November 6, 2017 Lecture 15, Oct. 25, 2017

Kernel Ridge Regression

Lecturer: Justin Domke *Scribe: Annamalai Natarajan*

1 Summary

In the last class we introduced kernel methods. In this class we focus on kernel methods for regression. We start with ridge regression, a popular approach for regression, and then look at other variants of ridge regression and finally look at kernel ridge regression. Variants of ridge regression are necessarily brief in order to focus on kernel methods.

Consider a dataset D of the form $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ of N examples. In this problem setup $x \in \mathbf{R}^p$ and $y \in \mathbf{R}$, where p is the number of features.

2 Regular ridge regression

In regular ridge regression, we compute coefficients β as,

$$\beta = (C + \lambda I)^{-1}\mathbf{X}^T\mathbf{y} \tag{2.1}$$

$$C = \sum_{i=1}^{N} x_i x_i^T \tag{2.2}$$

Both $C, \beta \in \mathbf{R}^p$ and λ is the regularization parameter.

- Time complexity to learn β is $p^2N + p^3$
- To predict a new examples x, $f(x) = \beta^T x$
- Time complexity to predict is p

3 Dual Ridge regression

In dual ridge regression, we compute coefficients α as,

$$\alpha = (K + \lambda I)^{-1}\mathbf{y} \tag{3.1}$$

$$K \in \mathbf{R}^{N \times N} matrix \tag{3.2}$$

$$k_{ij} = x_i^T x_j \tag{3.3}$$

where, $\alpha \in \mathbf{R}^N$, K is the gram matrix, k_{ij} is the $(i, j)^{th}$ entry in K matrix and λ is the regularization parameter.

- Time complexity to learn α is $N^2p + N^3$
- To predict a new examples x, $f(x) = \sum_{i=1}^{N} \alpha_i x^T x_i$
- Time complexity to predict is Np
- Not so useful in practice since typically $N >> p$

4 Basis expanded ridge regression

In basis expanded ridge regression, we compute coefficients β as,

$$\beta = (C + \lambda I)^{-1}\mathbf{H}^T\mathbf{y} \tag{4.1}$$

$$C = \sum_{i=1}^{N} h(x_i)h(x_i)^T \tag{4.2}$$

$$\mathbf{H} = \begin{bmatrix} h(x_1)^T \\ h(x_2)^T \\ \vdots \\ h(x_n)^T \end{bmatrix} \tag{4.3}$$

where $\mathbf{H} \in \mathbf{R}^{N \times M}$ matrix and M is the length of basis expansion h

- Time complexity to learn β is $M^2N + M^3$
- To predict a new examples x, $f(x) = \beta^T h(x)$
- Time complexity to predict is M
- These time complexities assume you can compute $h(x)$ in M time

5 Kernel ridge regression

In kernel ridge regression, we compute coefficients α as,

$$\alpha = (K + \lambda I)^{-1}\mathbf{y} \tag{5.1}$$

$$K \in \mathbf{R}^{N \times N} matrix \tag{5.2}$$

$$k_{ij} = x_i^T x_j \tag{5.3}$$

$$= h(x_i)^T h(x_j) \tag{5.4}$$

where $\alpha \in \mathbf{R}^N$, K is the gram matrix, k_{ij} is the $(i,j)^{th}$ entry in K matrix and λ is the regularization parameter.

- Time complexity to learn α is $N^2p + N^3$
- To predict a new examples x, $f(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i)$
- Time complexity to predict is Np

- This approach assume that K is computed in p time
- It has similar bias and variance as basis expanded ridge regression

Here is a comparison of basis expansion + ridge with kernel ridge. Despite both having identical bias and variance and produces similar results there are some crucial differences.
Basis expansion + Ridge

- Deals well with large N but scales poorly with M
- Can use any $h(x)$
- Can use different regularization parameter, λ

Kernel ridge

- Scales poorly with N but scales well with N
- Need fast computation of kernel $K(x, x')$ when compared to computing $h(x)^T h(x`)$
- Fixed to ridge regression

6 Basis expansions

The key to success of kernel methods is the basis expansions, h, which captures complex interactions between input features. Next, we are going to examine few popular basis expansions.

6.1 Naive polynomial basis

We compute the naive polynomial basis as,

$$h_m(x_j) = [x_{j_1}, x_{j_2}, ..., x_{j_d}] \tag{6.1}$$

where $M = p^d$ and d is the dimension of the polynomial kernel. This is compactly represented as $K(x, x`) = (x^T x`)^d$ which we assume is computed in M time. The alternate approach is to first expand basis vectors and then compute $K = h(x)^T h(x`)$ is more time consuming.

6.2 Radial Basis Function

One of the most popular basis expansion is radial basis expansion or RBF kernel which is of infinite dimension. The nice thing is we don't need to explicitly compute the infinite dimensional vector but the kernel computes the similarity between two input feature vectors in this infinite dimensions. The RBF kernel is represented as,

$$K(x, x') = \exp(-\gamma \|x - x'\|_2^2) \tag{6.2}$$

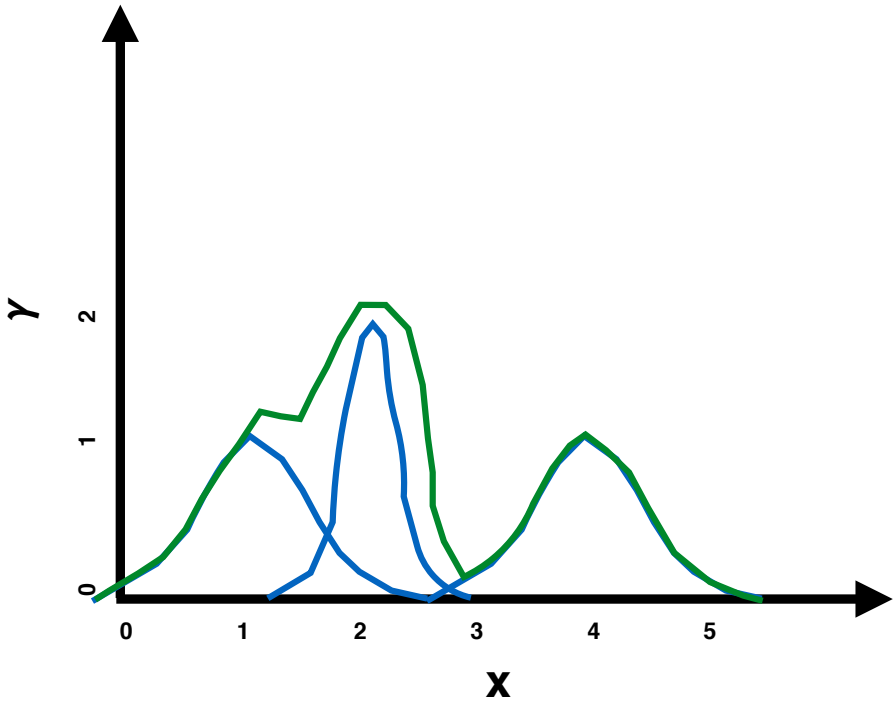

Figure 6.1: Example of RBF kernel in one dimension. Individual kernels for each data sample in blue and green envelope captures the overall influence of this small dataset of size $N = 3$

where γ captures the influence of examples on x‘ and it should be $\gamma > 0$. Consider this simple example of $N = 3$ data samples of $x \in \mathbf{R}$. The three examples are $x_1 = 1, x_2 = 2, x_3 = 4$ with $\gamma_1 = 1, \gamma_2 = 2, \gamma_3 = 1$. Graphically we represent x and γ as in Figure 6.1. To predict a new example x as $f(x) = \sum_{i=1}^{N} \gamma_i K(x, x_i)$. For the one dimension example a sample prediction would look like,

$$f(x) = \gamma_1 \times K(x, x_1) + \gamma_2 \times K(x, x_2) + \gamma_3 \times K(x, x_3) \tag{6.3}$$
$$= 1 \times K(x, 1) + 2 \times K(x, 2) + 1 \times K(x, 4) \tag{6.4}$$

This is summing up each bell shaped curve and weighting it by its respective γ. Flat curves have farther influence than spiked curves.

Lastly only K's that satisfy Mercer's theorem can be used as kernel. The kernels replace the dot product of two vectors thereby making computation efficient and capturing complex interactions. In the next class we will discuss representer theorem that relates empirical risk minimization with minimized coefficients β.