# KERNEL RIDGE REGRESSION

#### Chao Lan

Linear regression model assumes y is a linear function of x. In practice, their relation may be nonlinear.

[Discussion] What are the naive ways to address this problem?

Kernel ridge regression captures the nonlinear relation, by first mapping x into a higher dimensional feature space and then learning a linear model there. Its assumption is the relation is more likely to be linear in the new space.

[Discussion] Geometric interpretation of kernel regression.

## 1. Model Construction and Learning Objective

Let  $\phi$  be a function mapping x from its raw feature space to a higher dimensional feature space  $\mathcal{F}$ . In this space, we have a set of mapped training instances  $(\phi(x_1), y_1), \ldots, (\phi(x_n), y_n)$ .

KRR learns a linear model  $\beta$  in  $\mathcal{F}$  using ridge regression, i.e., it finds a  $\beta$  that minimizes

$$J(\beta) = \sum_{i=1}^{n} (\phi(x_i)^T \beta - y_i)^2 + \lambda \beta^T \beta.$$
 (1)

Note: KRR also regularizes  $\beta_0$  for numerical convenience. This will be clear in later discussions.

[Discussion] How is kernel regression different from feature engineering?

### 2. Model Learning

KRR aims to learn  $\beta$  from (1) without calculating the explicit representation of  $\phi(x)$ . It achieves so by jointly applying two tricks: (i) Representer Theorem and (ii) kernel function. We will introduce these ideas later.

For now, let us derive the optimal  $\beta$  of min  $J(\beta)$  in a standard fashion. We will have

$$\beta = \sum_{i=1}^{n} \alpha_i \, \phi(x_i),\tag{2}$$

where  $\alpha_i = -\frac{1}{\lambda}(\phi(x_i)^T\beta - y_i)$ . Note that  $\alpha_i$  is unknown since  $\beta$  is unknown.

[Discussion] Derive (2).

Plugging (2) back to (1), we turn the objective into a function of  $\alpha$ , i.e.,

$$J(\alpha) = \sum_{i=1}^{n} \left( \phi(x_i)^T \left( \sum_{j=1}^{n} \alpha_j \phi(x_j) \right) - y_i \right)^2 + \lambda \left( \sum_{i=1}^{n} \alpha_i \phi(x_i) \right)^T \left( \sum_{j=1}^{n} \alpha_j \phi(x_j) \right)$$

$$= \sum_{i=1}^{n} \left( \sum_{j=1}^{n} \alpha_j \phi(x_j)^T \phi(x_i) - y_i \right)^2 + \lambda \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \phi(x_i)^T \phi(x_j)$$

$$= \sum_{i=1}^{n} \left( \sum_{j=1}^{n} \alpha_j \kappa(x_j, x_i) - y_i \right)^2 + \lambda \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \kappa(x_i, x_j)$$
(3)

where we define a kernel function  $\kappa(\cdot, \cdot)$  as

$$\kappa(x_i, x_j) = \phi(x_i)^T \phi(x_j). \tag{4}$$

#### 3. Kernel Function

There are many commonly used kernel functions. For example, the Gaussian kernel is

$$\kappa(a,b) = \exp\left(-\frac{||a-b||^2}{2\sigma^2}\right),\tag{5}$$

where  $\sigma$  is a hyper-parameter. The polynomial kernel is

$$\kappa(a,b) = (a^T b + m)^d, \tag{6}$$

where m and d are hyper-parameters.

Although we do not calculate  $\phi$  explicitly, each kernel is essentially associated with an explicit  $\phi$ , e.g., Gaussian kernel is associated with a mapping to an infinitely high dimensional feature space. Assuming  $x \in \mathbb{R}$ , its mapping is

$$\phi(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right) \left[1, \frac{x}{\sigma\sqrt{1!}}, \frac{x^2}{\sigma^2\sqrt{2!}}, \frac{x^3}{\sigma^3\sqrt{3!}}, \dots\right]^T. \tag{7}$$

[Discussion] Why does KRR want to avoid calculating the explicit  $\phi$ ?

As another example, suppose  $a=[a_1,a_2]^T$  and  $b=[b_1,b_2]^T$ . The following polynomial kernel

$$\kappa(a,b) = (a^T b)^2 \tag{8}$$

is associated with mapping  $\phi(a) = [a_1^2, \sqrt{2}a_1a_2, a_2^2]^T$ .

[Discussion] Prove (10).

[Discussion] What is the associated mapping for  $\kappa(a,b)=(a^Tb+m)^2$  ?

In addition to these basic kernel functions, we can construct new kernel functions from the basic ones.

Example 1: If  $\kappa_1(a,b)$  is a kernel, then  $\kappa(a,b) := \lambda \cdot \kappa_1(a,b)$  is also a kernel.

Example 2: If  $\kappa_1(a,b)$  and  $\kappa_2(a,b)$  are two kernels, then  $\kappa(a,b) := \kappa_1(a,b) + \kappa_2(a,b)$  is also a kernel.

[Discussion] Prove the two examples. (proof by construction)

#### 2. Model Learning (Cont.)

After choosing a kernel function, we can write the new objective in a matrix form and optimize it.

Let  $K \in \mathbb{R}^{n \times n}$  be the Gram matrix, where

$$K_{ij} = \kappa(x_i, x_j), \tag{9}$$

and  $\alpha = [\alpha_1, \dots, \alpha_n]^T$  be the vector of unknown parameters. We have

$$J(\alpha) = (K\alpha - Y)^T (K\alpha - Y) + \lambda \alpha^T K\alpha. \tag{10}$$

[Discussion] Derive (10).

Clearly  $J(\alpha)$  is quadratic. Setting  $J'(\alpha) = 0$  and solving for  $\alpha$  gives

$$\alpha = (K + \lambda I)^{-1} Y. \tag{11}$$

[Discussion] Derive (15).

## 4. Model Prediction

Once  $\alpha$  is learned, we can predict the label of a testing instance  $\phi(z)$  by

$$\phi(z)^{T}\beta = \phi(z)^{T} \cdot \sum_{i=1}^{n} \alpha_{i}\phi(x_{i}) = \sum_{i=1}^{n} \alpha_{i}\phi(z)^{T}\phi(x_{i}) = \sum_{i=1}^{n} \alpha_{i}\kappa(z, x_{i}).$$
(12)

Let  $(z_1, t_1) \dots (z_m, t_m)$  be a set of testing instances with feature z and label t. We can evaluate testing error as

$$mse = \sum_{i=1}^{m} (\phi(z_i)^T \beta - t_i)^2 = ||K_{zx} \cdot \alpha - T||^2,$$
(13)

where  $T = [t_1, \dots, t_m]^T$  is the label vector, and  $K_{zx} \in \mathbb{R}^{m \times n}$  is a kernel matrix with element

$$K_{zx}(i,j) = \kappa(z_i, x_j). \tag{14}$$

[Discussion] Derive (13).

## 5. Representer Theorem

Kernel ridge regression is a special application of <u>kernel methods</u>. These methods have a common idea of first mapping data into a higher dimensional space and then learning a (linear) model there, either for regression or classification.

In KRR, we show the optimal model is a linear combination of training instances. This is in fact a general result.

(Representer Theorem) Let X be a sample space equipped with a kernel  $\kappa$ , and  $\mathcal{F}$  be its associated Reproducing Kernel Hilbert Space (RKHS). Let  $x_1, \ldots, x_n \in X$  be a set of instances. Consider the following optimization problem

$$\min_{f \in \mathcal{F}} L(f(x_1), \dots, f(x_n)) + \Omega(||f||), \tag{15}$$

where L depends on  $x_i$  only through f and  $\Omega$  is non-decreasing. If (15) has a minimizer, then one has the form

$$f_*(z) = \sum_{i=1}^n \alpha_i \cdot \kappa(z, x_i). \tag{16}$$

Further, if  $\Omega$  is strictly increasing, then every minimizer has the form (16).

#### 6. More Discussions

- (i) What are the pros and cons of kernel method?
- (ii) Can we have a kernel version of least square?
- (iii) How to improve the computational efficiency of kernel ridge regression?