Bias-Variance Tradeoff & Model Selection
Announcements

HW5 and P5 are coming out
Recap on Bias-Variance Tradeoff
Recap on Bias-Variance Tradeoff

Denote $h_{\mathcal{D}}$ as the ERM solution on dataset $\mathcal{D}$ w/ squared loss $\ell(h, x, y) = (h(x) - y)^2$
Recap on Bias-Variance Tradeoff

Denote $h_\mathcal{D}$ as the ERM solution on dataset $\mathcal{D}$ w/ squared loss $\ell(h, x, y) = (h(x) - y)^2$

What we have shown is the Bias-Variance decomposition:

$$
\mathbb{E}_{\mathcal{D}, x, y}(h_\mathcal{D}(x) - y)^2 = \mathbb{E}_{\mathcal{D}, x}(h_\mathcal{D}(x) - \bar{h}(x))^2 + \mathbb{E}_x(\bar{h}(x) - \bar{y}(x))^2 + \mathbb{E}_{x, y}(\bar{y}(x) - y)^2
$$

Variances $\bar{y} = \text{Bayes opt}$

$$
= \mathbb{E}(y | x)
$$
Recap on Bias-Variance Tradeoff

Denote $h_\mathcal{D}$ as the ERM solution on dataset $\mathcal{D}$ w/ squared loss $\ell(h, x, y) = (h(x) - y)^2$

What we have shown is the Bias-Variance decomposition:

$$\mathbb{E}_{\mathcal{D}, x, y}(h_\mathcal{D}(x) - y)^2 = \mathbb{E}_{\mathcal{D}, x}(h_\mathcal{D}(x) - \bar{h}(x))^2 + \mathbb{E}_x(\bar{h}(x) - \bar{y}(x))^2 + \mathbb{E}_{x, y}(\bar{y}(x) - y)^2$$
Outline of Today

1. Bias & Variance tradeoff demo on Ridge Linear Regression

2. Derivation of Bias / Variance for Ridge LR

2. Model selection in practice (Cross Validation)
Ridge Linear regression w/ fixed features and Gaussian noises

Let us consider the case where features are fixed, i.e., $x_1, \ldots, x_n$ fixed (no randomness)
Ridge Linear regression w/ fixed features and Gaussian noises

Let us consider the case where features are fixed, i.e., $x_1, \ldots, x_n$ fixed (no randomness)

But $y_i \sim (w^*)^\top x_i + \epsilon_i$, $\epsilon_i \sim \mathcal{N}(0,1)$

$$\mathbb{E}[y_i | x_i] = (w^*)^\top x_i$$
Ridge Linear regression w/ fixed features and Gaussian noises

Let us consider the case where features are fixed, i.e., $x_1, \ldots, x_n$ fixed (no randomness)

But $y_i \sim (w^*)^T x_i + \epsilon_i$, $\epsilon_i \sim \mathcal{N}(0,1)$

(This is called LR w/ fixed design)
Ridge Linear regression w/ fixed features and Gaussian noises

Let us consider the case where features are fixed, i.e., $x_1, \ldots, x_n$ fixed (no randomness)

But $y_i \sim (w^*)^T x_i + \epsilon_i$, $\epsilon_i \sim \mathcal{N}(0,1)$

(This is called LR w/ fixed design)

(So the only randomness of our dataset $\mathcal{D} = \{x_i, y_i\}$ is coming from the noises $\epsilon_i$)
Ridge Linear regression

Ridge Linear Regression formulation

\[ \hat{w} = \arg \min_w \sum_{i=1}^{n} (w^\top x_i - y_i)^2 + \lambda \| w \|_2^2 \]
Ridge Linear regression

Ridge Linear Regression formulation

$$\hat{w} = \arg \min_w \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \|w\|_2^2$$

What we will show now:

Larger $\lambda$ (model becomes “simpler”) $\Rightarrow$ larger bias, but smaller variance
Ridge Linear regression

Ridge Linear Regression formulation

\[ \hat{w} = \arg \min_w \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \| w \|_2^2 \]

What we will show now:

Larger \( \lambda \) (model becomes "simpler") \( \Rightarrow \) larger bias, but smaller variance

(Q: think about the case where \( \lambda \to +\infty \), what happens to \( \hat{w} \)?)
Ridge Linear regression

Demonstration for 2d ridge linear regression

1. We create 5000 datasets: $D_1, D_2, \ldots, D_{5000}$,

2. For a given $\lambda$, solve Ridge LR for each dataset, get $\hat{w}_1, \ldots, \hat{w}_{5000}$

3. Estimate the mean $\bar{w} = \sum_i \hat{w}_i / 5000$

4. Plot $\hat{w}_1, \ldots, \hat{w}_{5000}$, and mean $\bar{w}$, and the optimal $w^*$
Ridge Linear regression

We start with $\lambda = 0$, and gradually increase $\lambda$ to $+\infty$:
Ridge Linear regression

We start with $\lambda = 0$, and gradually increase $\lambda$ to $+\infty$: 
Outline of Today

1. Bias & Variance tradeoff demo on Ridge Linear Regression

2. Derivation of Bias / Variance for Ridge LR

2. Model selection in practice (Cross Validation)
Derivation of Bias and Variance for Ridge Linear regression

Denote $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}, Y = [y_1, \ldots, y_n]^T \in \mathbb{R}^n, \epsilon = [\epsilon_1, \ldots, \epsilon_n]^T \in \mathbb{R}^n$

Ridge LR in matrix / vector form:

$X = \begin{bmatrix} 1 & x_1 & \cdots & x_n \end{bmatrix}$

$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$

$\epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$

$Y_i = w^T x_i + \epsilon_i$
Derivation of Bias and Variance for Ridge Linear regression

Denote $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}$, $Y = [y_1, \ldots, y_n]^T \in \mathbb{R}^n$, $\epsilon = [\epsilon_1, \ldots, \epsilon_n]^T \in \mathbb{R}^n$

Ridge LR in matrix / vector form:

$$\hat{w} = \arg \min_w \|X^Tw - Y\|_2^2 + \lambda \|w\|_2^2$$
**Derivation of Bias and Variance for Ridge Linear regression**

Denote $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}$, $Y = [y_1, \ldots, y_n]^T \in \mathbb{R}^n$, $\epsilon = [\epsilon_1, \ldots, \epsilon_n]^T \in \mathbb{R}^n$

Ridge LR in matrix / vector form:

$$\hat{w} = \arg \min_w ||X^T w - Y||_2^2 + \lambda ||w||_2^2$$

Since $y_i = (w^*)^T x_i + \epsilon_i$ we have $Y = X^T w^* + \epsilon$

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} -x_1^T \\ \vdots \\ -x_n^T \end{bmatrix} w^* + \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_n \end{bmatrix}$$
The Expectation of the Ridge LR solution

Recall we have closed form solution for Ridge LR

\[ \hat{w} = (XX^T + \lambda I)^{-1} XY = (XX^T + \lambda I)^{-1}X(X^Tw^* + \epsilon) \]

\[ Y = X^Tw^* + \epsilon \]

\[ \hat{w} = w^* + \epsilon \]

\[ \overline{w} = E[\hat{w}] \]

\[ \overline{w} - \hat{w} \]
The Expectation of the Ridge LR solution

Recall we have closed form solution for Ridge LR

\[ \hat{w} = (XX^\top + \lambda I)^{-1}XY = (XX^\top + \lambda I)^{-1}X(X^\top w^* + \epsilon) \]

Source of the randomness of \( \hat{w} \)
The Expectation of the Ridge LR solution

Recall we have closed form solution for Ridge LR

\[ \hat{w} = (XX^T + \lambda I)^{-1}XY = (XX^T + \lambda I)^{-1}X(X^Tw^* + \epsilon) \]

Source of the randomness of \( w \)

Let us compute the average \( \bar{w} := \mathbb{E}_\epsilon[\hat{w}] \):
The Expectation of the Ridge LR solution

Recall we have closed form solution for Ridge LR

\[ \hat{w} = (XX^T + \lambda I)^{-1} XY = (XX^T + \lambda I)^{-1} X(X^T w^* + \epsilon) \]

Let us compute the average \( \bar{w} := \mathbb{E}_\epsilon[\hat{w}] : \)

\[ \mathbb{E}_\epsilon[\hat{w}] = (XX^T + \lambda I)^{-1} X[X^T w^* + \mathbb{E}_\epsilon[\epsilon]] \]

Source of the randomness of \( \hat{w} \)
The Expectation of the Ridge LR solution

Recall we have closed form solution for Ridge LR

\[ \hat{w} = (XX^\top + \lambda I)^{-1}XY = (XX^\top + \lambda I)^{-1}X(X^\top w^* + \epsilon) \]

Let us compute the average \( \tilde{w} := \mathbb{E}_\epsilon[\hat{w}] \):

\[ \mathbb{E}_\epsilon[\hat{w}] = (XX^\top + \lambda I)^{-1}X[X^\top w^* + \mathbb{E}_\epsilon[\epsilon]] \]

\[ = (XX^\top + \lambda I)^{-1}XX^\top w^* \]

Source of the randomness of \( \hat{w} \)

\[ \tilde{w} = \bar{w}, \quad \tilde{w} - w^* \]
The Expectation of the Ridge LR solution

Recall we have closed form solution for Ridge LR

\[ \hat{w} = (XX^T + \lambda I)^{-1}XY = (XX^T + \lambda I)^{-1}X(X^Tw^* + \epsilon) \]

Let us compute the average \( \bar{w} := \mathbb{E}_e[\hat{w}] \):

\[ \mathbb{E}_e[\hat{w}] = (XX^T + \lambda I)^{-1}X[X^Tw^* + \mathbb{E}_e[\epsilon]] \]

\[ = (XX^T + \lambda I)^{-1}XX^Tw^* \]

\[ = (XX^T + \lambda I)^{-1}(XX^T + \lambda I - \lambda I)w^* \]

Source of the randomness of \( \hat{w} \)
The Expectation of the Ridge LR solution

Recall we have closed form solution for Ridge LR

\[ \hat{w} = (XX^T + \lambda I)^{-1} X Y = (XX^T + \lambda I)^{-1} X (X^T w^* + \epsilon) \]

Let us compute the average \( \overline{w} := \mathbb{E}_\epsilon[\hat{w}] \):

\[
\mathbb{E}_\epsilon[\hat{w}] = (XX^T + \lambda I)^{-1} X [X^T w^* + \mathbb{E}_\epsilon[\epsilon]] \\
= (XX^T + \lambda I)^{-1} XX^T w^* \\
= (XX^T + \lambda I)^{-1} (XX^T + \lambda I - \lambda I) w^* = w^* - \lambda (XX^T + \lambda I)^{-1} w^*
\]
The Bias of Ridge Linear regression

\[ \tilde{w} = \mathbb{E}[\hat{w}] = w^* - \lambda(XX^T + \lambda)^{-1}w^* \]

Bias term: \( \sum_{i=1}^{n} \left( (\tilde{w} - w^*)^T x_i \right)^2 \)
The Bias of Ridge Linear regression

\[ \bar{w} = \mathbb{E}[\hat{w}] = w^* - \lambda (XX^\top + \lambda)^{-1}w^* \]

Bias term:

\[ \sum_{i=1}^{n} \left( (\bar{w} - w^*)^\top x_i \right)^2 \]

\[ = \sum_{i=1}^{n} \left( (\lambda (XX^\top + \lambda)^{-1}w^*)^\top x_i \right)^2 \]

\[ \Rightarrow \sum_{i=1}^{n} \left( \lambda (XX^\top + \lambda)^{-1}w^* \right)^\top x_i \cdot x_i^\top \left( \lambda (XX^\top + \lambda)^{-1}w^* \right) \]

\[ \Rightarrow (\lambda (XX^\top + \lambda)^{-1}w^*)^\top XX^\top (\lambda (XX^\top + \lambda)^{-1}w^*) \]
The Bias of Ridge Linear regression

\[ \bar{w} = \mathbb{E}[\hat{w}] = w^* - \lambda (XX^\top + \lambda)^{-1} w^* \]

Bias term:

\[ \sum_{i=1}^{n} \left( (\bar{w} - w^*)^\top x_i \right)^2 \]

\[ = \sum_{i=1}^{n} \left( (\lambda (XX^\top + \lambda)^{-1} w^*)^\top x_i \right)^2 \]

\[ = \lambda^2 (w^*)^\top (XX^\top + \lambda I)^{-1} XX^\top (XX^\top + \lambda I)^{-1} w^* \]
The Bias of Ridge Linear regression

\[
\text{Bias} = \lambda^2 (w^*)^T (XX^T + \lambda I)^{-1} XX^T (XX^T + \lambda I)^{-1} w^*
\]
The Bias of Ridge Linear regression

$$\text{Bias} = \lambda^2 (w^*)^T (XX^T + \lambda I)^{-1} XX^T (XX^T + \lambda I)^{-1} w^*$$

Eigendecomposition on $XX^T = U \Sigma U^T$
The Bias of Ridge Linear regression

Bias = $\lambda^2 (w^*)^T (XX^T + \lambda I)^{-1} XX^T (XX^T + \lambda I)^{-1} w^*$

Eigendecomposition on $XX^T = U \Sigma U^T$

\[
\begin{bmatrix}
\frac{\sigma_1}{(\sigma_1/\lambda + 1)^2} & 0 & 0 & \ldots \\
0 & \frac{\sigma_2}{(\sigma_2/\lambda + 1)^2} & 0 & \ldots \\
\vdots & \vdots & \ddots & \vdots \\
0, & \ldots & & \frac{\sigma_d}{(\sigma_d/\lambda + 1)^2}
\end{bmatrix}
\]
The Bias of Ridge Linear regression

$$\text{Bias} = \lambda^2(w^*)^\top (XX^\top + \lambda I)^{-1} XX^\top (XX^\top + \lambda I)^{-1} w^*$$

Eigendecomposition on $XX^\top = U\Sigma U^\top$

$$= (w^*)^\top U \begin{bmatrix}
\frac{\sigma_1}{(\sigma_1/\lambda + 1)^2} & 0 & 0 & \ldots \\
0 & \frac{\sigma_2}{(\sigma_2/\lambda + 1)^2} & 0 & \ldots \\
\vdots & \vdots & \ddots & \vdots \\
0, & \ldots & \frac{\sigma_d}{(\sigma_d/\lambda + 1)^2}
\end{bmatrix} U^\top w^*$$

Q: how does bias behave when $\lambda \to +\infty$

$$\Rightarrow w^T U \Sigma U^T w^*$$

$$= w^T X X^T X^T w^*$$

$$= \sum_{i=1}^n (w^T x_i)^2$$
The Bias of Ridge Linear Regression

\[
\text{Bias} = \lambda^2 (w^*)^T (XX^T + \lambda I)^{-1} XX^T (XX^T + \lambda I)^{-1} w^*
\]

Eigendecomposition on \( XX^T = U \Sigma U^T \)

Q: how does bias behave when \( \lambda \to +\infty \)

Q: how does bias behave when \( \lambda \to 0 \)

\[
(w^*)^T U \left[ \begin{array}{cccc}
\sigma_1 & 0 & 0 & \ldots \\
0 & \sigma_2 & 0 & \ldots \\
\ldots & \ldots & \ddots & \ldots \\
0 & \ldots & 0 & \sigma_d \\
\end{array} \right] U^T w^* = 0
\]
The Variance of Ridge Linear regression

\[ \bar{w} = \mathbb{E}[\hat{w}] = (XX^T + \lambda I)^{-1}XX^T \bar{w}^* \]
The Variance of Ridge Linear regression

\[ \bar{w} = \mathbb{E}[\hat{w}] = (XX^T + \lambda I)^{-1}XX^T \hat{w}^* \]

Variance term:

\[ \sum_{i=1}^{n} \mathbb{E}(\hat{w}^T x_i - \bar{w}^T x_i)^2 \]
The Variance of Ridge Linear regression

\[ \tilde{w} = \mathbb{E}[\hat{w}] = (XX^T + \lambda I)^{-1}XX^Tw^* \]

Variance term:

\[ \sum_{i=1}^{n} \mathbb{E}(\hat{w}^Tx_i - \tilde{w}^Tx_i)^2 \]

\[ = \sum_{i=1}^{d} \sigma_i^2/(\sigma_i + \lambda)^2 \]
The Variance of Ridge Linear regression

$$\tilde{w} = \mathbb{E}[\hat{w}] = (XX^T + \lambda I)^{-1}XX^T w^*$$

Variance term: 

$$\sum_{i=1}^{n} \mathbb{E}(\hat{w}^T x_i - \tilde{w}^T x_i)^2$$

$$= \sum_{i=1}^{d} \sigma_i^2 / (\sigma_i + \lambda)^2$$

(Optional — tedious but basic computation, see note)
The Variance of Ridge Linear regression

\[ \tilde{w} = \mathbb{E}[\hat{w}] = (XX^T + \lambda I)^{-1}XX^T w^* \]

Variance term:

\[ \sum_{i=1}^{n} \mathbb{E}(\hat{w}^T x_i - \tilde{w}^T x_i)^2 \]

\[ = \sum_{i=1}^{d} \frac{\sigma_i^2}{(\sigma_i + \lambda)^2} \]

(Optional — tedious but basic computation, see note)

Q: how does Var behave when \( \lambda \to +\infty \)

Q: how does Var behave when \( \lambda \to 0 \)
Ridge Linear regression

Tuning $\lambda$ allows us to control the generalization error of Ridge LR solution:

$$\mathbb{E}((\hat{w}^T x - y)^2) = \text{Variance} + \text{Bias} + \text{Inherent noise}$$
Ridge Linear regression

Tuning $\lambda$ allows us to control the generalization error of Ridge LR solution:

$$\mathbb{E}(\hat{w}^T x - y)^2 = \text{Variance} + \text{Bias} + \text{Inherent noise}$$
Outline of Today

1. Bias & Variance tradeoff demo on Ridge Linear Regression

2. Derivation of Bias / Variance for Ridge LR

2. Model selection in practice (Cross Validation)
How to select the best model from data

Examples:

1. Select the right order of polynomials for regression
How to select the best model from data

Examples:

1. Select the right order of polynomials for regression

2. Select the right ridge regularization weight $\lambda$
How to select the best model from data

Examples:

1. Select the right order of polynomials for regression

2. Select the right ridge regularization weight $\lambda$

3. Select the right penalty for slack variables in soft SVM (i.e., the C parameter)
How to select the best model from data

Examples:

1. Select the right order of polynomials for regression

2. Select the right ridge regularization weight $\lambda$

3. Select the right penalty for slack variables in soft SVM (i.e., the C parameter)
Select the right $\lambda$ for Ridge LR

Cross Validation:

Random shuffle data, split the data into K folds

For $i = 1$ to $K$:
Select the right $\lambda$ for Ridge LR

Cross Validation:

Random shuffle data, split the data into K folds

For $i = 1$ to $K$:

$\hat{\lambda}_i = \text{Ridge LR}(D_{-i}, \lambda),$
Select the right $\lambda$ for Ridge LR

Cross Validation:

Random shuffle data, split the data into K folds

For $i = 1$ to $K$:

$$\hat{w}_i = \text{Ridge LR}(\mathcal{D}_{-i}, \lambda),$$

$$\varepsilon_{\text{vad},i} = \sum_{x,y \in \mathcal{D}_i} (\hat{w}_i^T x - y)^2 / |\mathcal{D}_i|$$
Select the right $\lambda$ for Ridge LR

Cross Validation:

Random shuffle data, split the data into $K$ folds

For $i = 1$ to $K$:

\[ \hat{w}_i = \text{Ridge LR}(\mathcal{D}_{-i}, \lambda), \]

\[ e_{\text{vad};i} = \sum_{x,y \in \mathcal{D}_i} (\hat{w}_i^T x - y)^2 / |\mathcal{D}_i| \]

Output avg val-err over $K$ folds: \( \bar{e}_{\lambda} = \frac{1}{K} \sum_{i=1}^{K} e_{\text{vad};i} / K \)
Select the right $\lambda$ for Ridge LR

Cross Validation:

Random shuffle data, split the data into $K$ folds

For $i = 1$ to $K$:

\[
\hat{w}_i = \text{Ridge LR}(\mathcal{D}_{\neg i}, \lambda),
\]

\[
\epsilon_{vad,i} = \sum_{x,y \in \mathcal{D}_i} (\hat{w}_i^T x - y)^2 / \mathcal{D}_i
\]

Output avg val-err over $K$ folds: $\bar{\epsilon}_\lambda = \sum_{i=1}^{K} \epsilon_{vad,i} / K$

$\approx \mathbb{E}_{x,y \sim p} (\hat{w}_i^T x - y)^2$, i.e., test error of $\hat{w}_i$
Select the right $\lambda$ for Ridge LR

Cross Validation:

Random shuffle data, split the data into $K$ folds

For $i = 1$ to $K$:

$$\hat{w}_i = \text{Ridge LR}(\mathcal{D}_{-i}, \lambda),$$

$$e_{\text{vad};i} = \sum_{x,y \in \mathcal{D}_i} (\hat{w}_i^T x - y)^2 / \mathcal{D}_i$$

Output avg val-err over $K$ folds:

$$\bar{e}_\lambda = \frac{1}{K} \sum_{i=1}^{K} e_{\text{vad};i}$$

$$\approx \mathbb{E}_{x,y \sim P(\hat{w}_i^T x - y)^2}, \text{ i.e., test error of } \hat{w}_i$$

$$\approx \mathbb{E}_{\mathcal{D}} \left[ \mathbb{E}_{x,y \sim P(\hat{w}_{\mathcal{D}}^T x - y)^2} \right], \text{ i.e., Generalization error of Ridge LR w/ } \lambda$$
Select the right $\lambda$ for Ridge LR

By numerating a set of possible $\lambda \in \mathbb{R}^+$, we select the one that has the smallest Cross-Valid error:
Select the right $\lambda$ for Ridge LR

By numerating a set of possible $\lambda \in \mathbb{R}^+$, we select the one that has the smallest Cross-Valid error:

For $\lambda$ in $[1e-5, 1e-4, \ldots, 1e4, 1e5]$:
Select the right $\lambda$ for Ridge LR

By numerating a set of possible $\lambda \in \mathbb{R}^+$, we select the one that has the smallest Cross-Valid error:

For $\lambda$ in $[1e-5, 1e-4, \ldots, 1e4, 1e5]$:

Split the data into $K$ folds

For $i = 1$ to $K$:

\[
\hat{w}_i = \text{Ridge LR}(\mathcal{D}_{-i}, \lambda),
\]

\[
\epsilon_{\text{vad};i} = \sum_{x, y \in \mathcal{D}_i} (\hat{w}_i^T x - y)^2 / \mathcal{D}_i
\]

Output avg val-err over K folds: \( \bar{\epsilon}_\lambda = \sum_{i=1}^{K} \epsilon_{\text{vad};i} / K \)
Select the right $\lambda$ for Ridge LR

By numerating a set of possible $\lambda \in \mathbb{R}^+$, we select the one that has the smallest Cross-Valid error:

For $\lambda$ in $[1e-5, 1e-4, \ldots, 1e4, 1e5]$:

Split the data into $K$ folds

For $i = 1$ to $K$:

\[
\hat{w}_i = \text{Ridge LR}(\mathcal{D}_{-i}, \lambda),
\]

\[
\epsilon_{vad;i} = \sum_{x,y \in \mathcal{D}_i} (\hat{w}_i^T x - y)^2 / \mathcal{D}_i
\]

Output avg val-err over $K$ folds:

\[
\bar{\epsilon}_\lambda = \frac{1}{K} \sum_{i=1}^{K} \epsilon_{vad;i}
\]

Select $\lambda^* = \arg \min_{\lambda} \bar{\epsilon}_\lambda$
Select the right $\lambda$ for Ridge LR

By numerating a set of possible $\lambda \in \mathbb{R}^+$, we select the one that has the smallest Cross-Valid error:

For $\lambda$ in $[1e-5, 1e-4, \ldots 1e4, 1e5]$:

Split the data into $K$ folds

For $i = 1$ to $K$:

- $\hat{w}_i = \text{Ridge LR}(\mathcal{D}_{-i}, \lambda)$,
- $\epsilon_{vad;i} = \sum_{x,y \in \mathcal{D}_i} (\hat{w}_i^T x - y)^2 / \mathcal{D}_i$

Output avg val-err over $K$ folds: $\bar{\epsilon}_\lambda = \frac{1}{K} \sum_{i=1}^{K} \epsilon_{vad;i}$

Select $\lambda^* = \arg \min_{\lambda} \bar{\epsilon}_\lambda$
Select the right $\lambda$ for Ridge LR

By numerating a set of possible $\lambda \in \mathbb{R}^+$, we select the one that has the smallest Cross-Valid error:

For $\lambda$ in $[1e-5, 1e-4, \ldots 1e4, 1e5]$:

Split the data into $K$ folds
For $i = 1$ to $K$:

$\hat{w}_i = \text{Ridge LR}(\mathcal{D}_{-i}, \lambda)$,

$\epsilon_{vad; i} = \sum_{x,y \in \mathcal{D}_i} (\hat{w}_i^T x - y)^2 / \mathcal{D}_i$

Output avg val-err over $K$ folds: $\bar{\epsilon}_\lambda = \sum_{i=1}^K \epsilon_{vad; i} / K$

$\lambda^* = \arg \min_\lambda \bar{\epsilon}_\lambda$
Practical Suggestions for combating over/under fitting

Model capacity

Test error

Train error
Practical Suggestions for combating over/under fitting

R1: Underfitting (both train and test errors are large)
Practical Suggestions for combating over/under fitting

R1: Underfitting (both train and test errs are large)

Suggestions:
1. Increase complexity of models
Practical Suggestions for combating over/under fitting

R1: Underfitting (both train and test errs are large)

Suggestions:
1. Increase complexity of models
2. More features
Practical Suggestions for combating over/under fitting

R1: Underfitting (both train and test errs are large)

Suggestions:
1. Increase complexity of models
2. More features
3. Using Boosting (we will see it later)
Practical Suggestions for combating over/under fitting

R1: Underfitting (both train and test errs are large)

Suggestions:
1. Increase complexity of models
2. More features
3. Using Boosting (we will see it later)

R2: overfitting (small train err but large test err)
Practical Suggestions for combating over/under fitting

R1: Underfitting (both train and test errs are large)

Suggestions:
1. Increase complexity of models
2. More features
3. Using Boosting (we will see it later)

R2: overfitting (small train err but large test err)

Suggestions:
1. More train data
Practical Suggestions for combating over/under fitting

R1: Underfitting (both train and test errs are large)

Suggestions:
1. Increase complexity of models
2. More features
3. Using Boosting (we will see it later)

R2: overfitting (small train err but large test err)

Suggestions:
1. More train data
2. Reduce model capacity
Practical Suggestions for combating over/under fitting

R1: Underfitting (both train and test errs are large)

Suggestions:
1. Increase complexity of models
2. More features
3. Using Boosting (we will see it later)

R2: overfitting (small train err but large test err)

Suggestions:
1. More train data
2. Reduce model capacity
3. Using Bagging (we will see it later)