Lecture 3 – 5: Linear Regression

Shukai Gong

1 Polynomial Regression

Polynomial Regression

A polynomial with $N-1$ degrees is:

$$p(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{N-1} x^{N-1} = \sum_{j=1}^{N} c_{j-1} x^{j-1} = \boldsymbol{x}^\top \boldsymbol{c}$$

Given a set of data $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N \subset \mathcal{X}$, then the polynomial regression model is

$$p(\boldsymbol{x}) = \boldsymbol{X}\boldsymbol{c}$$

where $\boldsymbol{X}$ can be expressed as a Vandermonde matrix $\boldsymbol{X} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{N-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^{N-1} \end{bmatrix}$. Our goal is to find the optimal coefficients $\boldsymbol{c}$.

Naive Learning Strategy from MLE viewpoint

Given labeled data $\{x_i, y_i\}_{i=1}^N$, we want to train a D-th order polynomial regression model

$$p(x) = \sum_{j=1}^{D} w_{j-1} x^{j-1} + \epsilon$$

Assume noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Recall that $y = \boldsymbol{x}^\top \boldsymbol{w} + \boldsymbol{\epsilon} \Rightarrow y - \boldsymbol{x}^\top \boldsymbol{w} \sim \mathcal{N}(0, \sigma^2)$. Then we have

$$p(y|\boldsymbol{x}, \boldsymbol{w}) = p(y - \boldsymbol{x}^\top \boldsymbol{w}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y - \boldsymbol{x}^\top \boldsymbol{w})^2}{2\sigma^2}\right)$$

Given i.i.d. $\{x_i, y_i\}_{i=1}^N$, by MLE, the objective function is

$$\begin{aligned} &\max_{\boldsymbol{w}} \prod_{i=1}^{N} p(y_i|\boldsymbol{x}_i, \boldsymbol{w}) \iff \min_{\boldsymbol{w}} -\sum_{i=1}^{N} \log p(y_i|\boldsymbol{x}_i, \boldsymbol{w}) \\ \iff &\min_{\boldsymbol{w}} \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - \boldsymbol{x}_i^\top \boldsymbol{w})^2 - \text{Const.} \\ \iff &\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 \end{aligned}$$

So we can learn the model via $\min_{\boldsymbol{w}} \underbrace{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_p^p}_{L(\boldsymbol{w}, \boldsymbol{X}, \boldsymbol{y})}$. Specifically, we have

$$\frac{\partial L(\boldsymbol{w}, \boldsymbol{X}, \boldsymbol{y})}{\partial \boldsymbol{w}} = -2\boldsymbol{X}^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) = 0 \Rightarrow \boldsymbol{w} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

Time Complexity: The operations involved to computed $\boldsymbol{w}^* = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$ is $\mathcal{O}(ND^2 + D^3)$

Stochastic Gradient Descent(SGD)

1. Initialize $\boldsymbol{w}^{(0)}$ randomly
2. At iteration t, sample a batch of data $\{\boldsymbol{x}_i, y_i\}_{i=1}^{B}$ randomly
3. Compute the gradient $\dfrac{\partial L}{\partial \boldsymbol{w}} = 2\boldsymbol{X}_B^\top(\boldsymbol{X}_B\boldsymbol{w}_{t-1} - \boldsymbol{y}_B)$
4. Update $\boldsymbol{w}_t = \boldsymbol{w}_{t-1} - \tau\dfrac{\partial L}{\partial \boldsymbol{w}}$

2 Ordinary Linear Regression(OLR) and General Linear Model(GLM)

Linear Regression: OLR

Given arbitary ND-dimensional data $\boldsymbol{X} \in \mathbb{R}^{N\times D}$ and their corresponding labels $\boldsymbol{y} \in \mathbb{R}^N$, the OLR model is

$$y = \boldsymbol{x}^\top\boldsymbol{w} + \epsilon$$

And we learn the model via $\max\limits_{w} p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) \iff \min\limits_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2$.

[Note]: $\boldsymbol{X}$ are random variables, and a linear regression is interested in $\mathbb{E}[\boldsymbol{y}|\boldsymbol{X}]$.

Linear Regression: GLM

A natural extension of OLR, the general form is:

$$g(\mathbb{E}[\boldsymbol{y}|\boldsymbol{X}]) = \boldsymbol{X}\boldsymbol{w}$$

- **Linear Predictor:** $\eta = \boldsymbol{X}\boldsymbol{w}$
- **Link Function:** $g(\cdot)$. The link function connecting the prediction $\boldsymbol{\eta}$ and the conditional expectation $\mathbb{E}[\boldsymbol{y}|\boldsymbol{X}]$ can be nonlinear.
- **Distribution Family:** An **exponential family** of probability distributions to generate the output.

From the prospective of GLM, OLR ($y = \boldsymbol{x}^\top\boldsymbol{w} + \epsilon$) is a special case of GLM:

1. **Linear Predictor:** $\eta = \boldsymbol{x}^\top\boldsymbol{w}$
2. **Link Function:** $g^{-1}(\eta) = \eta$
3. **Distribution Family:** $y \sim \mathcal{N}(\boldsymbol{x}^\top\boldsymbol{w}, \sigma^2)$

The selection of link function is highly relevant to the distribution type: for $y = g^{-1}(\boldsymbol{x}^\top\boldsymbol{w}) \sim P$

- **Poisson Distribution:** $g(\cdot) = \log(\cdot)$
- **Gamma Distribution:** $g(\cdot) = \dfrac{1}{\cdot}$

The trade-off between bias and variance

Suppose that $\boldsymbol{w}$ is the **ground truth** parameter of a linear model. A set of data $(\boldsymbol{X}, \boldsymbol{y})$ are observed and yield

$$y = \boldsymbol{x}^\top \boldsymbol{w} + \epsilon = f_{\boldsymbol{w}}(\boldsymbol{x}) + \varepsilon,\ \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

And $\hat{\boldsymbol{w}}$ is the estimator obtained based on the data.

$$\begin{aligned}
MSE &= \mathbb{E}[(\boldsymbol{y} - \hat{\boldsymbol{y}})^2] = \mathbb{E}[(f_{\boldsymbol{w}}(\boldsymbol{x}) + \varepsilon - f_{\hat{\boldsymbol{w}}}(\boldsymbol{x}))^2] \\
&= \mathbb{E}[(f_{\boldsymbol{w}}(\boldsymbol{x}) - f_{\hat{\boldsymbol{w}}}(\boldsymbol{x}))^2] + \mathbb{E}[\varepsilon^2] + 2\mathbb{E}[(f_{\boldsymbol{w}}(\boldsymbol{x}) - f_{\hat{\boldsymbol{w}}}(\boldsymbol{x}))\varepsilon] \\
&= \mathbb{E}[(f_{\boldsymbol{w}}(\boldsymbol{x}) - f_{\hat{\boldsymbol{w}}}(\boldsymbol{x}))^2] + \sigma^2 \\
&= \mathbb{E}[(f_{\boldsymbol{w}}(\boldsymbol{x}) - \mathbb{E}[f_{\hat{\boldsymbol{w}}}(\boldsymbol{x})] + \mathbb{E}[f_{\hat{\boldsymbol{w}}}(\boldsymbol{x})] - f_{\hat{\boldsymbol{w}}}(\boldsymbol{x}))^2] + \sigma^2 \\
&= \mathbb{E}[(f_{\boldsymbol{w}}(\boldsymbol{x}) - \mathbb{E}[f_{\hat{\boldsymbol{w}}}(\boldsymbol{x})])^2] + \mathbb{E}[(\mathbb{E}[f_{\hat{\boldsymbol{w}}}(\boldsymbol{x})] - f_{\hat{\boldsymbol{w}}}(\boldsymbol{x}))^2] + \underbrace{2\mathbb{E}[(f_{\boldsymbol{w}}(\boldsymbol{x}) - \mathbb{E}[f_{\hat{\boldsymbol{w}}}(\boldsymbol{x})])(\mathbb{E}[f_{\hat{\boldsymbol{w}}}(\boldsymbol{x})] - f_{\hat{\boldsymbol{w}}}(\boldsymbol{x}))]}_{=0} + \sigma^2 \\
&= \mathbb{E}[(f_{\boldsymbol{w}}(\boldsymbol{x}) - \mathbb{E}[f_{\hat{\boldsymbol{w}}}(\boldsymbol{x})])^2] + \mathbb{E}[(\mathbb{E}[f_{\hat{\boldsymbol{w}}}(\boldsymbol{x})] - f_{\hat{\boldsymbol{w}}}(\boldsymbol{x}))^2] + \sigma^2 \\
&= \underbrace{(f_{\boldsymbol{w}}(\boldsymbol{x}) - \mathbb{E}[f_{\hat{\boldsymbol{w}}}(\boldsymbol{x})])^2}_{\text{Bias}^2(f_{\boldsymbol{w}}(\boldsymbol{x}), f_{\hat{\boldsymbol{w}}}(\boldsymbol{x}))} + \underbrace{\mathbb{E}[(f_{\hat{\boldsymbol{w}}}(\boldsymbol{x}) - \mathbb{E}[f_{\hat{\boldsymbol{w}}}(\boldsymbol{x})])^2]}_{\text{Variance}(f_{\hat{\boldsymbol{w}}}(\boldsymbol{x}))} + \underbrace{\sigma^2}_{\text{Irreducible Noise}}
\end{aligned}$$

This shows a trade-off between bias and variance:

- **Bias ⇒ Underfitting:** A high bias indicates that the model is **too simple and is missing** relevant relations between input and output variables.
- **Variance ⇒ Overfitting:** A high variance indicates that the model is **highly sensitive to the specific data it was trained on** and does not generalize well to new data in the input data.

Our goal: **minimize** MSE so as to strike a balance between bias and variance.

Underfitting and Overfitting:

- **Underfitting:** Model complexity $\ll$ data complexity, the number of model parameters is smaller than that of data points
- **Overfitting:** Model complexity $\gg$ data complexity, the number of model parameters is larger than that of data points
  - Possibility 1: The model is too complex, need simplification.
  - Possibility 2: The model is reasonably complex, but the data are insufficient.

3 Ridge Regression

Ridge Regression

To learn complicated models from **relatively sparse data**, Ridge Regression is introduced to impose side information on the model paramters. Ridge Regression targets at **minimizing** MSE **with** L_2 **regularization**:

$$\min_{\boldsymbol{w}} L(\boldsymbol{w}) = \min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda \|\boldsymbol{w}\|_2^2$$

which consider the data fidelity and penalize the energy of parameters.

Bayesian Viewpoint of Ridge Regression

The principle of Ridge Regression can be considered in a Bayesian viewpoint. Assume noise $y - \boldsymbol{x}^\top \boldsymbol{w} = \epsilon \sim \mathcal{N}(0, \sigma^2)$ and **weight has a Gaussian prior** $\boldsymbol{w} \sim \mathcal{N}(0, \gamma^2 \boldsymbol{I})$. Then by MAP

$$\begin{aligned}
&\max_{\boldsymbol{w}} p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}) \propto \max_{\boldsymbol{w}} p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) p(\boldsymbol{w}) \\
\Rightarrow &\max_{\boldsymbol{w}} \prod_{i=1}^{N} p(y_i|\boldsymbol{x}_i, \boldsymbol{w}) p(\boldsymbol{w}) \\
\Rightarrow &\min_{\boldsymbol{w}} -\sum_{i=1}^{N} \log p(y_i|\boldsymbol{x}_i, \boldsymbol{w}) - \log p(\boldsymbol{w})
\end{aligned}$$

Note that

$$\begin{aligned}
\sum_{i=1}^{n} \log p(y_n|\boldsymbol{x}_n, \boldsymbol{w}) &= \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \boldsymbol{x}_i^\top \boldsymbol{w})^2}{2\sigma^2}\right) \\
&= -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \boldsymbol{x}_i^\top \boldsymbol{w})^2 + \text{Const.} \\
&= -\frac{1}{2\sigma^2} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \text{Const.}
\end{aligned}$$

$$\begin{aligned}
\log p(\boldsymbol{w}) &= \log \frac{1}{(2\pi)^{\frac{D}{2}} |\gamma^2 \boldsymbol{I}|^{\frac{1}{2}}} \exp\left(-\frac{\boldsymbol{w}^\top (\gamma^2 \boldsymbol{I})^{-1} \boldsymbol{w}}{2}\right) \\
&= \log \frac{1}{(2\pi)^{\frac{D}{2}} |\gamma^2 \boldsymbol{I}|^{\frac{1}{2}}} - \frac{1}{2\gamma^2} \boldsymbol{w}^\top \boldsymbol{w} \\
&= -\frac{1}{2\gamma^2} \|\boldsymbol{w}\|_2^2 + \text{Const.}
\end{aligned}$$

Therefore, the objective function can be written as

$$\min_{\boldsymbol{w}} \frac{1}{2\sigma^2} |\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}|_2^2 + \frac{1}{2\gamma^2} \|\boldsymbol{w}\|_2^2 + \text{Const.}$$

Here, $\frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2$ is the **likelihood term**, and $\frac{1}{2\gamma^2}\|\boldsymbol{w}\|_2^2$ is the **regularization term**. γ is the **hyper-parameter** corresponding to λ in the original Ridge Regression expression above that controls the strength of the regularization,

- $\lambda = 0$: Ridge Regression degenerates to OLR.
- $\lambda \to \infty$: $\boldsymbol{w} \to \boldsymbol{0}$. Intuitively, to minimize $L(\boldsymbol{w})$, $\boldsymbol{w}$ should be close to 0 to cancel out the penalization effect brought by λ.

Closed-form solution for Ridge Regression

The closed form solution of Ridge Regression is given by

$$\frac{\partial L}{\partial \boldsymbol{w}} = -2\boldsymbol{X}^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) + 2\lambda \boldsymbol{w} = 0 \Rightarrow \boldsymbol{w} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

Tikhonov Regularization

A variant of Ridge Regression use **Tikhonov Regularization**, shown as below

$$\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{\Gamma}\boldsymbol{w}\|_2^2$$

where $\boldsymbol{\Gamma}_{D\times D}$ is a Tikhonov matrix. The closed form solution of Tikhonov Regularization is similarily given by $\boldsymbol{w} = (\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{\Gamma}^\top\boldsymbol{\Gamma})^{-1}\boldsymbol{X}^\top\boldsymbol{y}$.

4 LASSO Regression

LASSO Regression

LASSO(Least Absolute Shrinkage and Selection Operator) Regression targets at **minimizing** MSE **with** L_1 **regularization**:

$$\min_{\boldsymbol{w}} L(\boldsymbol{w}) = \min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_1$$

Bayesian Viewpoint of LASSO Regression

The principle of LASSO Regression can be considered in a Bayesian viewpoint. Assume noise $y - \boldsymbol{x}^\top\boldsymbol{w} = \epsilon \sim \mathcal{N}(0, \sigma^2)$ and **weight has a Laplace prior** $\boldsymbol{w} \sim \mathcal{L}(0, b\boldsymbol{I})$. Then similarly, by MAP

$$\begin{aligned}
&\max_{\boldsymbol{w}} p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}) \propto \max_{\boldsymbol{w}} p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w})p(\boldsymbol{w}) \\
\Rightarrow &\max_{\boldsymbol{w}} \prod_{i=1}^{N} p(y_i|\boldsymbol{x}_i, \boldsymbol{w})p(\boldsymbol{w}) \\
\Rightarrow &\min_{\boldsymbol{w}} -\sum_{i=1}^{N} \log p(y_i|\boldsymbol{x}_i, \boldsymbol{w}) - \log p(\boldsymbol{w})
\end{aligned}$$

Note that

$$\begin{aligned}
\log p(\boldsymbol{w}) &= \log \frac{1}{(2b)^D} \exp\left(-\frac{\|\boldsymbol{w}\|_1}{b}\right) \\
&= -\frac{1}{b}\|\boldsymbol{w}\|_1 + \text{Const.}
\end{aligned}$$

Therefore, the objective function can be written as

$$\min_{\boldsymbol{w}} \frac{1}{2\sigma^2}\,|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}|_2^2 + \frac{\lambda}{b}\,\|\boldsymbol{w}\|_1 + \text{Const.}$$

Closed-form solution for LASSO

The **closed-form solution of LASSO Regression** can be obtained by **soft-thresholding**: When $\boldsymbol{X} = [\boldsymbol{x_1}, \cdots, \boldsymbol{x_D}] \in \mathbb{R}^{N\times D}$ are orthonormal, i.e. $\boldsymbol{X}^\top\boldsymbol{X} = \boldsymbol{I_D}$, the closed-form solution of LASSO Regression under soft-thresholding can written as

$$\hat{w}_d^* = S_\lambda(\hat{w}_{OLS,d}) = \text{sign}(\hat{w}_{OLS,d}) \cdot \max\{|\hat{w}_{OLS,d}| - \lambda, 0\},\ d = 1, 2, \cdots, D$$

Note that $\boldsymbol{w}_{OLS} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y} = \boldsymbol{X}^\top \boldsymbol{y}$. We can take the derivative of $L(\boldsymbol{w})$ with respect to $\boldsymbol{w}$:

$$\begin{aligned}\frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}} &= -\frac{1}{2}\cdot 2\boldsymbol{X}^\top(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}) + \lambda\cdot \text{sign}(\boldsymbol{w}) \\ &= \boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{w} - \boldsymbol{X}^\top\boldsymbol{y} + \lambda\cdot\text{sign}(\boldsymbol{w}) \\ &= \boldsymbol{w} - \boldsymbol{X}^\top\boldsymbol{y} + \lambda\cdot\text{sign}(\boldsymbol{w}) \\ \Rightarrow \boldsymbol{w}^* &= \boldsymbol{w}_{OLS} - \lambda\cdot\text{sign}(\boldsymbol{w}^*)\end{aligned}$$

where $\text{sign}(x) = \begin{cases} 1, & x>0 \\ \text{any value between } [-1,1], & x=0 \\ -1, & x<0 \end{cases}$. Since L_1 is separable and thus we consider each of its components separately, so we consider the i-th component of $\boldsymbol{w}$:

$$w_i^* = w_{OLS,i} - \lambda\cdot\text{sign}(w_i^*)$$

Note that when $w_i^* \neq 0$

$$\begin{aligned}w_i^* < 0, w_{OLS,i} - \lambda(-1) < 0 \Rightarrow w_{OLS,i} < -\lambda \\ w_i^* > 0, w_{OLS,i} - \lambda(1) > 0 \Rightarrow w_{OLS,i} > \lambda\end{aligned}$$

Therefore, when $|w_{OLS,i}| > \lambda > 0$, it's equivalent to write

$$w_i^* = w_{OLS,i} - \lambda\cdot\text{sign}(w_{OLS,i})$$

When $w_i^* = 0$, it follows that

$$0 \in w_{OLS,i} - \lambda\cdot[-1,1] \Rightarrow |w_{OLS,i}| \leq \lambda$$

So that

$$\begin{aligned}w_i^* = S_\lambda(w_{OLS,i}) &= \begin{cases} w_{OLS,i} - \lambda\cdot\text{sign}(w_{OLS,i}), & |w_{OLS,i}|>\lambda \\ 0, & |w_{OLS,i}|\leq\lambda \end{cases} \\ &= \begin{cases} w_{OLS,i}\cdot\text{sign}(w_{OLS,i}) - \lambda\cdot\text{sign}(w_{OLS,i}), & |w_{OLS,i}|>\lambda \\ 0, & |w_{OLS,i}|\leq\lambda \end{cases} \\ &= \text{sign}(w_{OLS,i})\cdot\max\{|w_{OLS,i}|-\lambda, 0\}\end{aligned}$$

Hence the closed-form solution of LASSO Regression under soft-thresholding can written as

$$w_d^* = S_\lambda(w_{OLS,d}) = \text{sign}(w_{OLS,d})\cdot\max\{|w_{OLS,d}|-\lambda,0\},\ d=1,2,\cdots,D$$

Iterative soft-thresholding for general situations

More generally, when $\boldsymbol{X}^\top\boldsymbol{X} \neq \boldsymbol{I_D}$, we can construct orthonormal vectors column-wisely and update

parameters iteratively. In the t-th iteration, we update all $d = 1, \cdots, D$-th parameter by

$$\begin{aligned}
\hat{w}_d^{(t+1)} &= \arg\min_w \frac{1}{2}\|\boldsymbol{y} - \sum_{i \neq d} \boldsymbol{x}_i w_i^{(t)} - \boldsymbol{x}_d w\|_2^2 + \lambda|w| \\
&= \arg\min_{w_d} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}_{-d}\boldsymbol{w}_{-d}^{(t)} - \boldsymbol{x}_d w\|_2^2 + \lambda|w| \\
&= \arg\min_{w_d} \frac{1}{2}\|\frac{1}{\|\boldsymbol{x_d}\|_2}(\boldsymbol{y} - \boldsymbol{X}_{-d}\boldsymbol{w}_{-d}^{(t)}) - w \cdot \frac{\boldsymbol{x}_d}{\|\boldsymbol{x_d}\|_2}\|_2^2 + \lambda\frac{|w|}{\|\boldsymbol{x}_d\|_2^2} \\
&= S_{\frac{\lambda}{\|\boldsymbol{x}_d\|_2^2}}\left(\frac{\boldsymbol{x}_d^\top(\boldsymbol{y} - \boldsymbol{X}_{-d}\boldsymbol{w}_{-d}^{(t)})}{\|\boldsymbol{x_d}\|_2^2}\right)
\end{aligned}$$

Denote $\boldsymbol{e}^{(t)} = \boldsymbol{y} - \boldsymbol{X_{-d}}\boldsymbol{w_{-d}^{(t)}}$, then the update rule can be written as

$$S_{\frac{\lambda}{\|\boldsymbol{x}_d\|_2^2}}\left(\frac{\boldsymbol{x}_d^\top(\boldsymbol{y} - \boldsymbol{X}_{-d}\boldsymbol{w}_{-d}^{(t)})}{\|\boldsymbol{x_d}\|_2^2}\right) = \text{sign}(\frac{\boldsymbol{x}_d^\top\boldsymbol{e}^{(t)}}{\|\boldsymbol{x_d}\|_2^2}) \cdot \max\{\frac{|\boldsymbol{x}_d^\top\boldsymbol{e}^{(t)}|}{\|\boldsymbol{x_d}\|_2^2} - \frac{\lambda}{\|\boldsymbol{x}_d\|_2^2}, 0\}$$

One can easily tell that to compute the d-th dimension of $\boldsymbol{w}^{(t+1)}$, we need to use all the dimensions of $\boldsymbol{w}^{(t)}$.

Proof. Our goal is to minimize the following objective function

$$L(w) = \frac{1}{2}\|\frac{1}{\|\boldsymbol{x_d}\|_2}(\boldsymbol{y} - \boldsymbol{X}_{-d}\boldsymbol{w}_{-d}^{(t)}) - w \cdot \frac{\boldsymbol{x}_d}{\|\boldsymbol{x_d}\|_2}\|_2^2 + \frac{\lambda}{\|\boldsymbol{x}_d\|_2^2}|w|$$

Recall that when the column vectors $\boldsymbol{X} = [\boldsymbol{x_1}, \cdots, \boldsymbol{x_D}] \in \mathbb{R}^{N \times D}$ are orthogonal, we can use closed form solution for LASSO by soft thresholding straightforwardly. Here, since we are iteratively constructing orthonormal vectors $\frac{\boldsymbol{x_d}}{\|\boldsymbol{x_d}\|_2}$ column-wisely when optimizing each $\hat{w}_d{}^{(t+1)}$, we can first calculate $\hat{w}_{OLS,d}$ and plug it into the soft-thresholding function $S_\lambda(\cdot)$ to get the optimal $\hat{w}_d^{(t+1)}$.

$$\begin{aligned}
L(w) &= \frac{1}{2\|\boldsymbol{x_d}\|_2}\left(\boldsymbol{y} - \boldsymbol{X_{-d}}\boldsymbol{w_{-d}^{(t)}} - w \cdot \boldsymbol{x_d}\right)^\top\left(\boldsymbol{y} - \boldsymbol{X_{-d}}\boldsymbol{w_{-d}^{(t)}} - w \cdot \boldsymbol{x_d}\right) + \frac{\lambda}{\|\boldsymbol{x}_d\|_2^2}|w| \\
&= \frac{1}{2\|\boldsymbol{x_d}\|_2}(\boldsymbol{y}^\top\boldsymbol{y} - \boldsymbol{y}^\top\boldsymbol{X_{-d}}\boldsymbol{w_{-d}^{(t)}} - \boldsymbol{y}^\top w\boldsymbol{x_d} - (\boldsymbol{w_{-d}^{(t)}})^\top\boldsymbol{X}_{-d}^\top\boldsymbol{y} + (\boldsymbol{w_{-d}^{(t)}})^\top\boldsymbol{X}_{-d}^\top\boldsymbol{X_{-d}}\boldsymbol{w_{-d}^{(t)}} + (\boldsymbol{w_{-d}^{(t)}})^\top\boldsymbol{X}_{-d}^\top w\boldsymbol{x_d} \\
&\quad - w\boldsymbol{x_d}^\top\boldsymbol{y} + w\boldsymbol{x_d}^\top\boldsymbol{X_{-d}}\boldsymbol{w_{-d}^{(t)}} + w^2\boldsymbol{x_d}^\top\boldsymbol{x_d}) + \frac{\lambda}{\|\boldsymbol{x}_d\|_2^2}|w| \\
\Rightarrow \frac{\partial L(w)}{\partial w} &= \frac{1}{2\|\boldsymbol{x_d}\|_2}(-2\boldsymbol{y}^\top\boldsymbol{x_d} + 2(\boldsymbol{w_{-d}^{(t)}})^\top\boldsymbol{X}_{-d}^\top\boldsymbol{x_d} + 2w\boldsymbol{x_d}^\top\boldsymbol{x_d}) + \frac{\lambda}{\|\boldsymbol{x}_d\|_2^2}\text{sign}(w) = 0 \\
&= \frac{1}{\|\boldsymbol{x_d}\|_2}(w\boldsymbol{x_d}^\top\boldsymbol{x_d} + \boldsymbol{x_d}^\top(\boldsymbol{X_{-d}}\boldsymbol{w_{-d}^{(t)}} - \boldsymbol{y})) + \frac{\lambda}{\|\boldsymbol{x}_d\|_2^2}\text{sign}(w) = 0
\end{aligned}$$

Denote $L' = \frac{1}{2}\|\frac{1}{\|\boldsymbol{x_d}\|_2}(\boldsymbol{y} - \boldsymbol{X}_{-d}\boldsymbol{w}_{-d}^{(t)}) - w \cdot \frac{\boldsymbol{x}_d}{\|\boldsymbol{x_d}\|_2}\|_2^2$, $\lambda' = \frac{\lambda}{\|\boldsymbol{x}_d\|_2^2}$. We can derive the OLS estimator $\hat{w}_{OLS,d}$ by taking the derivative of L' w.r.t w:

$$\frac{\partial L'}{\partial w} = \frac{1}{\|\boldsymbol{x_d}\|_2}(w\boldsymbol{x_d}^\top\boldsymbol{x_d} + \boldsymbol{x_d}^\top(\boldsymbol{X_{-d}}\boldsymbol{w_{-d}^{(t)}} - \boldsymbol{y})) = 0 \Rightarrow \hat{w}_{OLS,d} = \frac{\boldsymbol{x_d}^\top(\boldsymbol{y} - \boldsymbol{X_{-d}}\boldsymbol{w_{-d}^{(t)}})}{\|\boldsymbol{x_d}\|_2^2}$$

Plug $\hat{w}_{OLS,d}$ into the soft thresholding function $S_{\lambda'}(\cdot)$, we can get the optimal $\hat{w}_d^{(t+1)}$:

$$\hat{w}_d^{(t+1)} = S_{\lambda'}(\hat{w}_{OLS,d}) = S_{\frac{\lambda}{\|\boldsymbol{x}_d\|_2^2}}\left(\frac{\boldsymbol{x}_d^\top(\boldsymbol{y} - \boldsymbol{X}_{-d}\boldsymbol{w}_{-d}^{(t)})}{\|\boldsymbol{x_d}\|_2^2}\right)$$

□

4.1 Achieving Model/Feature Selection Explicitly

This is achieved by minimizing MSE with an explicit sparsity constraint

$$\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 \text{ s.t. } \|\boldsymbol{w}\|_0 \leq L$$

- Because $\boldsymbol{w}$ is sparse, some features are ignored, leading to feature selection.
- Because $\boldsymbol{w}$ is sparse, the number of model parameters is regularized, leading to model selection.

4.2 Ways of Dealing with Outliners

Iteratively Reweighted Least Squares

Denote $\alpha_n(\boldsymbol{w}^{(t)}) = |y_n - \boldsymbol{x}_n^\top \boldsymbol{w}^{(t)}|^{-1}$ with $\alpha_n(\boldsymbol{w}^{(0)}) = 1$, the t-th iteration of IRLS is given by

$$\boldsymbol{w}^{(t+1)} = \arg\min_{\boldsymbol{w}} \sum_{n=1}^{N} \alpha_n(\boldsymbol{w}^{(t)})|y_n - \boldsymbol{x}_n^\top \boldsymbol{w}|^2$$

Denote $\boldsymbol{\alpha}(\boldsymbol{w}^{(t)}) = \begin{bmatrix} \alpha_1(\boldsymbol{w}^{(t)}) \\ \alpha_2(\boldsymbol{w}^{(t)}) \\ \vdots \\ \alpha_N(\boldsymbol{w}^{(t)}) \end{bmatrix}$, where $\alpha_n(\boldsymbol{w}^{(t)})$ is the t-th iteration of n-th dimension, we can write the objective function as

$$\begin{aligned}
\boldsymbol{w}^{(t+1)} &= \arg\min_{\boldsymbol{w}} \sum_{n=1}^{N} \left(\sqrt{\alpha_n(\boldsymbol{w}^{(t)})}\right)^2 |y_n - \boldsymbol{x}_n^\top \boldsymbol{w}|^2 \\
&= \arg\min_{\boldsymbol{w}} \|\text{diag}\left(\boldsymbol{\alpha}(\boldsymbol{w}^{(t)})\right)^{\frac{1}{2}} \cdot (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})\|_2^2 = \arg\min_{\boldsymbol{w}} \|(\boldsymbol{A}^{(t)})^{\frac{1}{2}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})\|_2^2
\end{aligned}$$

where $\boldsymbol{A}^{(t)} = \text{diag}(\boldsymbol{\alpha}(\boldsymbol{w}^{(t)})) = \begin{bmatrix} \frac{1}{|y_1 - \boldsymbol{x}_1^\top \boldsymbol{w}^{(t)}|} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{|y_N - \boldsymbol{x}_N^\top \boldsymbol{w}^{(t)}|} \end{bmatrix}$. To derive the closed-form iterative steps of IRLS, we first denote

$$\begin{aligned}
L &= \|(\boldsymbol{A}^{(t)})^{\frac{1}{2}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})\| = (\boldsymbol{y}^\top - w^\top \boldsymbol{X}^\top)((\boldsymbol{A}^{(t)})^{\frac{1}{2}})^\top (\boldsymbol{A}^{(t)})^{\frac{1}{2}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) \\
&= (\boldsymbol{y}^\top - w^\top \boldsymbol{X}^\top)\boldsymbol{A}^{(t)}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) \\
&= \boldsymbol{y}^\top \boldsymbol{A}^{(t)}\boldsymbol{y} - \boldsymbol{y}^\top \boldsymbol{A}^{(t)}\boldsymbol{X}\boldsymbol{w} - \boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{A}^{(t)}\boldsymbol{y} + \boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{A}^{(t)}\boldsymbol{X}\boldsymbol{w} \\
\frac{\partial L}{\partial \boldsymbol{w}} &= -2\boldsymbol{X}^\top \boldsymbol{A}^{(t)}\boldsymbol{y} + 2\boldsymbol{X}^\top \boldsymbol{A}^{(t)}\boldsymbol{X}\boldsymbol{w} = 0 \\
\Rightarrow \boldsymbol{w}^{(t+1)} &= (\boldsymbol{X}^\top \boldsymbol{A}^{(t)}\boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{A}^{(t)}\boldsymbol{y}
\end{aligned}$$

Since we know $\boldsymbol{w}^{(t)}$ and $\boldsymbol{A}^{(t)}$, we can calculate $\boldsymbol{w}^{(t+1)}$ iteratively.

Explanation: The idea of IRLS is to simply treat $\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \sum_{n=1}^{N} |y_n - \boldsymbol{x}_n^\top \boldsymbol{w}|^2$, but with a weight $\alpha_n(\boldsymbol{w}^{(t)})$ attached to each n. The initial weights are set as $\boldsymbol{\alpha} = \mathbf{1}$. **In each iteration, we can lower an observation's importance (i.e. weight) by $\alpha_n(\boldsymbol{w}^{(t)}) = |y_n - \boldsymbol{x}_n^\top \boldsymbol{w}^{(t)}|^{-1}$ if it has a large residual (i.e. outliners).**

- Naturally, IRLS works for p-norm with $p < 2$: $\alpha_n^{(t)} = |y_n - \boldsymbol{x}_n^\top \boldsymbol{w}^{(t)}|^{p-2}$.

4.3 Comparison between Ridge and LASSO

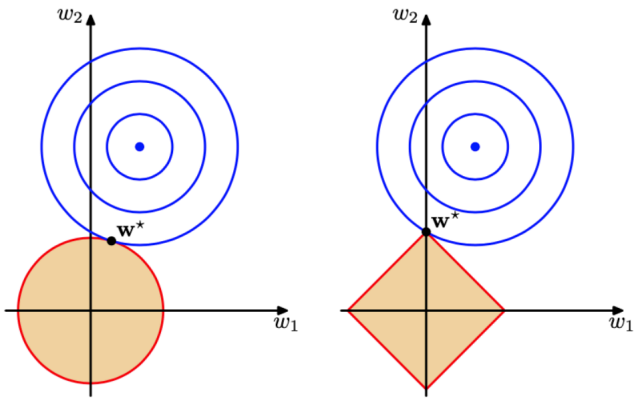
Ridge Regression

- Penalize the energy of parameters.
- **Strictly convex** and easy to solve with linear convergence

LASSO Regression

- **Convex but nonsmooth**, relatively hard to solve with sublinear convergence.
- Penalize the sparsity of parameters $\Rightarrow$ **beneficial for model and feature selection for high dimensional data with p features $\gg$ n data points.**

As is shown in the picture above, the feasible region of LASSO is a diamond, and the solution **is likely to be on the axis, meaning that some w_i are set to 0**, while the feasible region of Ridge is a circle, and the tangency point is likely to be on the circle, making it harder to set $w_i = 0$.
In summary, the **sparsity essence** of LASSO can set some $w_i = 0$, which is equivalent to **feature selection**.



5 Elastic Net Regularization

Elastic Net Regularization

Elastic Net Regularization is a combination of L_1 **and** L_2 **regularization**:

$$\min_{\boldsymbol{w}} L(\boldsymbol{w}) = \min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda_1\|\boldsymbol{w}\|_1 + \lambda_2\|\boldsymbol{w}\|_2^2$$

Closed-form solution for Elastic net Regularization

The basic idea is to integrate the L_2 regularization term into MSE $\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\|_2^2$ and then apply the iterative soft-thresholding w.r.t. L_1 regularization. Note that the term $\lambda_2\|\boldsymbol{w}\|_2^2$ can be written as

$$\lambda_2\|\boldsymbol{w}\|_2^2 = \|\sqrt{\lambda_2}\boldsymbol{w}\|_2^2 = \|\mathbf{0}_D - \sqrt{\lambda_2}\boldsymbol{I_D}\boldsymbol{w}\|_2^2$$

where $\mathbf{0}_D$ is a D-dimensional zero vector and $\boldsymbol{I_D}$ is a $D\times D$ identity matrix. We can therefore integrate the L_2 regularization term into the MSE term as

$$\begin{aligned}
&\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\|_2^2 + \|\mathbf{0}_D - \sqrt{\lambda_2}\boldsymbol{I_D}\boldsymbol{w}\|_2^2 + \lambda_1\|\boldsymbol{w}\|_1 \\
=&\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\|_2^2 + \frac{1}{2}\|\mathbf{0}_D - \sqrt{2\lambda_2}\boldsymbol{I_D}\boldsymbol{w}\|_2^2 + \lambda_1\|\boldsymbol{w}\|_1 \\
=&\frac{1}{2}\left\|\begin{bmatrix}\boldsymbol{y}\\ \mathbf{0}_D\end{bmatrix} - \begin{bmatrix}\boldsymbol{X}\\ \sqrt{2\lambda_2}\boldsymbol{I_D}\end{bmatrix}\boldsymbol{w}\right\|_2^2 + \lambda_1\|\boldsymbol{w}\|_1
\end{aligned}$$

And this goes back to L_1 regularization situation and we can put new matrix

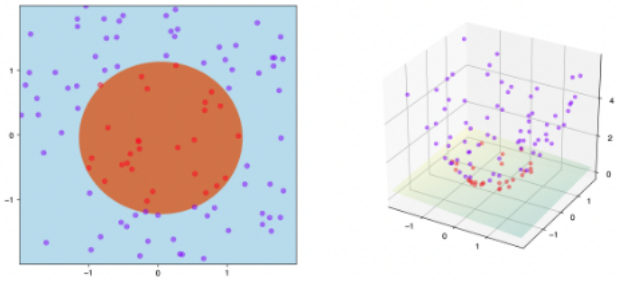
$$\boldsymbol{X'} = \begin{bmatrix}\boldsymbol{X}\\ \sqrt{2\lambda_2}\boldsymbol{I_D}\end{bmatrix}, \boldsymbol{y'} = \begin{bmatrix}\boldsymbol{y}\\ \mathbf{0}_D\end{bmatrix}$$

into the iterative soft-thresholding solver for LASSO.

6 Kernel Regression

6.1 Kernel

Practically, it's hard to achieve linear dependence and seperation of data in low dimensional spaces since their ability to represent data has reached the limit. An intuitive method of solving this problem is to map the data into a higher dimensional space where the data is linearly separable.



We need to find an appropriate mapping from low dimensional space to high dimensional space, making the mapped data somehow "linear seperable". The **"mapping"** can be achieved by **"kernel"**.

Kernel

Define a evaluation function L_x on Hilbert Space $\mathcal{H}$

$$\begin{aligned}
L_x : \mathcal{H} &\longmapsto \mathbb{R} \\
\phi &\longmapsto L_{\boldsymbol{x}}(\phi) = \phi(\boldsymbol{x})
\end{aligned}$$

where L_x is a bounded operator on $\mathcal{H}$. Consider the inner product on $\mathcal{H}$, satisfying

$$\forall \boldsymbol{x} \in \mathcal{X}, \exists M_{\boldsymbol{x}} > 0, \forall \phi \in \mathcal{H}, |L_{\boldsymbol{x}}(\phi)| \leq M_{\boldsymbol{x}} \langle \phi, \phi \rangle_{\mathcal{H}}$$

$\forall \boldsymbol{x} \in \mathcal{X}$, $\exists K_{\boldsymbol{x}} \in \mathcal{H}$, such that

$$f(\boldsymbol{x}) = L_{\boldsymbol{x}}(f) = \langle f, K_{\boldsymbol{x}} \rangle_{\mathcal{H}} = \int_{\boldsymbol{y} \in \mathcal{X}} f(\boldsymbol{y}) K_{\boldsymbol{x}}(\boldsymbol{y}) \mathrm{d}\boldsymbol{y}$$

Then we can define a **Reproducing Kernel** K on $\mathcal{H}$,

$$\begin{aligned} K : \mathcal{X} \times \mathcal{X} &\longmapsto \mathbb{R} \\ (\boldsymbol{x}, \boldsymbol{y}) &\longmapsto K(\boldsymbol{x}, \boldsymbol{y}) = \langle K_{\boldsymbol{x}}, K_{\boldsymbol{y}} \rangle_{\mathcal{H}} = K_{\boldsymbol{x}}(\boldsymbol{y}) = K_{\boldsymbol{y}}(\boldsymbol{x}) \end{aligned}$$

The Hilbert Sapce $\mathcal{H}$ is called **Reproducing Kernel Hilbert Space** (RKHS) with the reproducing kernel K.

Feature Space

Define a mapping

$$\begin{aligned} \phi : \mathcal{X} &\longmapsto \mathcal{H} \\ \boldsymbol{x} &\longmapsto K_x \end{aligned}$$

Since we have finite data point, the **Feature Space** can be expressed as

$$\mathcal{F} = \mathrm{span}\{\{K_{\boldsymbol{x}}\}_{x \in \mathcal{X}}\} \subset \mathcal{H}_K$$

that the feature space is a subspace of the RKHS.

The intuition of kernel is that it measures the similarity between two inputs x, z in characteristic space $\mathcal{H}$.

- Feature mapping function $\phi(x)$ maps input x to a higher dimensional space $\mathcal{H}$.
- Kernel function $K(x, z)$ calculates the **inner product of** x **and** z in high dimensional space $\mathcal{H}$ **without knowing the explicit form of** $\phi(x)$.
- Inner product $\langle \phi(x), \phi(z) \rangle_{\mathcal{H}}$ usually measures the similarity between x and z in $\mathcal{H}$. **For example, in Euclidean Space, the inner product is the square of the Euclidean distance.**

Valid Kernel Function

Valid Kernel Functions should satisfy:

(1) The Gram matrix $\boldsymbol{K} = [K(\boldsymbol{x_i}, \boldsymbol{x_j})] \in \mathbb{R}^{N \times N}$ is positive definite.
$\iff$ (2) Mercer's Theorem: for all square-integrable function $g(x)$

$$\int_{\mathcal{X} \times \mathcal{X}} g(x) K(x, z) g(z) \mathrm{d}x \mathrm{d}z \geq 0$$

[Note]: Some common kernel functions include

- **Linear Kernel:** $K(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x}^\top \boldsymbol{z}$
- **Polynomial Kernel:** $K(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}^\top \boldsymbol{z})^d$ where d is the degree of polynomial

- **Radial Basis Function Kernel:** $K(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|_2^2}{2\sigma^2}\right)$ where $\sigma > 0$ is the bandwidth
- **Laplace Kernel:** $K(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|_1}{\sigma}\right)$ where $\sigma > 0$ is the bandwidth
- **Sigmoid Kernel:** $K(\boldsymbol{x}, \boldsymbol{z}) = \tanh(\alpha \boldsymbol{x}^\top \boldsymbol{z} + \theta)$ with $\alpha > 0$ and $\theta < 0$

The properties of kernel functions include: if K_1 and K_2 are valid kernel functions, then

- **Linearity:** $\alpha K_1(\boldsymbol{x}, \boldsymbol{z}) + \beta K_2(\boldsymbol{x}, \boldsymbol{z})$ is a valid kernel function.
- **Product:** $K_1(\boldsymbol{x}, \boldsymbol{z}) K_2(\boldsymbol{x}, \boldsymbol{z})$ is a valid kernel function.
- $\forall f(\boldsymbol{x})$, $f(\boldsymbol{x}) K_1(\boldsymbol{x}, \boldsymbol{z}) f(\boldsymbol{z})$ is a valid kernel function.

6.2 Nadaraya-Watson Kernel Regression

Nadaraya-Watson Kernel Regression

Given $\{x_n, y_n\}_{n=1}^N$, for arbitary new input x, its output y can be estimated by

$$\hat{y} = \hat{f}_h(x) = \sum_{n=1}^{N} \frac{\kappa_h(x - x_n)}{\sum_{n=1}^{N} \kappa_h(x - x_n)} \cdot y_n$$

where $\kappa_h(x)$ is the kernel function with bandwidth h.

[Note]: The notion of $\hat{f}_h(x)$ is that it uses a measure function (kernel) to measure the similarity between x and x_n, and then **use the similarity to weigh the output y_n to get the output y.**

Representor Theorem

A minimizer f^* of **a regularized empirical risk functional** defined over a RKHS can be represented as **a finite linear combination of kernel products** evaluated on the input points in the training set data. Mathematically,

$$f^* := \arg\min_{f \in \mathcal{H}_K} \mathbb{E}_{x,y \sim P_{\mathcal{D}}}[\text{Loss}(y, f(x))] + \mathcal{R}(f) \iff \exists \boldsymbol{\alpha} \in \mathbb{R}^M, \text{ s.t.} f^*(x) = \sum_{n=1}^{M} \alpha_n K(x, x_n), \ M \leq |\mathcal{D}|$$

6.3 Kernel Ridge Regression

Closed-form Solution for Kernel Ridge Regression

Our model is $y = f(\boldsymbol{x}) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2), f \in \mathcal{H}_K$. Given $\{\boldsymbol{x_n}, y_n\}_{n=1}^N$, the objective function is

$$f^* = \min_{f \in \mathcal{H}_K} \sum_{n=1}^{N} (y_n - f(\boldsymbol{x}_n))^2 + \lambda \|f\|_{\mathcal{H}_K}^2$$

By Representor Theorem, we have

$$f^*(\boldsymbol{x}) = \sum_{n=1}^{N} \alpha_n K(\boldsymbol{x}, \boldsymbol{x}_n)$$

Then, we rewrite the objective function by replacing f with its representation

$$\begin{aligned}
&\sum_{n=1}^{N}(y_n - \sum_{m=1}^{N}\alpha_m K(\boldsymbol{x}_n, \boldsymbol{x}_m))^2 + \lambda \sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m K(\boldsymbol{x}_n, \boldsymbol{x}_m) \\
=&\sum_{n=1}^{N}(y_n - \boldsymbol{K}_n^\top \boldsymbol{\alpha})^2 + \lambda \boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha} \qquad \text{(Here } \boldsymbol{K} \in \mathbb{R}^{N\times N}, \boldsymbol{K}_n^\top \in \mathbb{R}^{1\times N}, \text{the } n\text{-th row of } \boldsymbol{K}) \\
=&\|\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\alpha}\|_2^2 + \lambda \boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha}
\end{aligned}$$

Therefore, our optimization target is

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\alpha}\|_2^2 + \lambda \boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha}$$

We take the gradient of loss function to get the closed-form solution of $\boldsymbol{\alpha}^*$

$$\begin{aligned}
\frac{\partial L}{\partial \alpha} =& 2(-\boldsymbol{K})^\top(\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\alpha}) + \lambda(\boldsymbol{K} + \boldsymbol{K}^\top)\boldsymbol{\alpha} = -2\boldsymbol{K}^\top\boldsymbol{y} + 2\boldsymbol{K}^\top\boldsymbol{K}\boldsymbol{\alpha} + \lambda 2\boldsymbol{K}\boldsymbol{\alpha} \\
=& 2\left((\lambda\boldsymbol{K} + \boldsymbol{K}^\top\boldsymbol{K})\boldsymbol{\alpha} - \boldsymbol{K}^\top\boldsymbol{y}\right) \\
=& 2\left((\lambda\boldsymbol{K} + \boldsymbol{K}^2)\boldsymbol{\alpha} - \boldsymbol{K}\boldsymbol{y}\right) \\
=& 2\left((\lambda\boldsymbol{I} + \boldsymbol{K})\boldsymbol{K}\boldsymbol{\alpha} - \boldsymbol{K}\boldsymbol{y}\right) = 0 \\
\Rightarrow \boldsymbol{\alpha}^* =& (\lambda\boldsymbol{I} + \boldsymbol{K})^{-1}\boldsymbol{y}
\end{aligned}$$

Although we have no knowledge of the concrete form of mapping ϕ, we can still do calculations in the feature space $\mathcal{F}$ and solve the optimization problem by the self-defined inner product matrix $\boldsymbol{K}$. This is what we called as kernel trick.

Essentially, we can equate a non-linear regression problem to a linear regression problem in the feature space $\mathcal{F}$ by kernel trick. Specifically, when we use linear kernel $K(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top\boldsymbol{y}$, the kernel ridge regression degenerates to ridge regression in the original space.