# Bias-Variance Tradeoff
# & Model Selection

# Announcements

HW5 and P5 are coming out

# Recap on Bias-Variance Tradeoff

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

# Recap on Bias-Variance Tradeoff

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

What we have shown is the Bias-Variance decomposition:

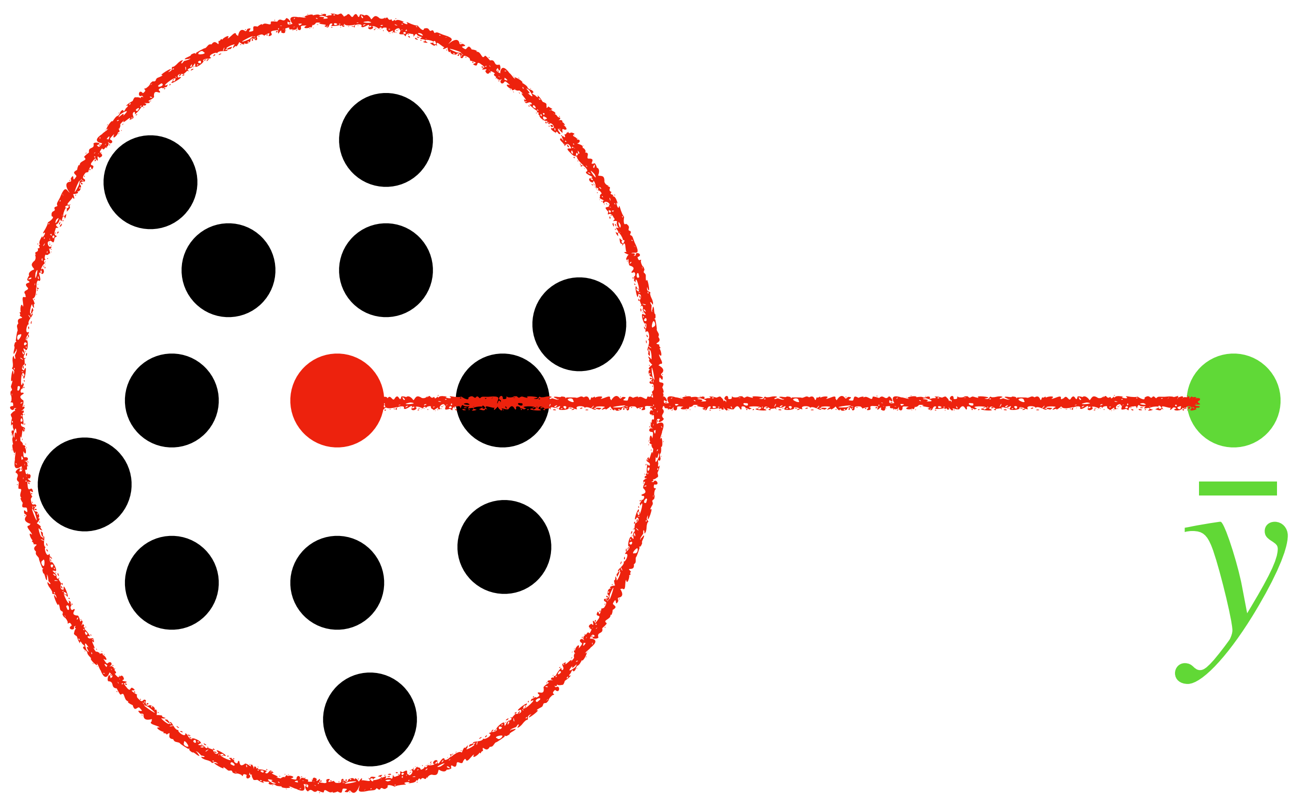
$$\mathbb{E}_{\mathscr{D},x,y}(h_{\mathscr{D}}(x) - y)^2 = \mathbb{E}_{\mathscr{D},x}(h_{\mathscr{D}}(x) - \bar{h}(x))^2 + \mathbb{E}_x(\bar{h}(x) - \bar{y}(x))^2 + \mathbb{E}_{x,y}(\bar{y}(x) - y)^2$$

# Recap on Bias-Variance Tradeoff

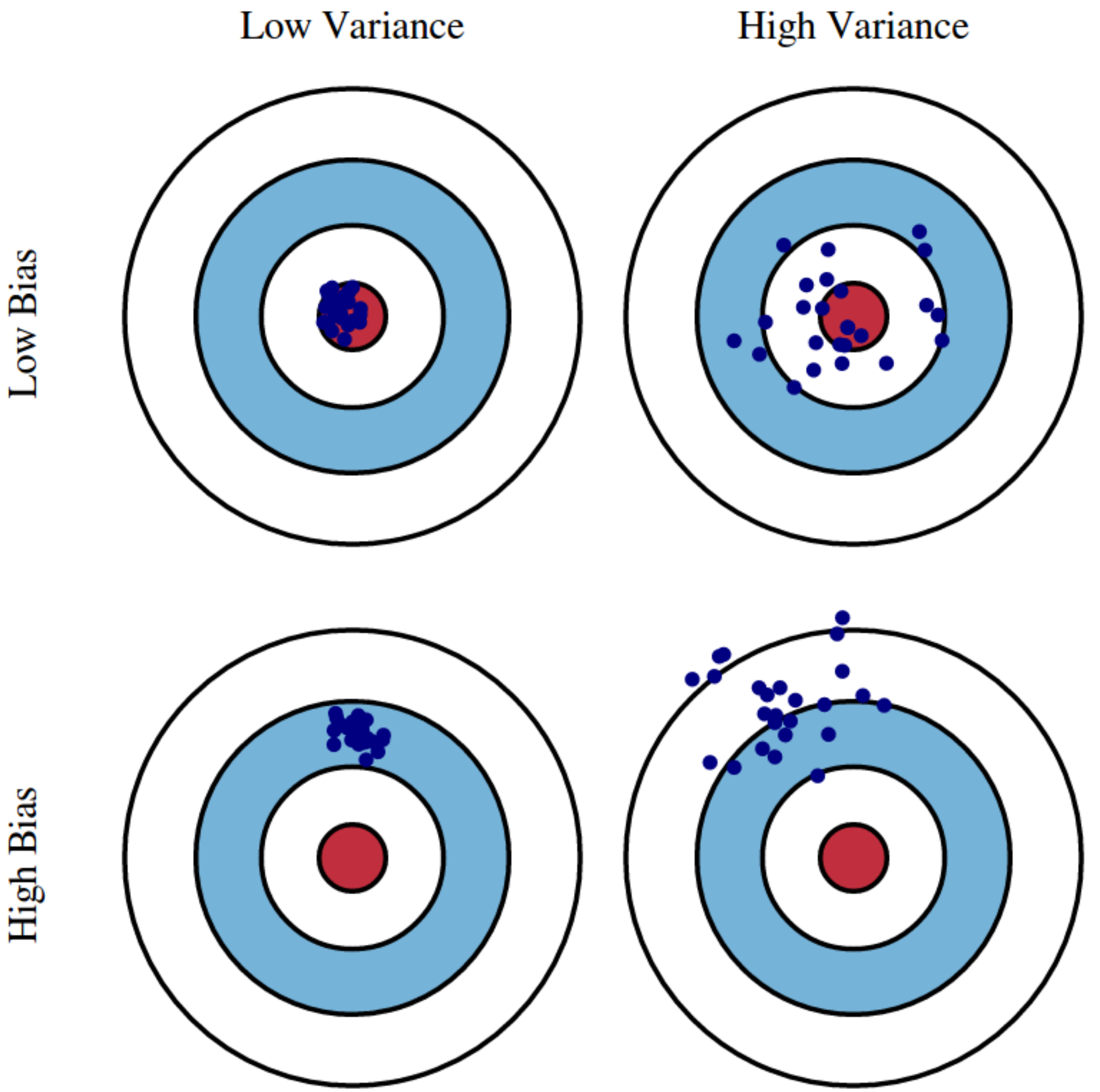
Denote $h_{\mathscr{D}}$ as the ERM solution on dataset $\mathscr{D}$ w/ squared loss $\ell(h, x, y) = (h(x) - y)^2$

What we have shown is the Bias-Variance decomposition:

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



# Recap on Bias-Variance Tradeoff



# 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 (re-visit 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 $y_i \sim (w^\star)^\top x_i + \epsilon_i$, $\epsilon_i \sim \mathcal{N}(0,1)$

# 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 $y_i \sim (w^\star)^\top x_i + \epsilon_i, \; \epsilon_i \sim \mathcal{N}(0,1)$

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

(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

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

**What we will show now:**

Larger $\lambda$ (model becomes "simpler") => larger bias, but smaller variance

# Ridge Linear regression

Ridge Linear Regression formulation

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

**What we will show now:**

Larger $\lambda$ (model becomes "simpler") => 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: $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_{5000}$,

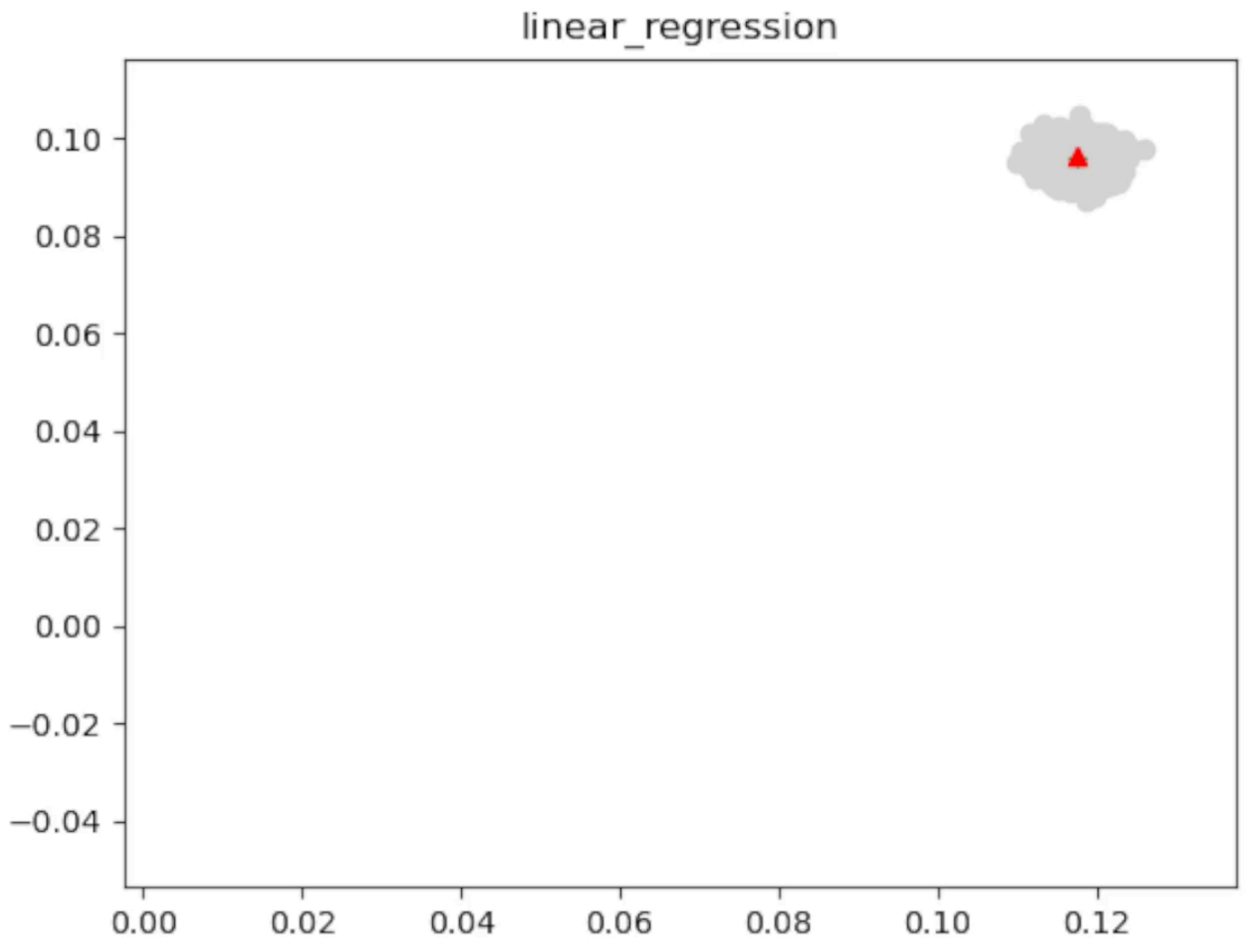
2. For a given $\lambda$, solve Ridge LR for each dataset, get $\hat{w}_1, \ldots, \hat{w}_{5000}$

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 $w^\star$

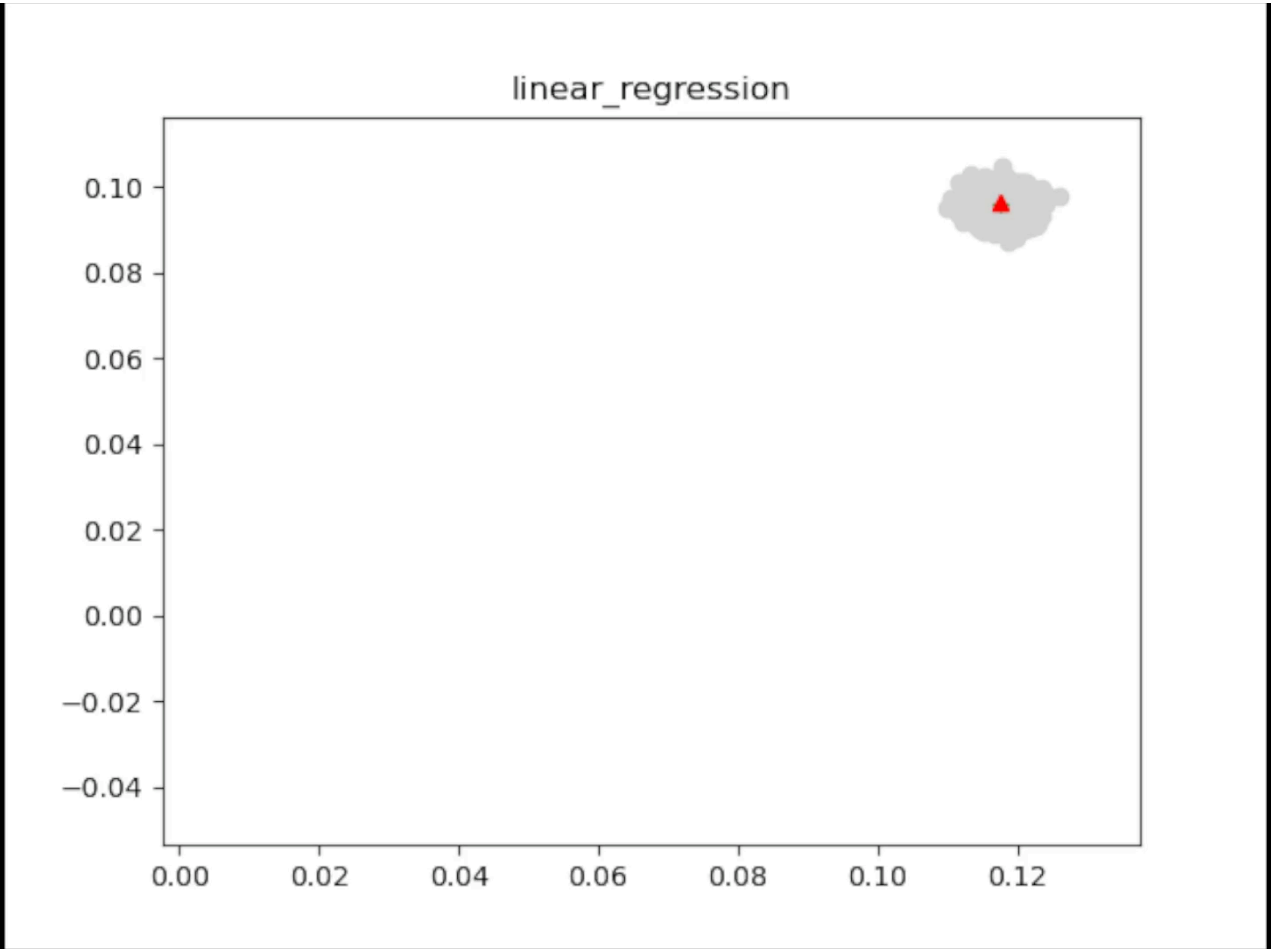
# Ridge Linear regression

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 (re-visit Cross Validation)

# Derivation of Bias and Variance for Ridge Linear regression

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

Ridge LR in matrix / vector form:

# Derivation of Bias and Variance for Ridge Linear regression

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

Ridge LR in matrix / vector form:

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

# Derivation of Bias and Variance for Ridge Linear regression

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

Ridge LR in matrix / vector form:

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

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

# The Expectation of the Ridge LR solution

Recall we have closed form solution for Ridge LR

$$\hat{w} = (XX^\top + \lambda I)^{-1} XY \; = (XX^\top + \lambda I)^{-1} X(X^\top w^\star + \epsilon)$$

# The Expectation of the Ridge LR solution

Recall we have closed form solution for Ridge LR

$$\hat{w} = (XX^\top + \lambda I)^{-1} XY = (XX^\top + \lambda I)^{-1} X(X^\top w^\star + \epsilon)$$

Source of the randomness of $\hat{w}$

# The Expectation of the Ridge LR solution

Recall we have closed form solution for Ridge LR

$$\hat{w} = (XX^\top + \lambda I)^{-1} XY = (XX^\top + \lambda I)^{-1} X(X^\top w^\star + \epsilon)$$

Source of the randomness of $\hat{w}$

**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

$$\hat{w} = (XX^\top + \lambda I)^{-1} XY = (XX^\top + \lambda I)^{-1} X(X^\top w^\star + \epsilon)$$

Source of the randomness of $\hat{w}$

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

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

# The Expectation of the Ridge LR solution

Recall we have closed form solution for Ridge LR

$$\hat{w} = (XX^\top + \lambda I)^{-1} XY = (XX^\top + \lambda I)^{-1} X(X^\top w^\star + \epsilon)$$

Source of the randomness of $\hat{w}$

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

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

$$= (XX^\top + \lambda I)^{-1} XX^\top w^\star$$

# The Expectation of the Ridge LR solution

Recall we have closed form solution for Ridge LR

$$\hat{w} = (XX^\top + \lambda I)^{-1}XY = (XX^\top + \lambda I)^{-1}X(X^\top w^\star + \epsilon)$$

Source of the randomness of $\hat{w}$

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

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

$$= (XX^\top + \lambda I)^{-1}XX^\top w^\star$$

$$= (XX^\top + \lambda I)^{-1}(XX^\top + \lambda I - \lambda I)w^\star$$

# The Expectation of the Ridge LR solution

Recall we have closed form solution for Ridge LR

$$\hat{w} = (XX^\top + \lambda I)^{-1} XY \; = (XX^\top + \lambda I)^{-1} X(X^\top w^\star + \epsilon)$$

Source of the randomness of $\hat{w}$

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

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

$$= (XX^\top + \lambda I)^{-1} XX^\top w^\star$$

$$= (XX^\top + \lambda I)^{-1}(XX^\top + \lambda I - \lambda I) w^\star \; = w^\star - \lambda (XX^\top + \lambda I)^{-1} w^\star$$

# The Bias of Ridge Linear regression

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

Bias term: $\sum_{i=1}^{n} \left( (\bar{w} - w^\star)^\top x_i \right)^2$

# The Bias of Ridge Linear regression

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

Bias term: $\sum_{i=1}^{n} \left((\bar{w} - w^\star)^\top x_i\right)^2$

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

# The Bias of Ridge Linear regression

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

Bias term: $\sum_{i=1}^{n} \left((\bar{w} - w^\star)^\top x_i\right)^2$

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

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

# The Bias of Ridge Linear regression

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

# The Bias of Ridge Linear regression

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

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

# The Bias of Ridge Linear regression

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

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

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

# The Bias of Ridge Linear regression

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

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

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

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

# The Bias of Ridge Linear regression

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

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

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

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

Q: how does bias behave when $\lambda \rightarrow 0$

# The Variance of Ridge Linear regression

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

# The Variance of Ridge Linear regression

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

Variance term: $\sum_{i=1}^{n} \mathbb{E}(\hat{w}^\top x_i - \bar{w}^\top x_i)^2$

# The Variance of Ridge Linear regression

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

Variance term: $\sum_{i=1}^{n} \mathbb{E}(\hat{w}^\top x_i - \bar{w}^\top x_i)^2$

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

# The Variance of Ridge Linear regression

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

Variance term: $\sum_{i=1}^{n} \mathbb{E}(\hat{w}^\top x_i - \bar{w}^\top x_i)^2$

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

(Optional — tedious but basic computation, see note)

# The Variance of Ridge Linear regression

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

Variance term: $\sum_{i=1}^{n} \mathbb{E}(\hat{w}^\top x_i - \bar{w}^\top x_i)^2$

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

(Optional — tedious but basic computation, see note)

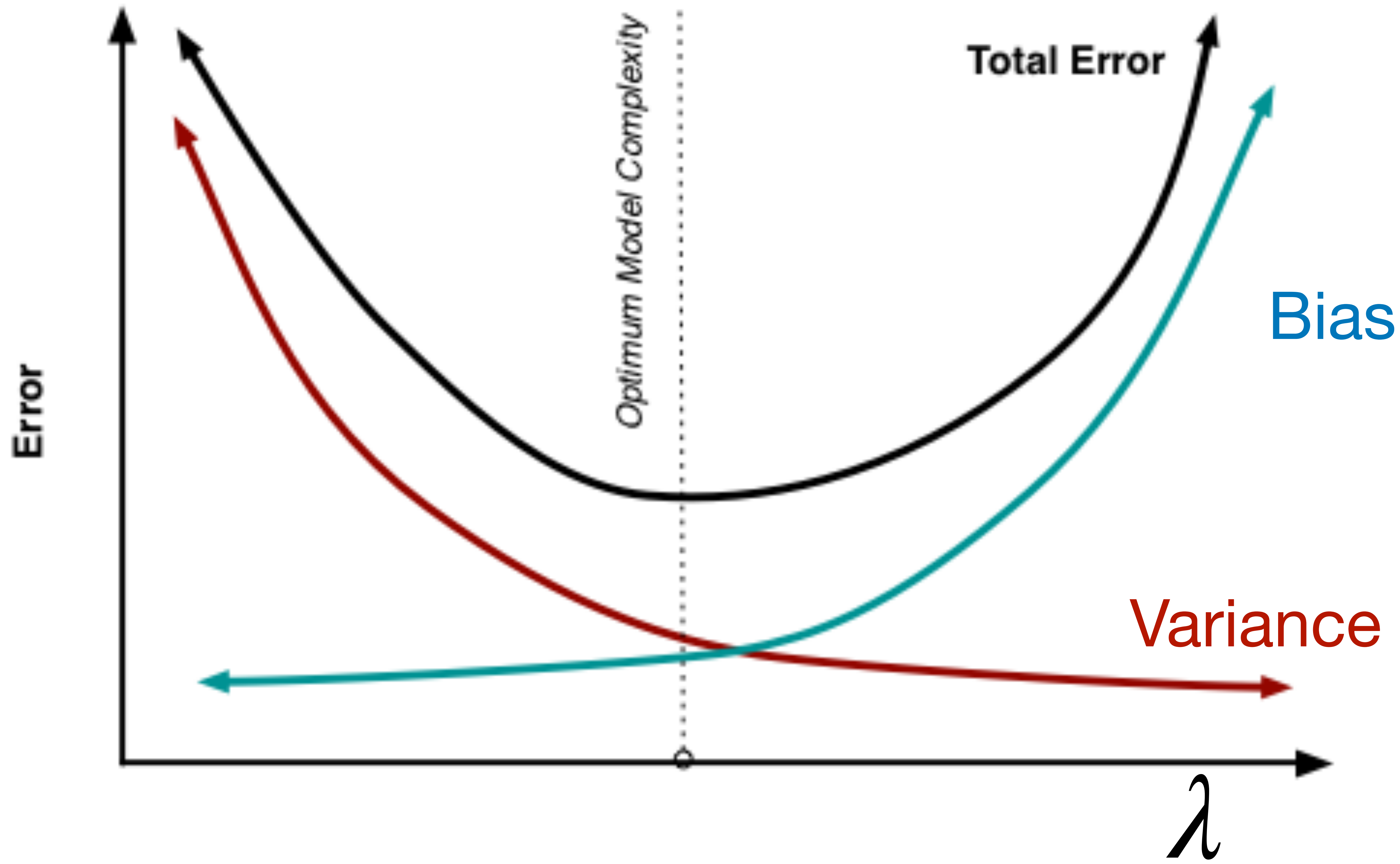
Q: how does Var behave when $\lambda \rightarrow +\infty$

Q: how does Var behave when $\lambda \rightarrow 0$

# Ridge Linear regression

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

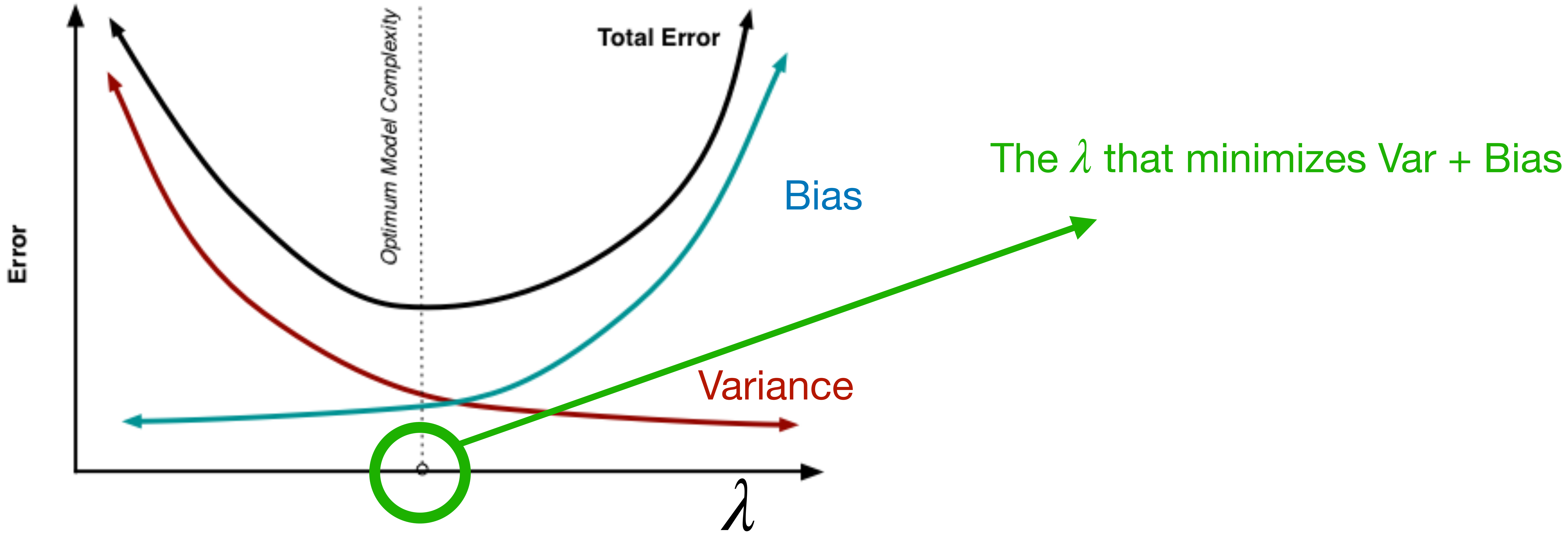
$$\mathbb{E}(\hat{w}^\top x - y)^2 = \text{Variance} + \text{Bias} + \text{Inherent noise}$$



# Ridge Linear regression

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

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



# 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 (re-visit 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 revisit:

Split the data into K folds

For i = 1 to K:

# Select the right $\lambda$ for Ridge LR

Cross Validation revisit:

Split the data into K folds

For i = 1 to K:

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

# Select the right $\lambda$ for Ridge LR

Cross Validation revisit:

Split the data into K folds

For i = 1 to K:

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

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

# Select the right $\lambda$ for Ridge LR

Cross Validation revisit:

Split the data into K folds

For i = 1 to K:

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

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

Output avg validation error $\bar{\epsilon}_\lambda = \sum_{i=1}^{K} \epsilon_{vad;i} / K$

# Select the right $\lambda$ for Ridge LR

Cross Validation revisit:

Split the data into K folds

For i = 1 to K:

$\hat{w}_k =$ Ridge LR$(\mathscr{D}_{-i}, \lambda)$,

$$\epsilon_{vad;k} = \sum_{x,y \in \mathscr{D}_i} (\hat{w}_i^\top x - y)^2 / |\mathscr{D}_i|$$

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

Output avg validation error $\bar{\epsilon}_\lambda = \sum_{i=1}^{K} \epsilon_{vad;i} / K$

# Select the right $\lambda$ for Ridge LR

Cross Validation revisit:

Split the data into K folds

For i = 1 to K:

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

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

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

Output avg validation error $\bar{\epsilon}_\lambda = \sum_{i=1}^{K} \epsilon_{vad;i}/K$

$\approx \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{x,y\sim P}(\hat{w}_{\mathcal{D}}^\top x - y)^2\right]$, 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, … 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, … 1e4,1e5]:

Split the data into K folds

For i = 1 to K:

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

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

Output avg validation error $\bar{\epsilon}_\lambda = \sum_{i=1}^{K} \epsilon_{vad;i}/K$

# 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, … 1e4,1e5]:

Split the data into K folds

For i = 1 to K:

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

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

Output avg validation error $\bar{\epsilon}_\lambda = \sum_{i=1}^{K} \epsilon_{vad;i}/K$

Select $\lambda^\star = \arg\min_\lambda \bar{\epsilon}_\lambda$

# 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, … 1e4,1e5]:

Split the data into K folds

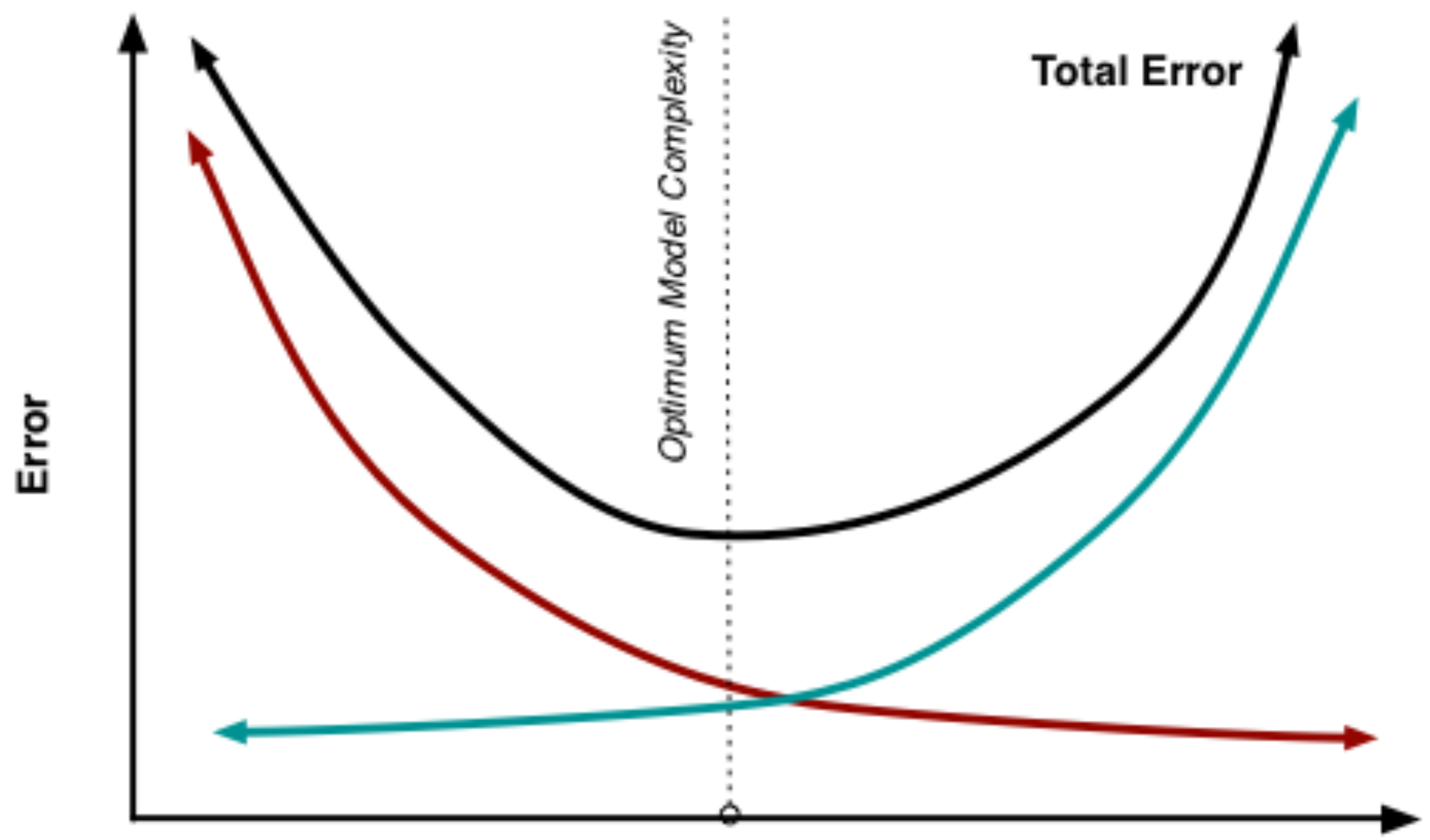
For i = 1 to K:

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

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

Output avg validation error $\bar{\epsilon}_\lambda = \sum_{i=1}^{K} \epsilon_{vad;i} / K$

Select $\lambda^\star = \arg\min_\lambda \bar{\epsilon}_\lambda$



# 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, … 1e4,1e5]:

Split the data into K folds

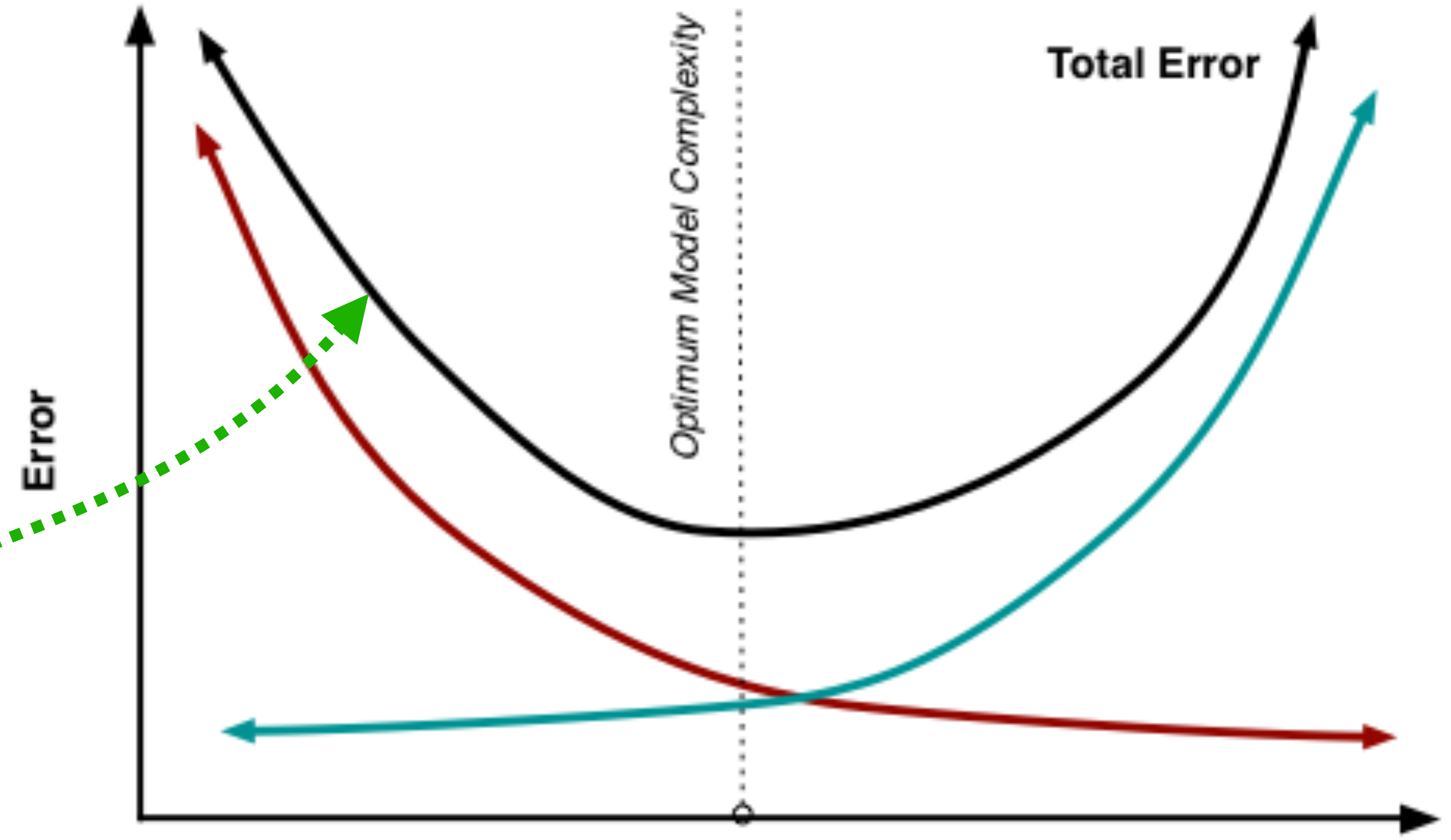
For i = 1 to K:

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

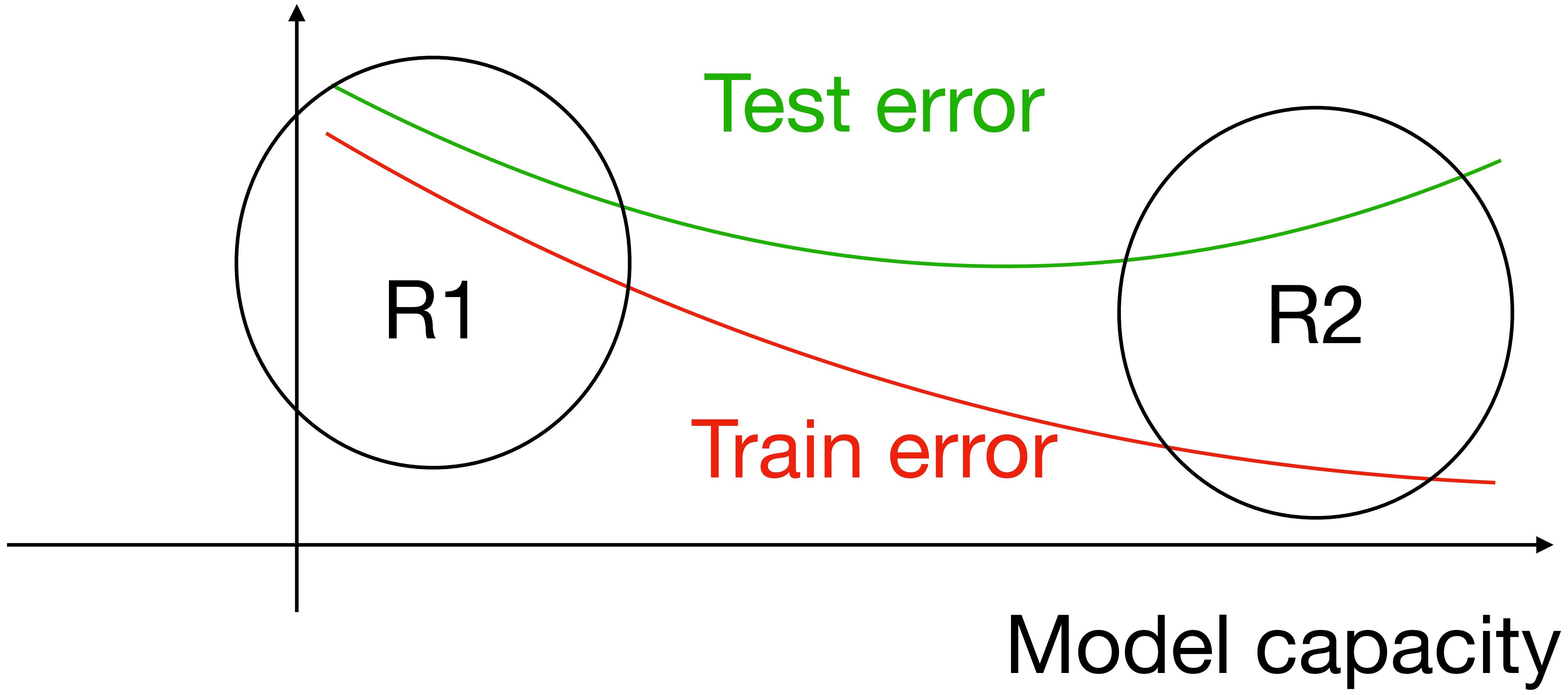
$$\epsilon_{vad;k} = \sum_{x,y \in \mathcal{D}_i} (\hat{w}_i^\top x - y)^2 / |\mathcal{D}_i|$$

Output avg validation error $\bar{\epsilon}_\lambda = \sum_{i=1}^{K} \epsilon_{vad;i} / K$

Select $\lambda^\star = \arg\min_\lambda \bar{\epsilon}_\lambda$

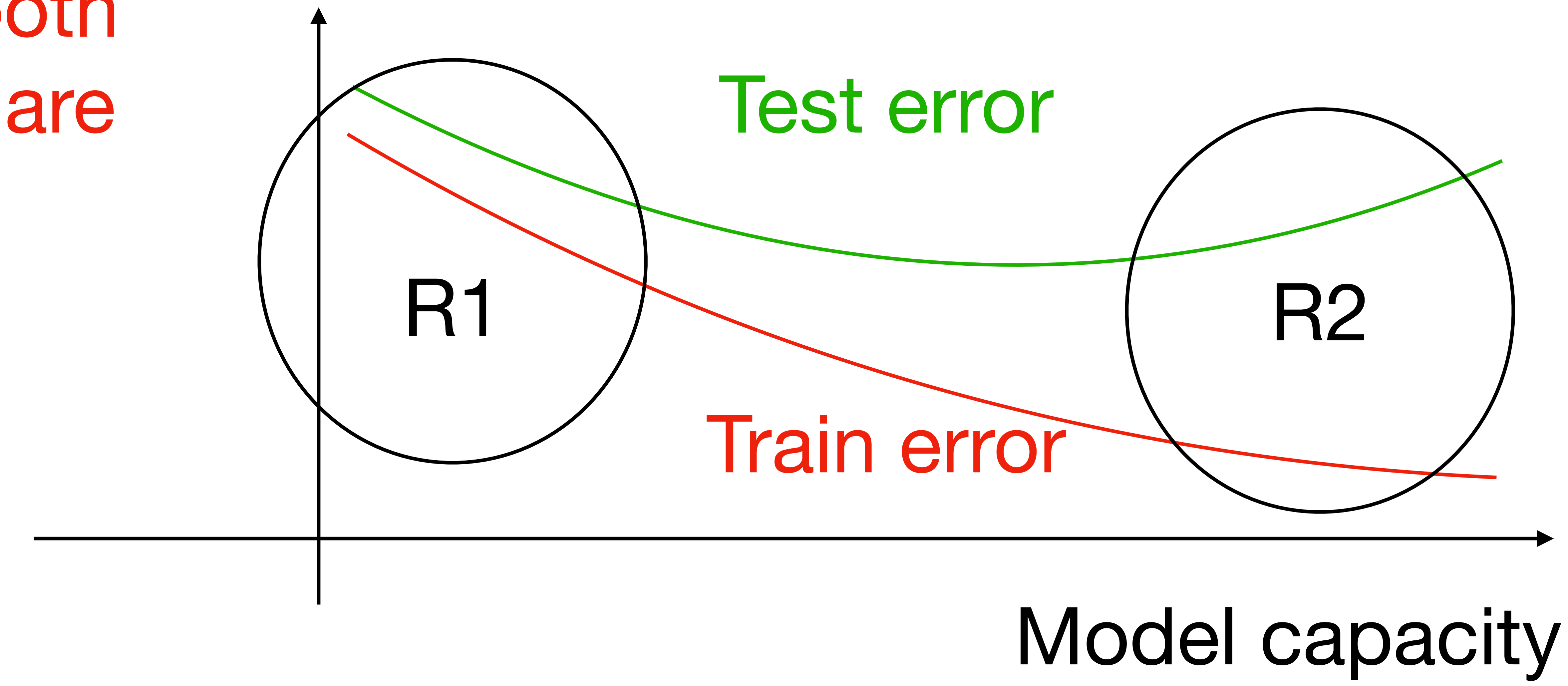


# Practical Suggestions for combating over/under fitting



# Practical Suggestions for combating over/under fitting

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

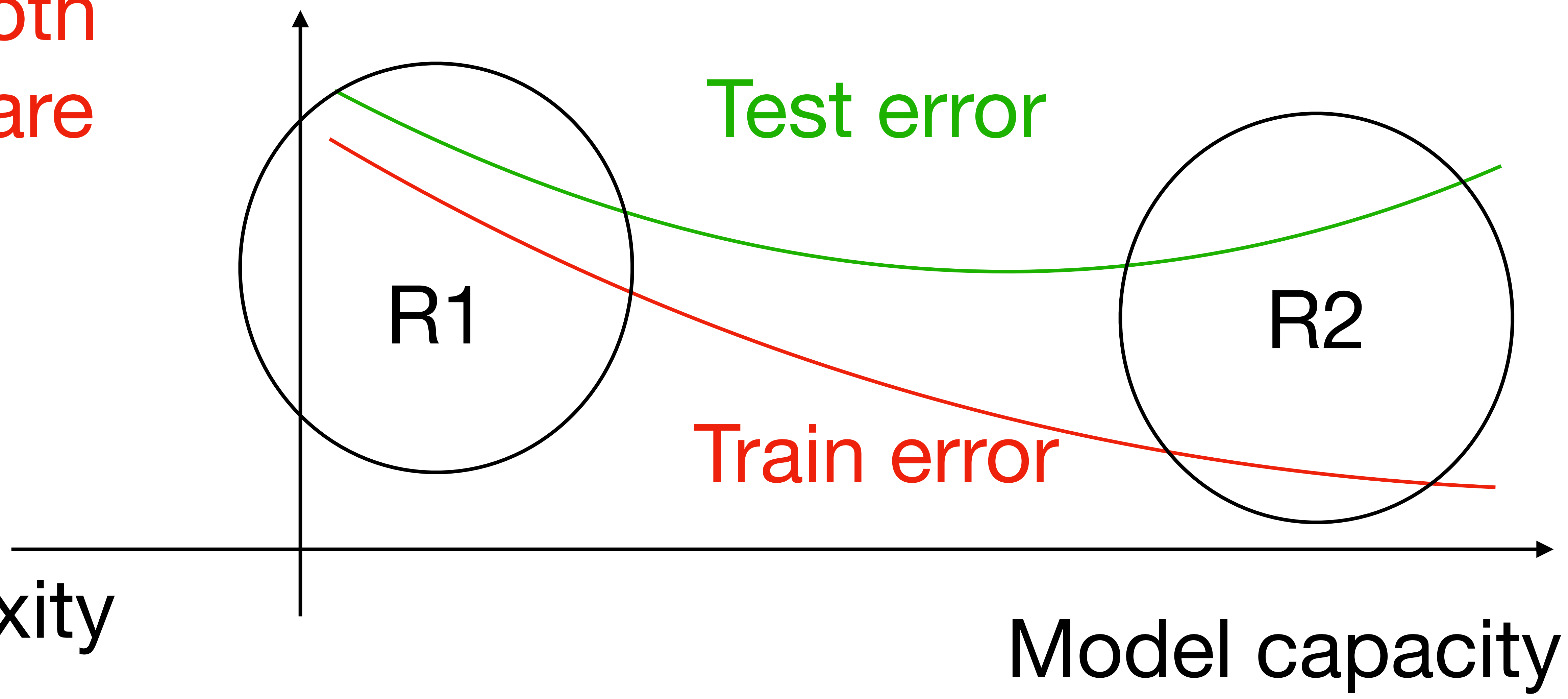


# Practical Suggestions for combating over/under fitting

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

Suggestions:

1. Increase complexity of models



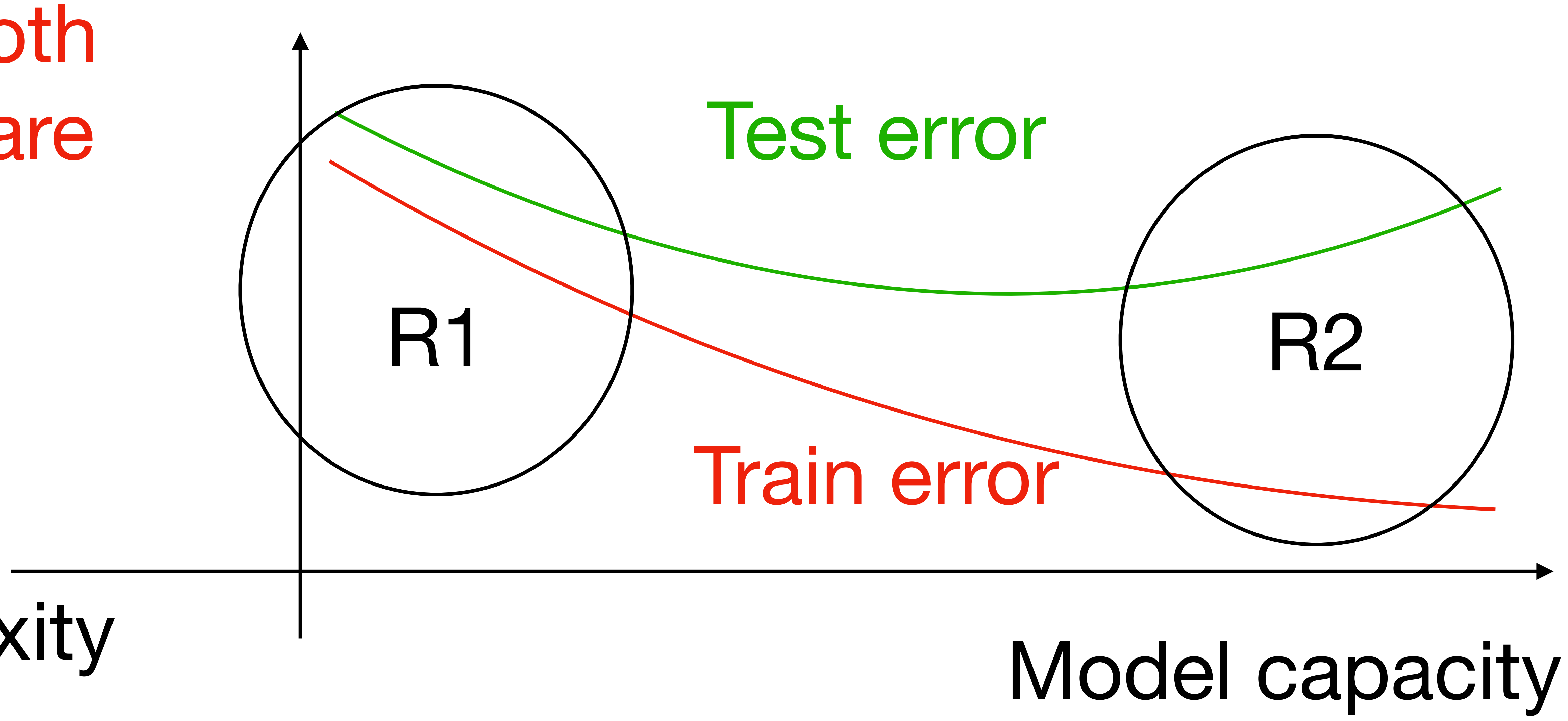
# Practical Suggestions for combating over/under fitting

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

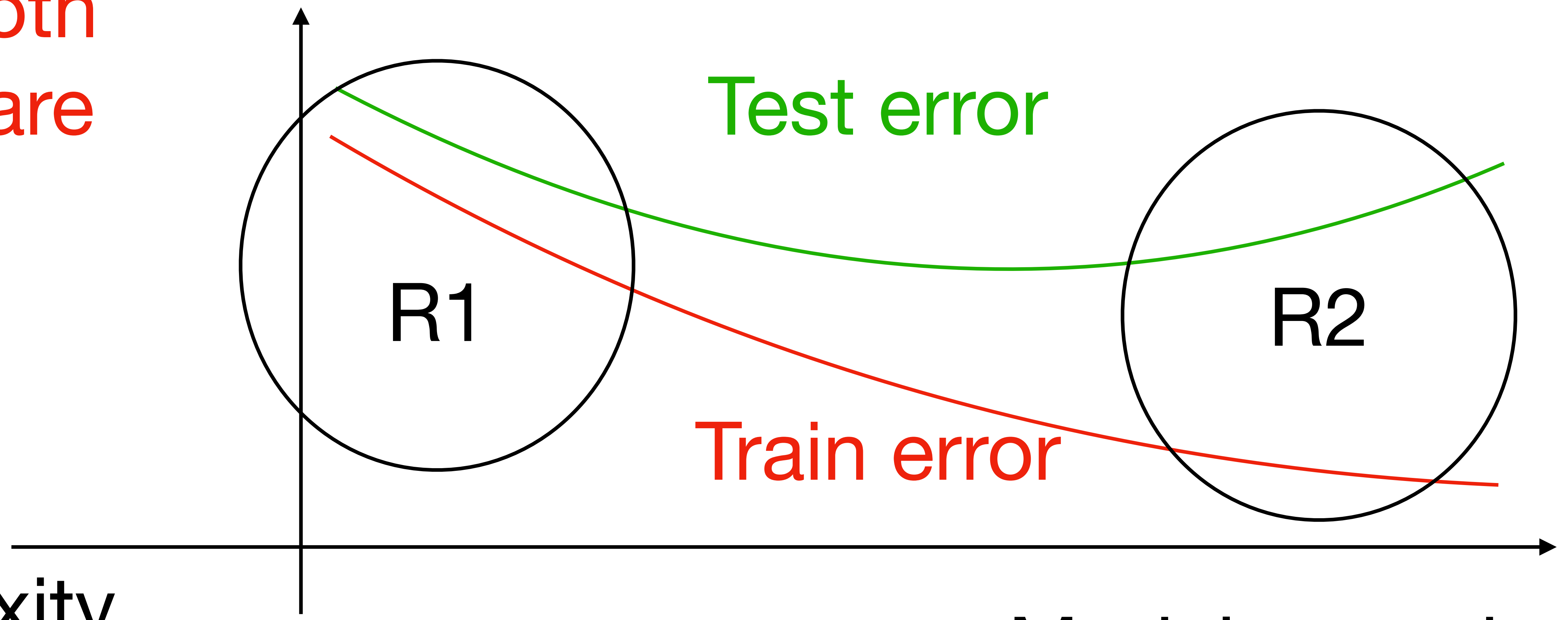
R1: Underfitting (both train and test errs are large)

Suggestions:

1. Increase complexity of models

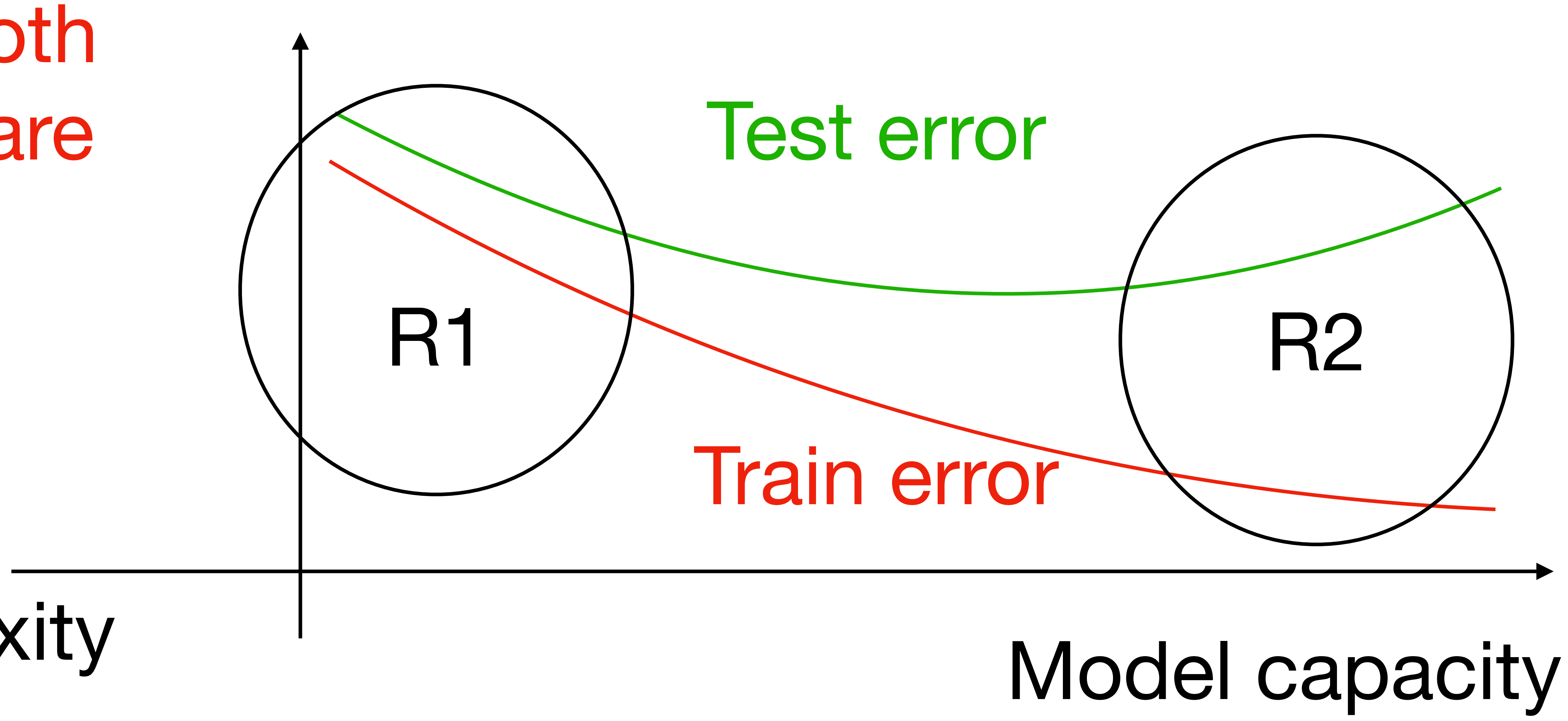
2. More features

3. Using Boosting (we will see it later)



# Practical Suggestions for combating over/under fitting

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




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

Suggestions:

1. Increase complexity of models

2. More features

3. Using Boosting (we will see it later)

# Practical Suggestions for combating over/under fitting

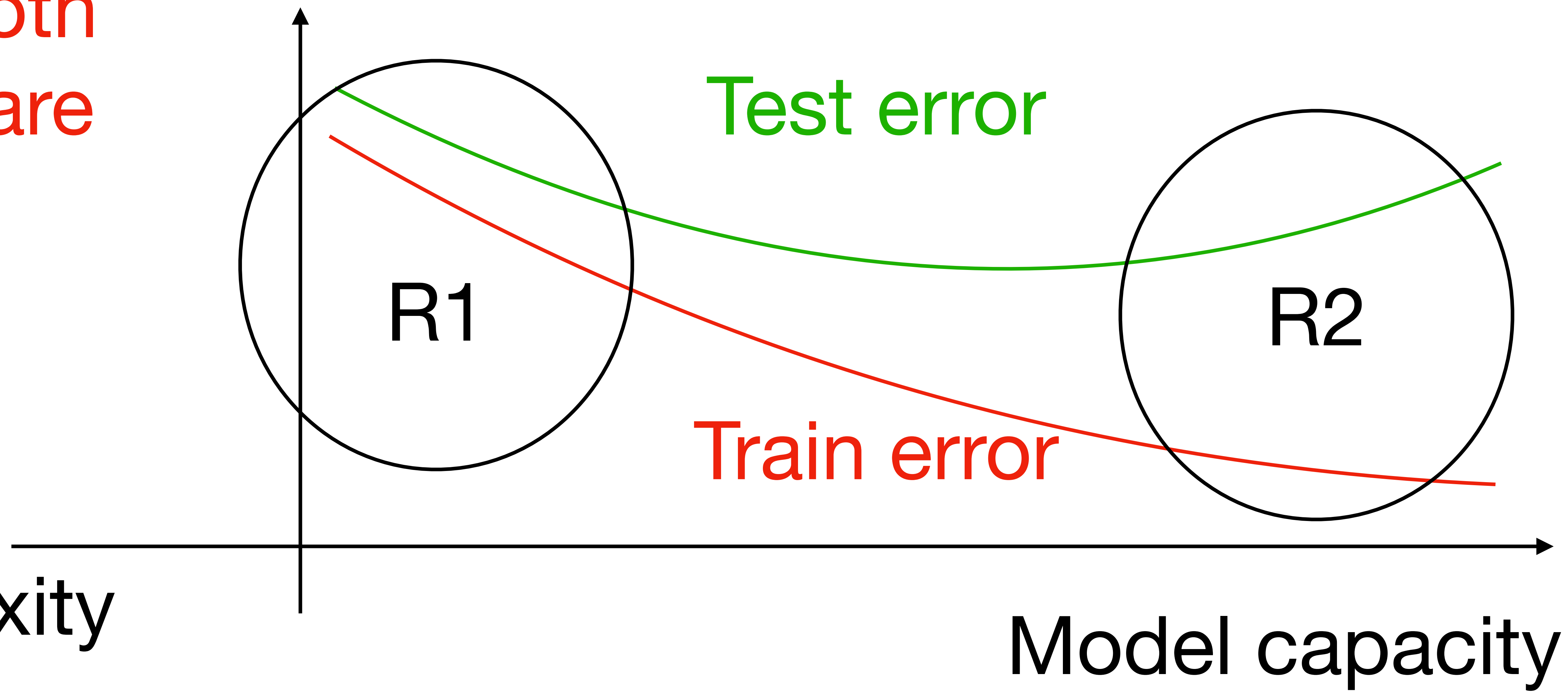
R1: Underfitting (both train and test errs are large)

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)

Suggestions:

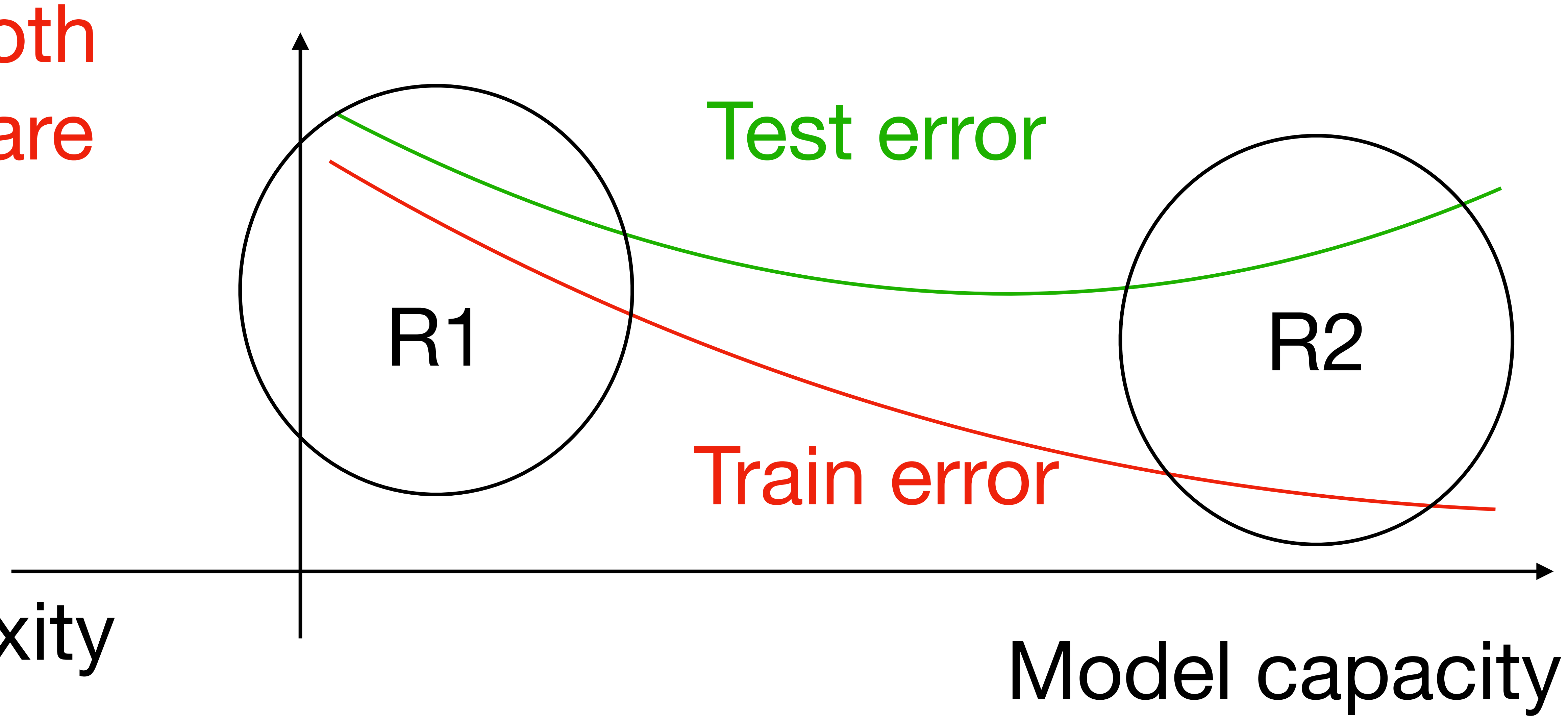
1. More train data

# 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

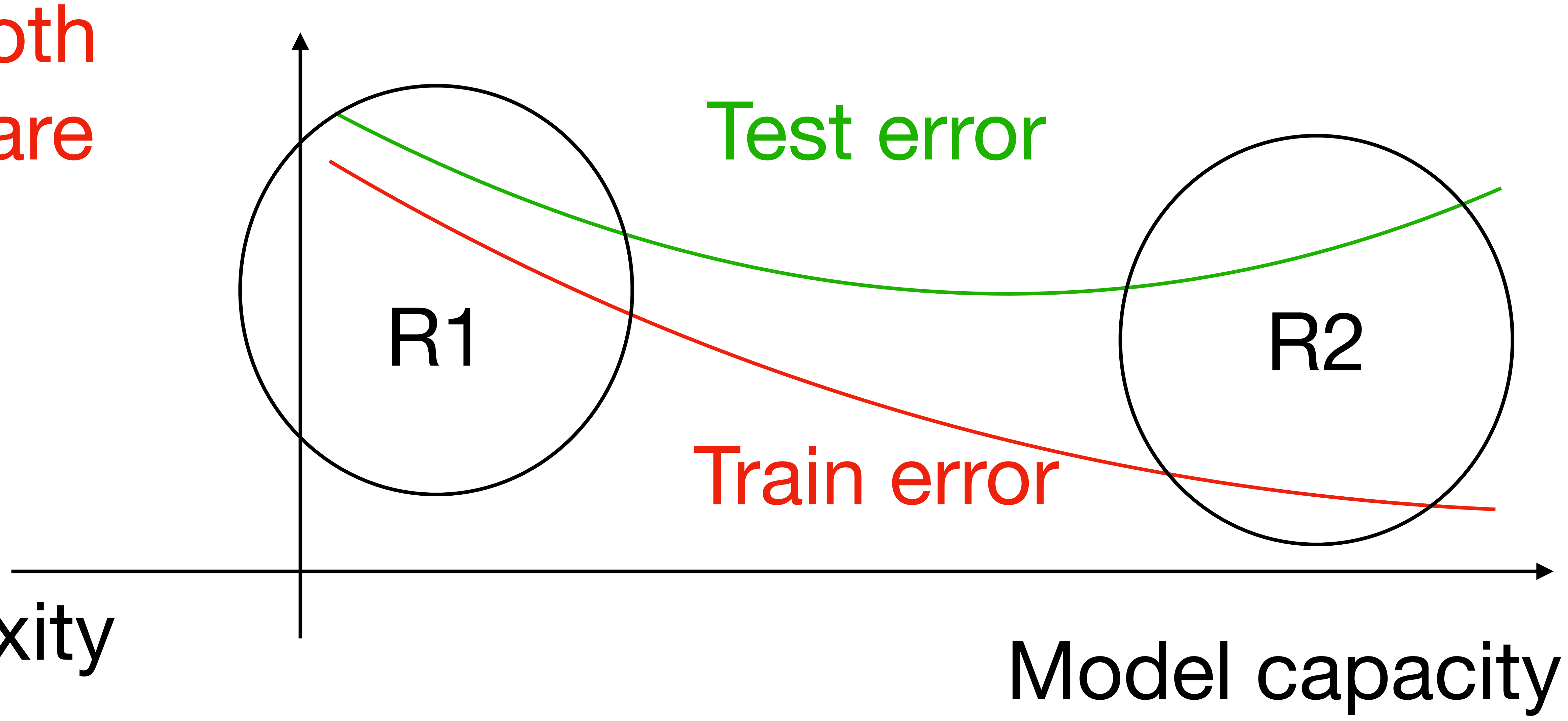
R1: Underfitting (both train and test errs are large)

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)