AIML - CS 337

Lecture 12: Kernel Mode

26 / 09/ 2022

Lecturer: Abir De *Scribe: Dyuneesh, Harsha, Sanchit, Yasaswi*

In this Lecture we discuss the smooth transition from Linear to Non-Linear space of functions. Since we are searching for non-linear functions, f(x) is non-linear, that is, $f(x) \neq w^T x$. We discuss treatments where f(x) is non-linear in x but we can construct it using our previous knowledge. We also look at tricks to reduce the order of computation required for the models.

1 Kernel Methods

The Main Concept behind Kernel Methods is that there is a possibility that, data-points that are not linearly-separable are linearly-separable in higher dimensions.

The Midsem Question which stated that the points labeled '+1' are in the range $[-1, 1]$ and the points labeled '-1' are in the range $[-6, -2] \cup [2, 6]$. This is not linearly-separable but on converting to a higher dimension: $\phi(x) = (x, x^2)$, it converts to a form that we are more familiar with.

The Basic Algorithm will be:-

- Given some Data Set $S = (\{\boldsymbol{x_i}\}_{i=0}^n, \{y_i\}_{i=0}^n)$, where $\boldsymbol{x_i} \in \mathbb{R}^d$
- Consider a function $\boldsymbol{\phi}$ such that $\boldsymbol{\phi(x)} \in \mathbb{R}^{d'}$ where $d' > d$ (and can even be infinity)
- Create a new Data Set $\hat{S} = (\{\boldsymbol{\phi(x_i)}\}_{i=0}^n, \{y_i\}_{i=0}^n)$
- Train a linear Predictor h over $\hat{S}$
- And then the prediction of any point $\boldsymbol{x_{test}}$ in the test dataset is given by $h(\boldsymbol{\phi(x_{test})})$

Thus the prediction is given by $\boldsymbol{w^T \phi(x)}$ where both $\boldsymbol{w}, \boldsymbol{\phi(x)} \in \mathbb{R}^{d'}$

As d' can reach infinity, calculating and storing $\boldsymbol{w}$ and $\boldsymbol{\phi(x)}$ becomes practically impossible, but the dot product $\boldsymbol{w^T \phi(x)}$ is a scalar.

Here we are enriching the expressive power of halfspaces by first mapping the data into a high dimensional feature space, and then learning a linear predictor in that space. While this approach greatly extends the expressiveness of halfspace predictors, it raises both sample complexity and computational complexity challenges. We tackle this using the method of *kernels*.

A popular choice for the mapping ϕ is a polynomial mapping that is $x \to (1, x, x^2, \ldots)$

The Setup:
Given a mapping ϕ we are left to solve the optimization problem:

$$\min_w f(\{y_i\}_i^n, \{\boldsymbol{w^T \phi(x_i)}\}_i^n) + \lambda R(\boldsymbol{w})$$

Where f is a loss function and R is a monotonic Regularization function.

2 Solution to Setup

- In a finite case if we sample d dimension vector N times such that $d << N$. .It is with high probability that there will be d vectors that are linearly independent of each other. But as the value of d itself tends to infinity it becomes less and less likely
- In the given case the vectors $\{\boldsymbol{\phi(x_i)}\}_i^n$ are of infinite dimension. So if the optimum solution is $\boldsymbol{w^*}$ then there exists $\{\alpha_i\}_i^n$ such that $\boldsymbol{w^*}$ can be represented as

$$\boldsymbol{w^*} = \sum_{i=0}^{n} \alpha_i * \boldsymbol{\phi(x_i)} \ + \boldsymbol{v}$$

  where $\boldsymbol{v^T \phi(x_i)} = 0$ for all i, that is, **v** is orthogonal to the span of the vectors mapped by the function ϕ.
- Let us define $\boldsymbol{w} = \boldsymbol{w^*} - \boldsymbol{v}$

$$\begin{aligned}
&||\boldsymbol{w}||^2 = ||\boldsymbol{w^*} - \boldsymbol{v}||^2 \\
&||\boldsymbol{w}||^2 = ||\boldsymbol{w^*}||^2 - 2\boldsymbol{w^*}.\boldsymbol{v} + ||\boldsymbol{v}||^2 \qquad \text{As the value } \boldsymbol{w^*}.\boldsymbol{v} = \boldsymbol{v}.\boldsymbol{v} = ||\boldsymbol{v}||^2 \\
&||\boldsymbol{w}||^2 = ||\boldsymbol{w^*}||^2 - 2||\boldsymbol{v}||^2 + ||\boldsymbol{v}||^2 \\
&||\boldsymbol{w}||^2 = ||\boldsymbol{w^*}||^2 - ||\boldsymbol{v}||^2
\end{aligned}$$

- So as norm is a positive function we have $||\boldsymbol{w}|| \leq ||\boldsymbol{w^*}||$
- And since R is non-decreasing, we obtain $R(\boldsymbol{w}) \leq R(\boldsymbol{w^*})$
- Also as $\boldsymbol{w^T \phi(x)} = (\boldsymbol{w^*} - \boldsymbol{v})^T \boldsymbol{\phi(x)} = \boldsymbol{w^{*T} \phi(x)}$
- And so,

$$f(\{y_i\}_i^n, \{\boldsymbol{w^T \phi(x_i)}\}_i^n) = f(\{y_i\}_i^n, \{\boldsymbol{w^{*T} \phi(x_i)}\}_i^n)$$

- We have shown that the loss function is same for both $\boldsymbol{w^*}$ and $\boldsymbol{w}$ and the regularization function is less for $\boldsymbol{w}$ than $\boldsymbol{w^*}$,
- Hence the objective function for $\boldsymbol{w}$ is less than that for $\boldsymbol{w^*}$, but as $\boldsymbol{w^*}$ is the optimum solution, we must have that $\boldsymbol{w}$ is also an optimum solution .
- Hence we have proved that the value

$$\boldsymbol{w} = \sum_{i=0}^{n} \alpha_i * \boldsymbol{\phi(x_i)}$$

  is an optimum solution for the objective function.[1]

[1] Also remember that this was the result that we got by applying the Lagrange multiplier on the objective functions

Theorem 2.1 (**Representer Theorem**). *Given a mapping from $\mathbb{R}^d$ to $\mathbb{R}^{d'}$, there exists a vector $\boldsymbol{\alpha} \in \mathbb{R}^{d'}$ such that $\boldsymbol{w} = \sum_{i=1}^{d'} \alpha_i \boldsymbol{\phi}(\boldsymbol{x_i})$ is an optimal solution.*

$$\min_{w} f(\{y_i\}_i^n, \{\boldsymbol{w^T \phi}(\boldsymbol{x_i})\}_i^n) + \lambda R(\boldsymbol{w})$$

3 Reducing Computation

- Substituting the fact that $\boldsymbol{w} = \sum_{i=1}^{d'} \alpha_i \boldsymbol{\phi}(\boldsymbol{x_i})$ is an optimum solution into the value $\boldsymbol{w^T \phi}(\boldsymbol{x})$

$$\begin{aligned}
\boldsymbol{w^T \phi}(\boldsymbol{x}) &= \left(\sum_{i=1}^{d'} \alpha_i \boldsymbol{\phi}(\boldsymbol{x_i})^{\boldsymbol{T}} \right) \boldsymbol{\phi}(\boldsymbol{x}) \\
&= \sum_{i=1}^{d'} \alpha_i \boldsymbol{\phi}(\boldsymbol{x_i})^{\boldsymbol{T}} \boldsymbol{\phi}(\boldsymbol{x}) \qquad \text{consider a function } K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \boldsymbol{\phi}(\boldsymbol{x_i})^{\boldsymbol{T}} \boldsymbol{\phi}(\boldsymbol{x_j}) \\
&= \sum_{i=1}^{d'} \alpha_i K(\boldsymbol{x_i}, \boldsymbol{x})
\end{aligned}$$

- Similarly we have that,

$$\begin{aligned}
||\boldsymbol{w}||^2 &= \boldsymbol{w^T w} \\
&= \left(\sum_{i=1}^{d'} \alpha_i \boldsymbol{\phi}(\boldsymbol{x_i})^{\boldsymbol{T}} \right) \left(\sum_{i=1}^{d'} \alpha_i \boldsymbol{\phi}(\boldsymbol{x_i}) \right) \\
&= \sum_{i=1}^{d'} \sum_{j=1}^{d'} \alpha_i \alpha_j \boldsymbol{\phi}(\boldsymbol{x_i})^{\boldsymbol{T}} \boldsymbol{\phi}(\boldsymbol{x_j}) \\
&= \sum_{i=1}^{d'} \sum_{j=1}^{d'} \alpha_i \alpha_j K(\boldsymbol{x_i}, \boldsymbol{x_j})
\end{aligned}$$

- K is called a *Kernel* Function and it denotes *Similarity* between the data points $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$.
- Some Popular Kernels are:
  - **SVM** $K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \boldsymbol{x_i^T x_j}$
  - **Gaussian** $K(\boldsymbol{x_i}, \boldsymbol{x_j}) = e^{\frac{-||\boldsymbol{x_i} - \boldsymbol{x_j}||^2}{\sigma^2}}$
  - **k degree polynomial** $K(\boldsymbol{x_i}, \boldsymbol{x_j}) = (1 + \langle \boldsymbol{x_i}, \boldsymbol{x_j} \rangle)^k$

  where $\langle ., . \rangle$ denotes the inner product.

4 Kernel Function

To show that a function is indeed a Kernel Funtion, We need to show that there exists a mapping $\boldsymbol{\phi}$ to a higher dimension space such that

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

To show that K is a Kernel function, we first need it to be Positive Semi-Definite. That is,

for all real-valued c_i, c_j and for all $\boldsymbol{x_i}, \boldsymbol{x_j}$ in the domain of $\boldsymbol{\phi}$,

$$\sum_{i=1}^{D} \sum_{j=1}^{D} c_i c_j \boldsymbol{\phi}(\boldsymbol{x_i})^{\boldsymbol{T}} \boldsymbol{\phi}(\boldsymbol{x_j}) \geq 0$$

Since we have shown that K is Positive Semi-Definite, K implements an inner product in some Hilbert space and hence is a valid Kernel function.

Now going back to the dual formulation of Hard-SVM:

$$\max_{\boldsymbol{\alpha} \in R^D : \boldsymbol{\alpha} \geq \mathbf{0}} \left(\sum_{i=1}^{D} \alpha_i - \frac{1}{2} \sum_{i=1}^{D} \sum_{j=1}^{D} \alpha_i \alpha_j y_i y_j \boldsymbol{x_i^T x_j} \right)$$

$$= \max_{\boldsymbol{\alpha} \in R^D : \boldsymbol{\alpha} \geq \mathbf{0}} \left(\sum_{i=1}^{D} \alpha_i - \frac{1}{2} \sum_{i=1}^{D} \sum_{j=1}^{D} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x_i}, \boldsymbol{x_j}) \right)$$

$$= \max_{\boldsymbol{\alpha} \in R^D : \boldsymbol{\alpha} \geq \mathbf{0}} \left(\sum_{i=1}^{D} \alpha_i - \frac{1}{2} (\boldsymbol{\alpha} * \boldsymbol{Y})^{\boldsymbol{T}} \boldsymbol{G} (\boldsymbol{Y} * \boldsymbol{\alpha}) \right)$$

$*$ denotes element-wise multiplication

$$\boldsymbol{G} = \{K(\boldsymbol{x_i}, \boldsymbol{x_j})\}_{i,j=1}^{D \times D} \qquad \text{Known as the Gram matrix}$$

Hence, we are optimising for $\boldsymbol{\alpha}$ instead of $\boldsymbol{w}$.
The advantage of working with kernels rather than directly optimizing w in the feature space is that in some situations the dimension of the feature space is extremely large while implementing the kernel function is very simple.

References

[1] S. B.-D. Shai Shalev-Shwartz. *Understanding Machine Learning, Chapter 16*. Cambridge University Press, 2014.