CS419-Intro to ML

Lecture 15: Kernel Methods

15 March 2023

Lecturer: Abir De *Scribe: Group 5,6*

In the previous lecture, we have had an introduction on the basics of Kernel and its properties. Then we went on to see some populars Kernels in the tutorial session also. In this lecture we go on to build up on Kernels and solve SVM problems through Kernel Methods.

1 Introduction

Let us take a short recap of what we studied in the last lecture and then continue on to the todays content for a better flow.
Before we understand Kernels, we need to understand the inner product and its propertise in a precise manner.

1.1 Inner Product Space

An inner product space over reals is a vector space $\mathcal{V}$ and an inner product, which is a mapping

$$\langle \cdot, \cdot \rangle : \mathcal{V} \times \mathcal{V} \to \mathcal{R}$$

Following are the properties of inner product $\forall x, y, z \in \mathcal{V}$ and $a, b \in \mathcal{R}$:-

- Symmetry: $\langle x, y \rangle = \langle y, x \rangle$
- Linearity: $\langle ax + by, z \rangle = a\langle x, z \rangle + b\langle y, z \rangle$
- Positive-definiteness: $\langle x, x \rangle \geq 0$ and $\langle x, x \rangle = 0 \iff x = 0$

1.2 Kernel

The kernel by definition avoids the explicit mapping that is needed to get linear learning algorithms to learn a nonlinear function or decision boundary.

For all x and x' in the input space Φ certain functions $K(x, x')$ can be expressed as an inner product in another space Ψ. The function

$$K : \Phi \times \Phi \to \mathbb{R}$$

$\forall x, x' \in \mathcal{X}$,

$$K(x, x') = \langle \phi(x), \phi(x') \rangle$$

The Gram matrix is then defined as $G_{i,j} = K(x_i, x_j)$.
The two necessary condition for a kernel are:

- Symmetric: $K(x_i, x_j) = K(x_j, x_i)$
- Positive Semi Definite : $G_{i,j} >= 0$

From the above understanding we can confirm that for a set of N data points there exist N Kernel Mappings.

2 SVM Objective Function

For the SVM mechanism we had already learnt the following objective function formulation:

$$\underset{w}{min} \;\; l(\{w^T \phi(x_i)\}_{i \in D}, \{y_i\}_{i \in D}) + \lambda R(||w||) \tag{1}$$

where $l : \mathbb{R}^{|D|} \rightarrow \mathbb{R}$ is an arbitrary function and $R : \mathbb{R}_+ \rightarrow \mathbb{R}$ is a monotonically non-decreasing Regularization function.

We found out the optimal ω as:

$$w* = \sum_{i=1}^{|D|} \alpha_i \phi(x_i)$$

Form of f is

$$\begin{aligned} f(x) &= {w^*}^T \phi(x_i) \\ &= \sum_{i=1}^{|D|} \alpha_i \phi^T(x_i) \phi(x) \end{aligned}$$

Here $\phi^T(x_i)\phi(x)$ is like a similarity measure If $\phi(\cdot)$ is ∞-dimensional, we can write it as

$$f(x) = \sum_{i=1}^{|D|} \alpha_i \sum_{j=0}^{\infty} \phi(x_i)[j] \phi(x)[j]$$

Thus, if $\phi(\cdot)$ is ∞-dimensional, it is very high computational task to compute ${w^*}^T \phi(x_i)$ as w as it has the same dimension as ϕ. So, we try to represent the objective function in functional form or through the kernel formulation so that we would not have to do such computations.

3 Kernel Formualation

Now as in the last topic, we got stuck in the case when ω and $\phi(x_i)$ are of infinite dimension, it gets impossible to find these vectors.
Hence here we try to formulate the objective function with the use of Kernels that we studied earlier.

Writing $w = \sum_{j=1}^{|D|} \alpha_j \phi(x_j)$,
we have that for all i

$$\langle w, \phi(x_i)\rangle = \langle \sum_{j=1}^{|D|} \alpha_j \phi(x_j), \phi(x_i\rangle = \sum_{j=1}^{|D|} \alpha_j \langle \phi(x_j), \phi(x_i)\rangle.$$

Similarly,

$$||w||^2 = \langle \sum_{j=1}^{|D|} \alpha_j \phi(x_j), \sum_{j=1}^{|D|} \alpha_j \phi(x_j)\rangle = \sum_{i,j=1}^{|D|} \alpha_i \alpha_j \langle \phi(x_i), \phi(x_j)\rangle.$$

Let $K(x, x') = \langle \phi(x), \phi(x')\rangle$ be a function that implements the kernel function with respect to the feature space. Hence, instead of solving Equation 1, we can solve the equivalent problem

$$\underset{\alpha\in\mathbb{R}^{|D|}}{min}\, l(\{\sum_{j=1}^{|D|} \alpha_j K(x_j, x_i)\}_{i\in D}, \{y_i\}_{i\in D}) + \lambda R(\sqrt{\sum_{i,j=1}^{|D|} \alpha_i \alpha_j K(x_j, x_i)}) \tag{2}$$

4 Probability Gaussian Process

Now that we have gained enough knowledge on Kernels, its properties and also its formulation. Here we take our discussion further on another application of kernels in the context of Gaussian Processes and how to deal with smaller training sets to still give fair results.

Now we already know the objective function as

$$w^{\text{regression}} \rightarrow \min \left[\sum_{i\in D} (y_i - w^T x_i)^2\right]$$

The solution to the above problem is:

$$w^* = (\sum_{i\in D} x_i x_i^T)^{-1} \cdot (\sum_{i\in D} x_i y_i)$$

The predictions are made using function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, $f(x_i) = w^T \cdot x_i$

An different approach to this could be to design a distribution on the function we are trying to predict such that every point in the training data must have exactly the same output in the hypothesis as the training label. More precisely, we would like to design a non linear estimator f to model the training data with the additional restriction that $\forall x_i \in D\ f(x_i) = y_i$; for the other points

$x \notin D$, $f(x)$ is a random variable with an associated probability distribution, while having certain guarantees on accuracy on test set and assuming train and test set are from same distribution.

According to the above hypothesis, the function will look something as in Figure 1.

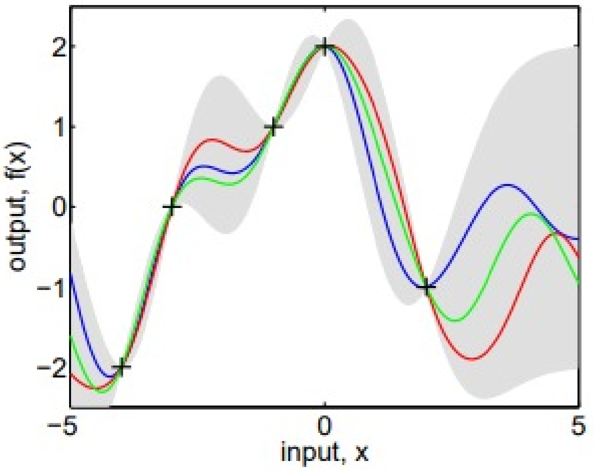

Figure 1: Graphical Representation

Here in this figure you can see that the points marked as + are the points in our dataset, for which the output is exactly one value while it is a distribution (as given by the shaded area) for all the other points (here assumed to be normal distribution).

Whenever we add an extra point to the dataset the mean line changes and passes through that point and the variance at that point becomes 0.

4.1 Gaussian Process

Gaussian processes are a method for non parametric estimation to provide confidence on the seen data and some kind of distribution on unseen data. For any subset of the training data, we must have that the joint prior distribution of this subset is normally distributed for some mean and covariance matrix.

For any subset $\{x_1...x_m\}$ of the training data, the prior distribution follows:

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_m) \end{bmatrix} \quad \sim \quad \mathcal{N}(\vec{\mu}(x_1, \ldots, x_m), \Sigma(x_1, \ldots, x_m))$$

where $\vec{\mu}$ and Σ are deterministic functions.

As discussed earlier, On introducing a new data point into any subset of the training data, we expect the resulting conditional distribution to also follow the normal distribution.

For the data point x^*

$$f(x^*)|(f(x_1), \ldots f(x_m), x^*) \sim \mathcal{N}(\vec{\mu}(x_1, \ldots, x_m, x^*), \Sigma(x_1, \ldots, x_m, x^*))$$

4.2 Conditional Rule for Multi-variate Gaussian

Intuitively, if we start with a Gaussian distribution and update our knowledge given the observed value of one of its components(that is, find conditional probability distribution), then the resulting distribution is still Gaussian! Mathematically,
Let $[x \; y]$ jointly form multi variate Gaussian random variable,

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}\right)$$

Here Σ_{ab} represents covariance matrix between random vectors a and b and $f(\cdot)$ represents the PDF.

$$f(x|y) = \frac{f(x,y)}{f(y)}$$

Now, we will substitute $f(x,y)$ with the expression for multi-variate Gaussian distribution($\mathcal{N}(\mu, \Sigma)$), and $f(y)$ with $\mathcal{N}(\mu_y, \Sigma_{yy})$. Simplifying the equations, we get

$$f(x|y) = \mathcal{N}(\Sigma_{xy}\Sigma_{yy}^{-1}y, \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx})$$

(μ is assumed to be zero for simplicity)

4.3 Updating Parameters on New data

Suppose we have a training set {y,x} and a test set and we try to fit a model through it. Under what conditions will the model go through all the points? Fitting a higher order model would ensure high accuracy on the training set, but test set accuracy must be satisfied.

Given P(y|x) ⇒ As long as we observe y, it's variance should go to zero.

$$P(y|x) = \mathcal{N}(\mu, \Sigma)$$

but if (x,y) $\in \{(x_i, y_i)\}_{i=1}^{N}$ then $\sigma = 0$ There can be error on points we don't observe, but there mustn't be error on already observed points.

To achieve this $\Rightarrow \underset{w}{min} \Sigma (y_i - w^T x_i)^2 \to 0$ where $x_i \to \phi(x_i)$ which dimension may tend to ∞.

$\phi(x_i)$ can't be finite as then it won't work for arbitrary number of training set. w and $\phi(x)$ is not computable but $f(x_i) = \Sigma \alpha_i k(x, x_i) y_i$.

$$\underset{\alpha}{min} \Sigma (y_i - \alpha_i k(x, x_i) y_i$$

$$\underset{\alpha}{min} \sum_j \left[\sum_i y_i - \alpha_i k(x, x_i) y_i) \right]^2$$

If we observe new point y_i, we have to adjust α_i such that to keep variance = 0.