November 2, 2017 Lecture 16, Oct. 30, 2017

Kernel Methods

Lecturer: Justin Domke *Scribe: Lakshmi Vikraman*

1 Summary

We will continue the discussion on Kernel Methods.

2 Polynomial Kernels and Copies

We are going to discuss the reasoning behind the discrepancies in output values observed in HW03 Question 1d. The values obtained after performing Kernel Ridge Regression using the polynomial kernel does not match the values obtained after performing basis expansion followed by ridge regression(BERR+Ridge).

The polynomial kernel defined is given below

$$k(x, x') = (1 + x^T x')^d$$

For the case where p=1 and d=2,

$$\begin{aligned} k(x, x') &= (1 + xx')^2 \\ &= 1 + 2xx' + x^2 x'^2 \end{aligned}$$

The basis expansion h for d=2 is

$$h(x) = [1, x, x^2]$$

$$h(x)^T h(x) = 1 + xx' + x^2 x'^2$$

As you can see $k(x, x') \neq h(x)^T h(x)$.

However, $k(x, x')$ is equal to the modified $h(x)$ given below

$$h(x) = [1, x, x, x^2]$$

$$h(x)^T h(x) = 1 + 2xx' + x^2 x'^2$$

More generally we can define k as

$$\begin{aligned} k(x, x') &= (x^T x')^d \\ &= (\sum_{i=1}^{p} x_i x'_i)^d \end{aligned}$$

Let us apply the following multinomial theorem

$$(\sum_{i=1}^{p} z_i)^d = \sum_{k_1+k_2+...+k_p=d} \frac{d!}{k_1!k_2!......k_p!} \prod_{i=1}^{p} z_i^{k_i}$$

Thus $k(x,x') = (x^T x')^d$ corresponds to $h(x)$ where $h_m(x) = x_1^{k_1} x_2^{k_2} ... x_p^{k_p}$ and has $\frac{d!}{k_1!k_2!......k_p!}$ copies. Here $k_1, k_2, ...k_p$ has to add up to d. This can take many different combinations. For instance k_1 can be d and the rest 0 or k_1, k_2 is each set to $\frac{d}{2}$ and rest to 0 etc.

The bottom line is that the basis expansion for a polynomial kernel has lots of copies of say $x_1 x_7 x_9 x_{21}$ where we have different features combined together but few copies of say x_{15}^4 where we have one feature raised to a high power.

How does this impact linear models? If we have a dataset a with one feature $\beta_a = 3$ and dataset b with two copies of the feature $\beta_b = (1.5, 1.5)$, if we don't regularize, then their is no difference between the two. But if we use ridge regularization, then in the first case, the penalty would be 3^2 while the penalty for the second case is $2(1.5)^2$ which is less. So when we have multiple copies of the features, then those features get less regularized as compared to the features with less number of copies. For a higher order polynomial kernel, most of the predictive power comes from the interaction between different features and not much from the single features.

3 Representer Theorem

Let us consider the risk function $R(\beta)$

$$R(\beta) = \sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda \|\beta\|_2^2$$

If β minimizes R, then their exists a vector $\alpha \in R^N$ such that

$$\beta = \sum_{j=1}^{N} \alpha_j h(x_j)$$

For example, if we have a dataset with two datapoints and $h(x)$ where $h \in R^M$,$M = 10^6$ and $\beta \in R^M$, then this is a huge space to search through to find the optimal β. But according to the theorem,we can represent β in terms of $\alpha \in R^2$ and the search space becomes much smaller. We need to search over only 2 parameters as opposed to the earlier 10^6 parameters.

$$\beta = \alpha_1 h(x_1) + \alpha_2 h(x_2)$$

3.1 Proof of Representer Theorem

$$\begin{aligned}\nabla R(\beta) &= \sum_{i=1}^{N} \frac{df(x_i)}{d\beta}\frac{dL(y_i, f(x_i))}{df} + 2\lambda\beta \\ &= \sum_{i=1}^{N} h(x_i)\frac{dL(y_i, f(x_i))}{df} + 2\lambda\beta\end{aligned}$$

Here $\frac{df(x_i)}{d\beta} \in R^p$,$\frac{dL(y_i,f(x_i))}{df} \in R$
Also , $f(x) = \beta^T h(x)$, hence $\nabla_\beta f(x) = h(x)$

Solving for β by setting the gradient to zero, we get

$$\begin{aligned}\beta &= \frac{-1}{2\lambda}\sum_{i=1}^{N} h(x_i)\frac{dL(y_i, f(x_i))}{df} \\ &= \sum_{i=1}^{N} h(x_i)\alpha_i \\ \text{where } \alpha_i &= \frac{-1}{2\lambda}\frac{dL(y_i, f(x_i))}{df}\end{aligned}$$

3.2 Meaning of Representer Theorem

If $h(x) \in R^M$, then the solution β can't be just anything. It must be a combination of $h(x_1), h(x_2), ...h(x_N)$.
We know that $\beta = \sum_{j=1}^{N} \alpha_j h(x_j)$. To classify a new point x,

$$\begin{aligned}f(x) &= \beta^T h(x) \\ &= \sum_{j=1}^{N} \alpha_j h(x_j)^T h(x) \\ &= \sum_{j=1}^{N} \alpha_j k(x, x_j)\end{aligned}$$

Substituting the representer back into the risk function , we get

$$\min_\beta \sum_{i=1}^{N} L(y_i, \beta^T h(x_i)) + \lambda\|\beta\|_2^2$$

$$\exists\alpha \text{ such that } \beta = \sum_{j=1}^{N} \alpha_j h(x_j), \beta \in R^M, \alpha \in R^N$$

$$\beta^T h(x_i) = \sum_{j=1}^{N} \alpha_j h(x_j)^T h(x_i)$$

We can rewrite the equation in terms of α

$$\min_{\alpha} \sum_{i=1}^{N} L(y_i, \sum_{j=1}^{N} \alpha_j h(x_j)^T h(x_i)) + \lambda \| \sum_{j=1}^{N} \alpha_j h(x_j) \|_2^2$$

$$\min_{\alpha} \sum_{i=1}^{N} L(y_i, \sum_{j=1}^{N} \alpha_j h(x_j)^T h(x_i)) + \lambda \sum_{i=1}^{N} \sum_{j=1}^{N} h(x_j)^T h(x_i) \alpha_i \alpha_j$$

$$\min_{\alpha} \sum_{i=1}^{N} L(y_i, \sum_{j=1}^{N} \alpha_j k(x_i, x_j)) + \lambda \sum_{i=1}^{N} \sum_{j=1}^{N} k(x_i, x_j) \alpha_i \alpha_j$$

$$\min_{\alpha} \sum_{i=1}^{N} L(y_i, (\mathbf{K}, \alpha)_i) + \lambda \alpha^T \mathbf{K} \alpha$$

Here $\mathbf{K}$ is the kernel matrix and k is the kernel function.
$\mathbf{K}_{ij} = k(x_i, x_j)$
$(\mathbf{K}, \alpha)_i = \sum_j \mathbf{K}_{ij} \alpha_j$
The books uses the convention $L(\mathbf{Y}, \mathbf{Y}') = \sum_{i=1}^{N} L(y_i, y_i')$ and the equation can be written as

$$\min_{\alpha} \sum_{i=1}^{N} L(\mathbf{Y}, \mathbf{K}\alpha) + \lambda \alpha^T \mathbf{K} \alpha$$

4 String Kernels

String Kernels are used where we want to predict the numerical ratings from textual ratings. Consider the following dataset where x_i is the input and y_i is the corresponding rating.

i	x_i	y_i
1	average	3.5
2	avaragee	4.3
3	got me fired	1.2

Here how do we deal with x. The answer is to perform preprocessing a.k.a basis expansion. We create a big vector with n-gram counts. A few examples of how the ngram counts are calculated is shown below.

ngrams	average	avaragee
aaa	0	0
aab	0	0
age	1	1
ara	0	1
ave	1	0
zzz	0	0

Here $h_m(x) = \#u(x)$ where $\#u(x)$ indicates number of occurences of u in x. For eg, $\#\text{ave(average)} = 1$. This is good for text processing where the length of $h(x)$ is 26^n. This has other applications such as computational biology where n is very high and therefore a naive basis expansion would be very expensive. This motivates finding a kernel, which we will turn to next time.