which implies that $\|\phi(x)\| = \sqrt{K(x, x)}$.

**Distance between Points:** The distance between $\phi(x_i)$ and $\phi(x_j)$ is

$$\|\phi(x_i) - \phi(x_j)\|^2 = \|\phi(x_i)\|^2 + \|\phi(x_j)\|^2 - 2\phi(x_i)^T \phi(x_j)$$

$$= K(x_i, x_i) + K(x_j, x_j) - 2K(x_i, x_j)$$

which implies that

$$\|\phi(x_i) - \phi(x_j)\| = \sqrt{K(x_i, x_i) + K(x_j, x_j) - 2K(x_i, x_j)}$$
Basic Kernel Operations in Feature Space

**Kernel Value as Similarity:** We can rearrange the terms in

\[
\|\phi(x_i) - \phi(x_j)\|^2 = K(x_i, x_i) + K(x_j, x_j) - 2K(x_i, x_j)
\]

to obtain

\[
\frac{1}{2} \left( \|\phi(x_i)\|^2 + \|\phi(x_j)\|^2 - \|\phi(x_i) - \phi(x_j)\|^2 \right) = K(x_i, x_j) = \phi(x_i)^T \phi(x_j)
\]

The more the distance \(\|\phi(x_i) - \phi(x_j)\|\) between the two points in feature space, the less the kernel value, that is, the less the similarity.

**Mean in Feature Space:** The mean of the points in feature space is given as

\[
\mu_\phi = \frac{1}{n} \sum_{i=1}^{n} \phi(x_i)
\]

Thus, we cannot compute it explicitly. However, the the squared norm of the mean is:

\[
\|\mu_\phi\|^2 = \mu_\phi^T \mu_\phi = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} K(x_i, x_j)
\]

(1)

The squared norm of the mean in feature space is simply the average of the values in the kernel matrix \(K\).
Input space

\[
M_\phi = \frac{1}{n} \sum_{i=1}^{n} \phi(x_i)
\]

\[
M_\phi^T M_\phi = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \phi(x_i)^T \phi(x_j) \cdot k(x_i, x_j)
\]
**Total Variance in Feature Space:** The total variance in feature space is obtained by taking the average squared deviation of points from the mean in feature space:

\[
\sigma_{\Phi}^2 = \frac{1}{n} \sum_{i=1}^{n} \| \phi(x_i) - \mu_{\phi} \|^2 = \frac{1}{n} \sum_{i=1}^{n} K(x_i, x_i) - \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} K(x_i, x_j)
\]

**Centering in Feature Space** We can center each point in feature space by subtracting the mean from it, as follows:

\[
\hat{\phi}(x_i) = \phi(x_i) - \mu_{\phi}
\]

The kernel between centered points is given as

\[
\hat{K}(x_i, x_j) = \hat{\phi}(x_i)^T \hat{\phi}(x_j)
\]

More compactly, we have:

\[
\hat{K} = \left( I - \frac{1}{n} 1_{n \times n} \right) K \left( I - \frac{1}{n} 1_{n \times n} \right) \]

where \(1_{n \times n}\) is the \(n \times n\) matrix of ones.
$D \rightarrow \phi \rightarrow K = \frac{n \times n}{\phi(x_i)^T \phi(x_j)} \rightarrow \phi(x_i) - \mu \beta \rightarrow K^c = (\phi(x_i) - \mu \beta)^T (\phi(x_j) - \mu \beta)$
Normalizing in Feature Space: The dot product between normalized points in feature space corresponds to the cosine of the angle between them

\[ \phi_n(x_i)^T \phi_n(x_j) = \frac{\phi(x_i)^T \phi(x_j)}{\| \phi(x_i) \| \cdot \| \phi(x_j) \|} = \cos \theta \]

If the mapped points are both centered and normalized, then a dot product corresponds to the correlation between the two points in feature space.

The normalized kernel matrix, \( K_n \), can be computed using only the kernel function \( K \), as

\[ K_n(x_i, x_j) = \frac{\phi(x_i)^T \phi(x_j)}{\| \phi(x_i) \| \cdot \| \phi(x_j) \|} = \frac{K(x_i, x_j)}{\sqrt{K(x_i, x_i) \cdot K(x_j, x_j)}} \]

\( K_n \) has all diagonal elements as 1.
Spectrum Kernel for Strings

Given alphabet $\Sigma$, the $l$-spectrum feature map is the mapping $\phi: \Sigma^* \rightarrow \mathbb{R}^{|\Sigma|^l}$ from the set of substrings over $\Sigma$ to the $|\Sigma|^l$-dimensional space representing the number of occurrences of all possible substrings of length $l$, defined as

$$\phi(x) = \left( \cdots, \#(\alpha), \cdots \right)_{\alpha \in \Sigma^l}^T$$

where $\#(\alpha)$ is the number of occurrences of the $l$-length string $\alpha$ in $x$.

The (full) spectrum map considers all lengths from $l = 0$ to $l = \infty$, leading to an infinite dimensional feature map $\phi : \Sigma^* \rightarrow \mathbb{R}^\infty$:

$$\phi(x) = \left( \cdots, \#(\alpha), \cdots \right)_{\alpha \in \Sigma^*}^T$$

where $\#(\alpha)$ is the number of occurrences of the string $\alpha$ in $x$.

The ($l$-)spectrum kernel between two strings $x_i, x_j$ is simply the dot product between their ($l$-)spectrum maps:

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

The (full) spectrum kernel can be computed efficiently via suffix trees in $O(n + m)$ time for two strings of length $n$ and $m$. 
Diffusion Kernels on Graph Nodes

Let $S$ be some symmetric similarity matrix between nodes of a graph $G = (V, E)$. For instance, $S$ can be the (weighted) adjacency matrix $A$ or the Laplacian matrix $L = A - \Delta$ (or its negation), where $\Delta$ is the degree matrix for an undirected graph $G$, defined as $\Delta(i, i) = d_i$ and $\Delta(i, j) = 0$ for all $i \neq j$, and $d_i$ is the degree of node $i$.

**Power Kernels:** Summing up the product of the base similarities over all $l$-length paths between two nodes, we obtain the $l$-length similarity matrix $S^{(l)}$, which is simply the $l$th power of $S$, that is,

$$S^{(l)} = S^l$$

Even path lengths lead to positive semidefinite kernels, but odd path lengths are not guaranteed to do so, unless the base matrix $S$ is itself a positive semidefinite matrix.

Power kernel $K$ can be obtained via the eigen-decomposition of $S^l$:

$$K = S^l = (U \Lambda U^T)^l = U (\Lambda^l) U^T$$
The exponential diffusion kernel we can obtain a new kernel between nodes of a graph by paths of all possible lengths, but damps the contribution of longer paths

\[
K = \sum_{l=0}^{\infty} \frac{1}{l!} \beta^l S^l
\]

\[
= I + \beta S + \frac{1}{2!} \beta^2 S^2 + \frac{1}{3!} \beta^3 S^3 + \cdots
\]

\[
= \exp\{\beta S\}
\]

where \(\beta\) is a damping factor, and \(\exp\{\beta S\}\) is the matrix exponential. The series on the right hand side above converges for all \(\beta \geq 0\).

Substituting \(S = U \Lambda U^T\) the kernel can be computed as

\[
K = I + \beta S + \frac{1}{2!} \beta^2 S^2 + \cdots
\]

\[
= U \begin{pmatrix}
\exp\{\beta \lambda_1\} & 0 & \cdots & 0 \\
0 & \exp\{\beta \lambda_2\} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \exp\{\beta \lambda_n\}
\end{pmatrix} U^T
\]

where \(\lambda_i\) is an eigenvalue of \(S\).
The von Neumann diffusion kernel is defined as

\[ K = \sum_{l=0}^{\infty} \beta^l S^l \]

where \( \beta \geq 0 \). Expanding and rearranging the terms, we obtain

\[ K = (I - \beta S)^{-1} \]

The kernel is guaranteed to be positive semidefinite if \( |\beta| < 1/\rho(S) \), where \( \rho(S) = \max_i \{|\lambda_i|\} \) is called the spectral radius of \( S \), defined as the largest eigenvalue of \( S \) in absolute value.
Graph Diffusion Kernel: Example

Adjacency and degree matrices are given as

\[ A = \begin{pmatrix}
0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0
\end{pmatrix} \quad \Delta = \begin{pmatrix}
2 & 0 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 & 0 \\
0 & 0 & 3 & 0 & 0 \\
0 & 0 & 0 & 3 & 0 \\
0 & 0 & 0 & 0 & 2
\end{pmatrix} \]
Let the base similarity matrix $S$ be the negated Laplacian matrix $S = -L = A - D = \begin{pmatrix} -2 & 0 & 1 & 1 & 0 \\ 0 & -2 & 1 & 0 & 1 \\ 1 & 1 & -3 & 1 & 0 \\ 1 & 0 & 1 & -3 & 1 \\ 0 & 1 & 0 & 1 & -2 \end{pmatrix}$

The eigenvalues of $S$ are as follows:

$\lambda_1 = 0 \quad \lambda_2 = -1.38 \quad \lambda_3 = -2.38 \quad \lambda_4 = -3.62 \quad \lambda_5 = -4.62$

and the eigenvectors of $S$ are

$U = \begin{pmatrix} u_1 & u_2 & u_3 & u_4 & u_5 \\ 0.45 & -0.63 & 0.00 & 0.63 & 0.00 \\ 0.45 & 0.51 & -0.60 & 0.20 & -0.37 \\ 0.45 & -0.20 & -0.37 & -0.51 & 0.60 \\ 0.45 & -0.20 & 0.37 & -0.51 & -0.60 \\ 0.45 & 0.51 & 0.60 & 0.20 & 0.37 \end{pmatrix}$
Graph Diffusion Kernel: Example

Assuming $\beta = 0.2$, the exponential diffusion kernel matrix is given as

$$K = \exp\{0.2S\} = U \begin{pmatrix} \exp\{0.2\lambda_1\} & 0 & \cdots & 0 \\ 0 & \exp\{0.2\lambda_2\} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \exp\{0.2\lambda_n\} \end{pmatrix} U^T$$

$$= \begin{pmatrix} 0.70 & 0.01 & 0.14 & 0.14 & 0.01 \\ 0.01 & 0.70 & 0.13 & 0.03 & 0.14 \\ 0.14 & 0.13 & 0.59 & 0.13 & 0.03 \\ 0.14 & 0.03 & 0.13 & 0.59 & 0.13 \\ 0.01 & 0.14 & 0.03 & 0.13 & 0.70 \end{pmatrix}$$

Assuming $\beta = 0.2$, the von Neumann kernel is given as

$$K = U(I - 0.2\Lambda)^{-1}U^T = \begin{pmatrix} 0.75 & 0.02 & 0.11 & 0.11 & 0.02 \\ 0.02 & 0.74 & 0.10 & 0.03 & 0.11 \\ 0.11 & 0.10 & 0.66 & 0.10 & 0.03 \\ 0.11 & 0.03 & 0.10 & 0.66 & 0.10 \\ 0.02 & 0.11 & 0.03 & 0.10 & 0.74 \end{pmatrix}$$