STAT4609 Example Class 3

Technical Details for Kernel Regression

LIU CHEN

Ordinary Linear Regression

a recap on linear regression. Given a feature vector$x = [x_1, x_2, ... x_n]$, consisting of n features and the corresponding labels y, linear regression tries to find the optimal coefficients $\beta_i, i \in \{0,...,n\}$of the line equation $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n$ usually by gradient descent and measured on the RMSE metric. The equation is then used to predict the target for new unseen inputs.

Linear regression is a simple algorithm that cannot model very complex relationships between the input features. Mathematically, this is because well, it is linear with the degree of the equation being 1, which means that linear regression will always model a straight line.

Ordinary Linear Regression

Maximum Likelihood Estimator

In assignment 1, we have derived the likelihood estimator for regression weights,

The likelihood and loglikelihood function are shown below,

$$P(y\,|\,\beta,\sigma^2) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^N \exp\left\{\frac{-1}{2\sigma^2}(Y - X\beta)^T(Y - X\beta)\right\}$$

$$\log P(y\,|\,\beta,\sigma^2) \propto (Y - X\beta)^T(Y - X\beta).$$

Then, we let the derivative be zero,

$$\frac{\partial \log P(y\,|\,\beta,\sigma^2)}{\partial\beta} = -\,2X^TY + 2X^TX\beta.$$

$$\hat{\beta} = (X^TX)^{-1}X^TY$$

Ordinary Linear Regression

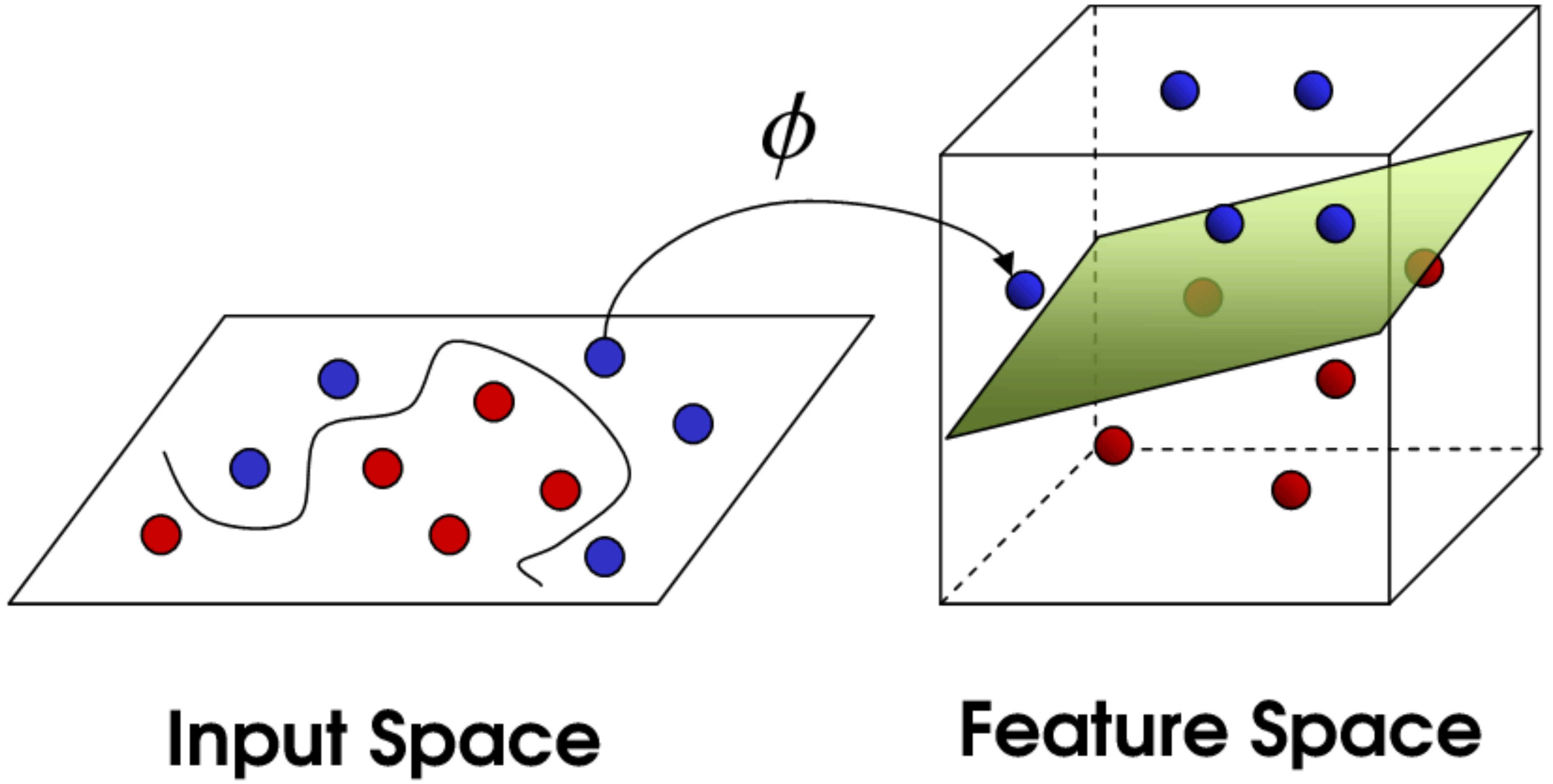
Limitations

But what if our data doesn't have the form of a straight line? In this case, we can use polynomial regression in which the degree of the aforementioned equation is $n, n \gneq 1$ However, with polynomial regression another problem arises: as a data analyst, you cannot know what the degree of the equation should be (which is trial and error) and moreover, the model built using polyreg is difficult to visualize above degree 3.

So, what do we do? We use Kernel Regression

What is Kernel Regression?

We now know how to compute a linear regression. But often data is not linear … we could come up with non linear regression methods. But we just picked up linear regression so let’s see what we can do. If we add a few dimensions to our input data we might be able to a linear regression in the higher dimension(feature space). Or a linear separation as depicted below:



What is Kernel Regression?

example

What does that transformation look like? Just to give you an idea a simple polynomial mapping could look like this:

$$\mathbf{X} = (X_1, X_2)$$

$$\phi(\mathbf{X}) = (X_1^2, X_2^2, \sqrt{2}X_2, \sqrt{2}X_1X_2, 1)$$

Well that certainly look like a lot of work for us and the computer. We went from 2 dimension to 6.

Not only do we have to compute that embedding in the first place,

anything we do from here on has to work on the 6 dimensions rather than 2.

What is Kernel Regression?

Luckily some smart people noticed that there are certain special transformation We can equivalently represent basis function regression with a “kernel function”.

Kernel is a $N \times N$ matrix,

$$K(x, x') = \Phi(x)\Phi(x')^T$$

What is a valid kernel?

- A valid kernel, $K(x, x')$, must be symmetric and positive semidefinite for any arbitrary input.

We can define a kernel without needing to define a basis function.

Evaluating kernel is generally faster than evaluating basis function.

Kernel Regression

Formulation

Now we just take our by now familiar linear regression and replace all dot products with kernel functions,

$$\hat{\beta} = \left(\Phi^T \Phi\right)^{-1} \Phi^T y = \Phi^T \left(\Phi \Phi^T\right)^{-1} y.$$

Let $k\left(x, x'\right) = \Phi(x)\Phi\left(x'\right)^T$, then we have

$$\hat{\beta} = \Phi^T \left(K\right)^{-1} y$$

$$K_{ij} = k(X_i, X_j)$$

To predict a new value

$$y* = \Phi\left(X*\right)\hat{\beta} = \Phi\left(X*\right)\Phi^T (K)^{-1} y = \mathbf{k}(X*)(K)^{-1} y$$

$$\boxed{\mathbf{k}(X^*) = \begin{bmatrix} k(X^*, X_0) \\ \dots \\ k(X^*, X_n) \end{bmatrix}}$$

Coding Exercise!

Have Fun!