CS419M

Lecture 14: Kernel Methods

10/03/2023

Lecturer: Abir De *Scribe: Groups 3 & 4*

1 Introduction to Kernel Methods

Most of the time real world data has a non-linear decision boundary, which is difficult to learn using simple models as they fail to capture the non-linear relationships between input features and the classes. Kernel methods provide a solution to this problem by mapping the input features to a higher dimensional space where a linear classifier can separate the data effectively. In this method, we use a kernel function which computes the similarity in the higher dimensional space without explicitly computing the feature vectors (less computational cost). Here's an example: (Assuming x,y are a 2D vectors)

$$\begin{aligned} K(x,y) &= (x^T y)^2 \\ &= (x_1 y_1 + x_2 y_2)^2 \\ &= x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2 \\ &= \langle (x_1^2, \sqrt{2}x_1 x_2, x_2^2), (y_1^2, \sqrt{2}y_1 y_2, y_2^2) \rangle \end{aligned}$$

Here (x_1, x_2) in 2D space is mapped to $(x_1^2, \sqrt{2}x_1x_2, x_2^2)$ in 3D. Note : For a given K(x,y) the mapping to higher dimension is not unique. e.g. we can also find a mapping from $\mathbb{R}^2 \longrightarrow \mathbb{R}^4$ with the same function $(x^T y)^2$ i.e. $(x_1, x_2) \longrightarrow (x_1^2, x_2^2, x_1x_2, x_1x_2)$

2 Kernel functions

A function x x $x \longrightarrow \mathbb{R}$ is a kernel iff for any $x_1, x_2, ...x_m \in \mathbb{R}$ and any m $\in \mathbb{N}$

- K(x_i, x_j) = K(x_j, x_i) i.e. the kernel function must be symmetric
- the Gram matrix $\mathbb{G}$ given by $G_{ij} = K(x_i, x_j)$ where $x_1 ... x_n$ are the data points must be positive semi definite.

Any real symmetric matrix $\mathbb{G}$ can be written as UDU^T where D is a diagonal matrix with positive entries (as the eigen values are positive). Thus D can be written as $\sqrt{D}\sqrt{D}$ and $\mathbb{G}$ = U$\sqrt{D}\sqrt{D}U^T$ = (U$\sqrt{D}$)(U$\sqrt{D}$)^T which means that there always exists a mapping ϕ = U$\sqrt{D}$ to a higher dimension.

2.1 Examples

- K(x_i, x_j) = $x_i x_j$ (where x,y are real numbers)
  This is obviously symmetric.
  For positive semi definiteness if we assume a to be some column vector [$a_1 ... a_n$], then $a^T Ga$ evaluates to:
  $$\sum a_i a_j K(x_i, x_j) = \sum_{i,j} a_i a_j x_i x_j = (\sum a_i x_i)^2 \geq 0$$
  The intermediate step is evident from the expansion.

  Similarly if x_i, x_j are d-dimensional vectors i.e. $x \in \mathbb{R}^d$ and the inner product is defined as K(x_i, x_j) = $\langle x_i x_j \rangle$, $\sum a_i a_j \langle x_i x_j \rangle = || \sum a_i x_i||^2 \geq 0$. A similar argument can be given if we apply a transformation $x \longrightarrow \phi(x)$ and define the inner product as $\langle \phi(x_i)\phi(x_j) \rangle$.

- Polynomial Kernel : $(x^T y)^d$ (x,y $\in \mathbb{R}^n$)
  From this function the mapping can be obtained using multinomial expansion as shown:
  $$(x^T y)^d = (\sum_{i=1}^{n} x_i y_i)^d = \sum_{k_{1,2,...k_n}} \binom{d}{k_1 k_2 ... k_n} x_1^{k_1} x_2^{k_2} .. x_n^{k_n} y_1^{k_1} y_2^{k_2} .. y_n^{k_n}$$
  The above summation can be represented as an inner product $\langle \phi(x), \phi(y) \rangle$

If K_1, K_2 are two kernel functions such that:

$$K_1 = \phi_1^T \phi_1, K_2 = \phi_2^T \phi_2$$

then K = $K_1 + K_2$ is also a kernel function where the equivalent transformation to the higher dimension, ϕ is given by:

$$\begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix}$$

3 Applications of Kernel methods

3.1 Support Vector Regression

Consider the loss function:

$$||w||^2 + \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

Upon applying the mapping $x \longrightarrow \phi(x)$, computing $w^T \phi(x)$ is not always computationally feasible as ϕ can be of very large dimensions. In the previous lecture we saw that for any loss function of the form:

$$\ell((w^T x_i), y_i) + \lambda ||w||^2$$

the optimal ω ($\omega*$) can be written as $\sum_1^N \alpha_i x_i y_i$ for any α_i not necessarily the Lagrangian multiplier which was the case for SVM. Upon applying the transformation the loss function becomes:

$$||w||^2 + \sum_{i=1}^{n}(y_i - w^T\phi(x_i))^2$$

and the optimal value of w = $\sum_1^N \alpha_i\phi(x_i)$ (Assume y_i is included in α_i)
Substituting the optimal value of w and using the fact that K(x_i, x_j) = $\langle\phi(x_i), \phi(x_j)\rangle = \phi(x_i)^T\phi(x_j)$, the problem translates to finding optimal α_i such that the following loss function is minimised:

$$\sum \alpha_i\alpha_j K(x_i, x_j) + \sum_{i=1}^{N}(y_i - \sum_{j=1}^{N}\alpha_j K(x_i, x_j))^2$$

Thus after getting optimal α_i, the optimal value of w can also be obtained. Now for testing a new vector x_{test} instead of computing $w^T x_{test}$ we can simply compute $\sum \alpha_i K(x_i, x_{test})$. This method can be used for any loss function which includes x_i^T, x_j as it can be replaced with $K(x_i, x_j)$ and non-linear data can be fitted with linear classifier using kernel functions.

3.2 Principal Component Analysis

In this method of dimensionality reduction we project find the eigenvectors and thus the eigenvalues of the covariance matrix. Then, we sort the eigenvalues and for reduction to k-dimensions we pick the top k vectors each of which acts as a feature. Next, we project the entire dataset onto these k eigenvectors such that the variance is maximum i.e. minimum information is lost. However, in some cases, the data may not be well-separated in the original feature space, and linear methods like PCA may not be effective. The kernel trick in PCA allows us to perform a nonlinear transformation of the data into a higher-dimensional space, where the data may be more easily separated. We can then apply PCA to this transformed data to obtain a set of principal components that capture the most variation in the transformed data.

4 Inner Product Space

4.1 Definition

An inner product space is a vector space $\mathcal{V}$ over field $\mathbb{R}$ together with an inner product, that is a map $\langle\cdot, \cdot\rangle : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$

$$\langle X,\, Y\rangle = x_1y_1 + x_2y_2 + \cdots + x_ny_n.$$

that satisfies the following properties for all vectors $X, Y, Z \in \mathcal{V}$ and all scalars $a, b \in \mathbb{R}$

- $\langle X,\, X\rangle \geq 0$. $\langle X,\, X\rangle = 0$ iff $X = 0$.
- $\langle Y,\, X\rangle = \langle X,\, Y\rangle$ (symmetry)
- $\langle aX + bY,\, Z\rangle = a\langle X,\, Z\rangle + b\langle Y,\, Z\rangle$ (linearity)

4.2 Inner Product Space of functions

Consider a function $f : \mathcal{X} \to \mathcal{V}$ from input space $\mathcal{X}$ to vector space $\mathcal{V}$.

$$y = f(x) = \sum_{\alpha \in \mathrm{A}} \alpha \phi_\alpha(x)$$

where $f(x)$ is a non-linear function. where $\phi_\alpha : \mathcal{X} \to \mathcal{V}$ belongs to the class of feature maps. If the indexing set A is finite, we can write

$$y = f(x) = w^T \phi(x)$$

for appropriate w and ϕ. But if the indexing set A is infinite, we need to find another way to compute f.
Suppose there exists a Taylor Expansion of f. Let us represent y in terms of the kernel function

$$y = f(x) = \sum_{i=1}^{N} \frac{\alpha_i y_i K(x_i, x)}{N}$$

where $K(\cdot, \cdot)$ is the kernel(similarity function) and sum is over all training examples.
Consider the space of functions spanned by the set $\{K(x_i, \cdot) \mid i \in [n]\}$. Observe f lies in this vector space. We define an inner product on this vector space.

$$\langle K(x_i, \cdot),\, K(x_j, \cdot) \rangle = K(x_i, x_j)$$

Note that we are only defining the inner product on the spanning set and the definition on any vector which is a linear combination of the spanning set follows from linearity of inner product. Further, observe that symmetry of inner product follows from symmetry of the kernel function. In order to ensure that the induced norm is positive for non-zero vectors, we need further restrictions on the kernel. Consider $g \neq 0$, then $\langle g,\, g \rangle > 0$

$$\begin{aligned} \langle g,\, g \rangle &= \langle \sum_i a_i K(x_i, \cdot),\, \sum_j a_j K(x_j, \cdot) \rangle \\ &= \sum_{(i,j)} a_i a_j K(x_i, x_j) \\ &= a^T K a > 0 \end{aligned}$$

where g belongs to the space of functions described above, K is a matrix defined as $K_{(i,j)} = K(x_i, x_j)$. In other words, K is positive definite.

Now,

$$\begin{aligned}
\langle f,\, K(x,\cdot)\rangle &= \langle\sum_{i=1}^{N}\frac{\alpha_i y_i K(x_i,\cdot)}{N},\, K(x,\cdot)\rangle \\
&= \sum_{i=1}^{N}\frac{\alpha_i y_i \langle K(x_i,\cdot),\, K(x,\cdot)\rangle}{N} \\
&= \sum_{i=1}^{N}\frac{\alpha_i y_i K(x_i,x)}{N} = f(x)
\end{aligned}$$

A key point to note is that we don't need to explicitly compute the feature map as long as we have the kernel. This solves the issue with infinite dimensional feature space.