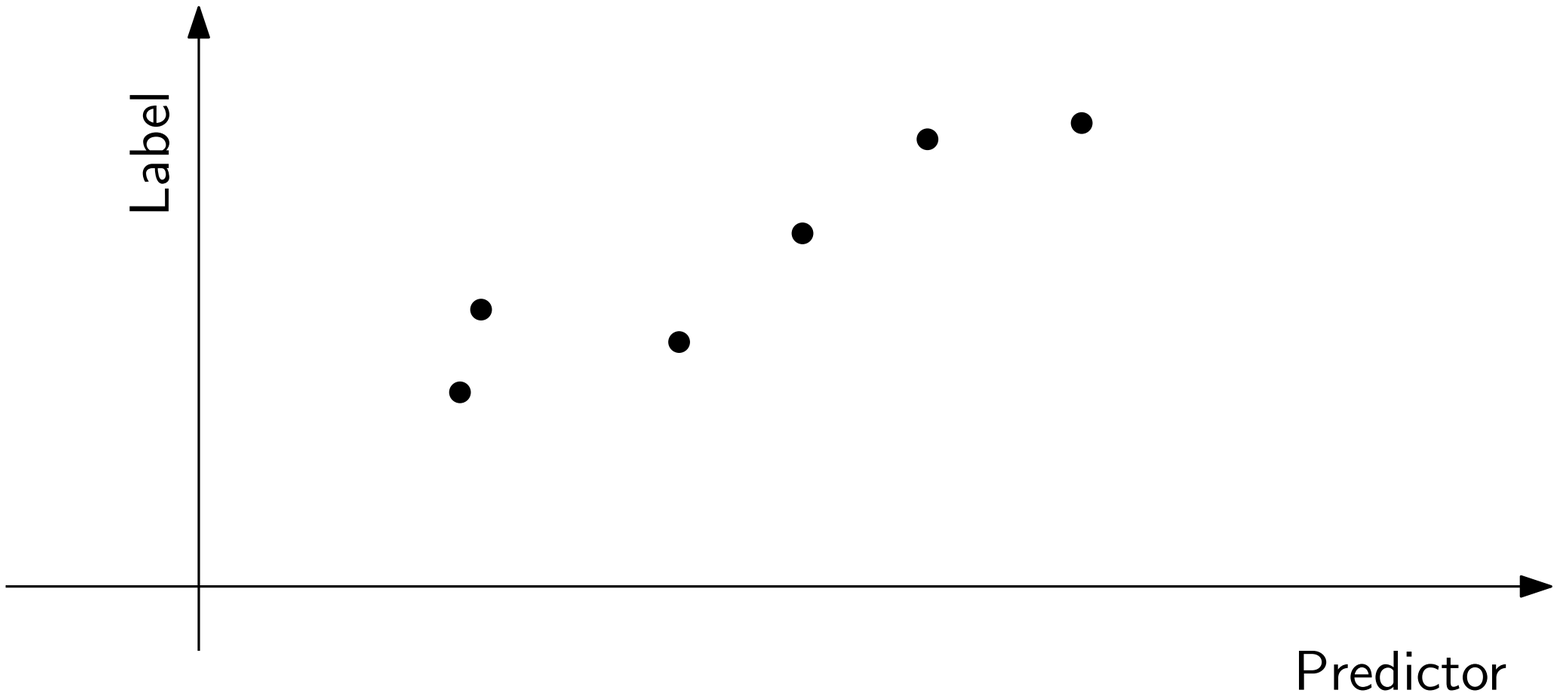
Statistical Learning STA4042

INTERPOLATE, EXTRAOPLATE & OVERFIT

SCHOOL OF DATA SCIENCE

Interpolation and Extrapolation

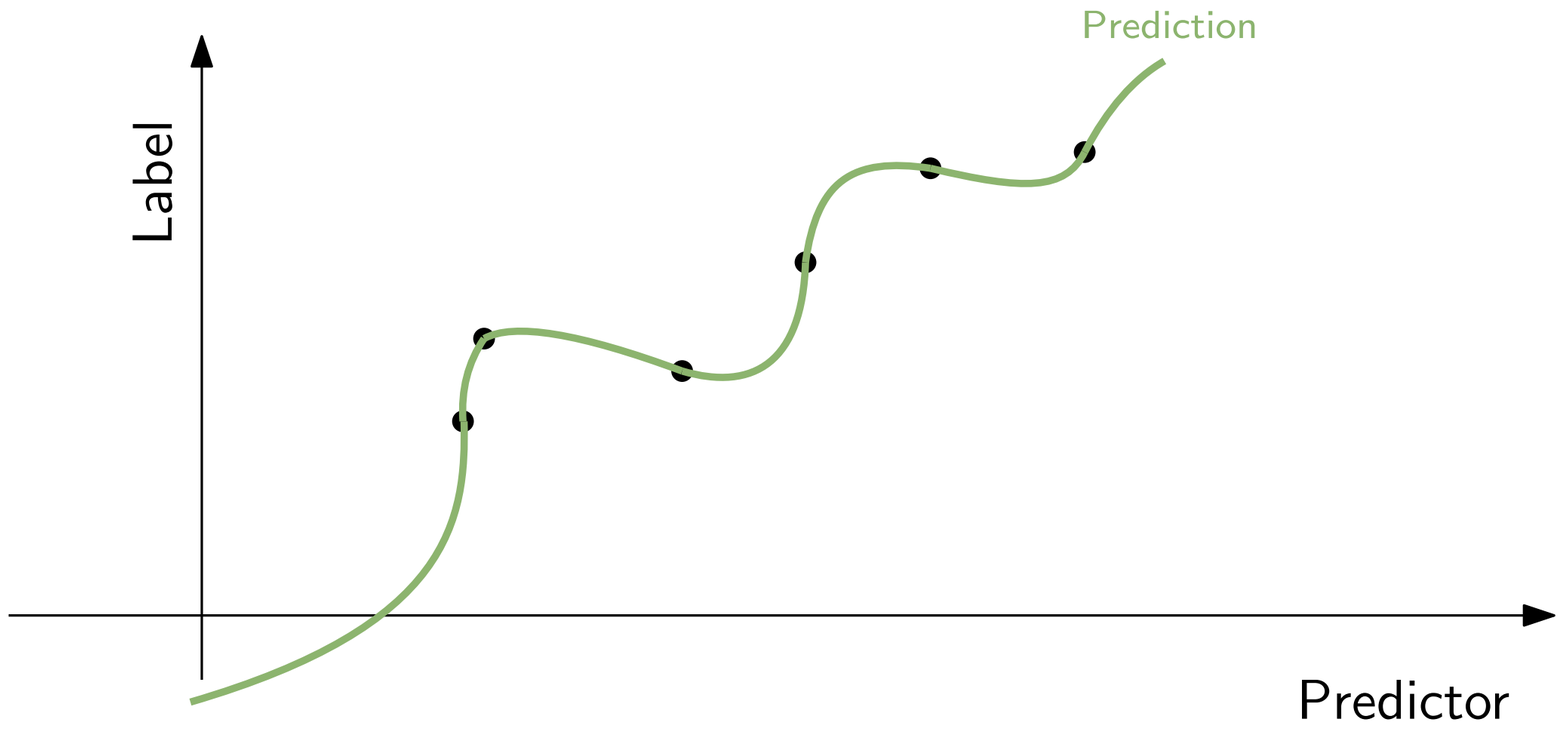


Statistical Learning STA4042

INTERPOLATE, EXTRAOPLATE & OVERFIT

SCHOOL OF DATA SCIENCE

Interpolation and Extrapolation

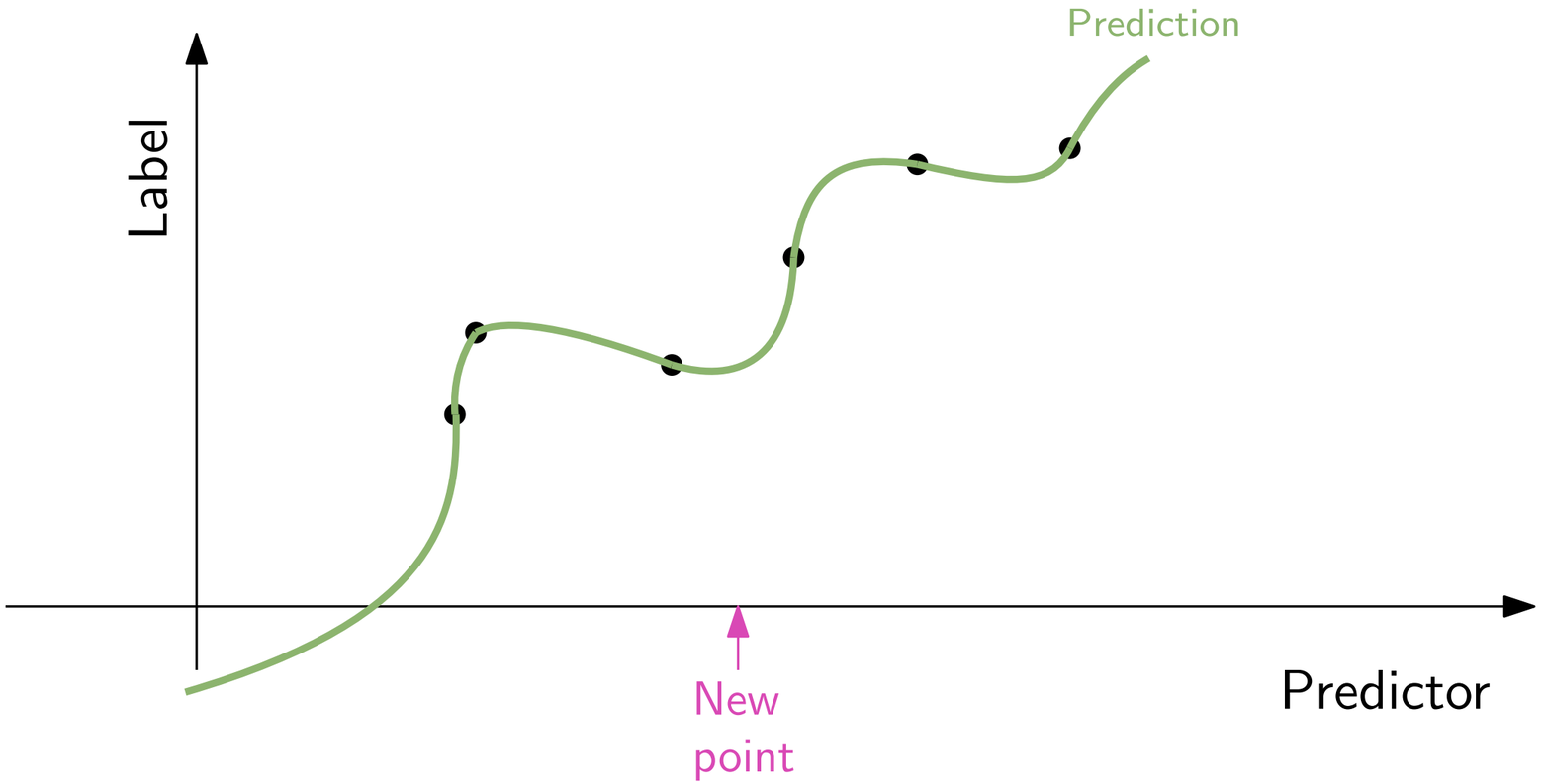


Statistical Learning STA4042

INTERPOLATE, EXTRAOPLATE & OVERFIT

SCHOOL OF DATA SCIENCE

Interpolation and Extrapolation

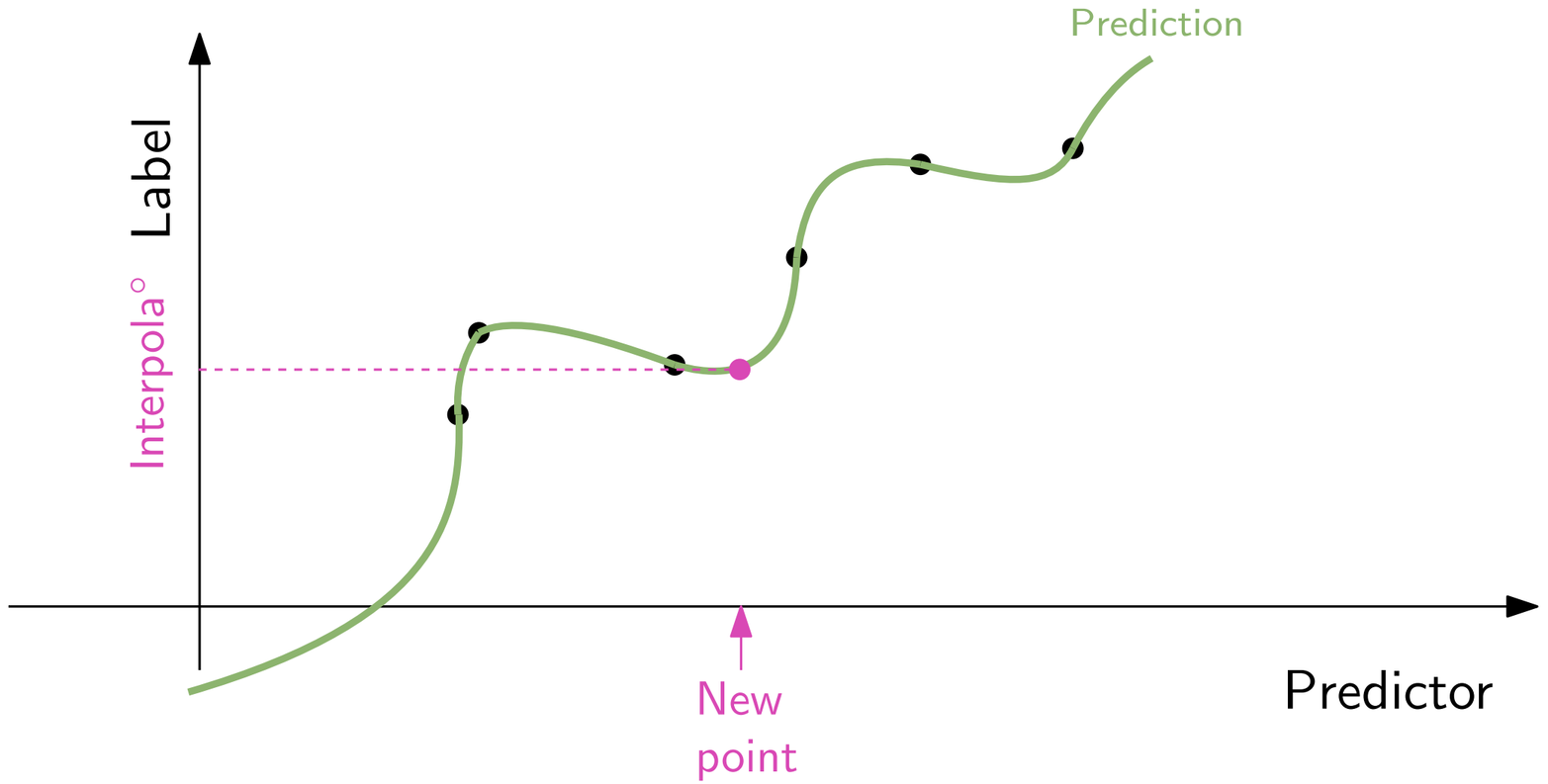


Statistical Learning STA4042

INTERPOLATE, EXTRAOPLATE & OVERFIT

SCHOOL OF DATA SCIENCE

Interpolation and Extrapolation

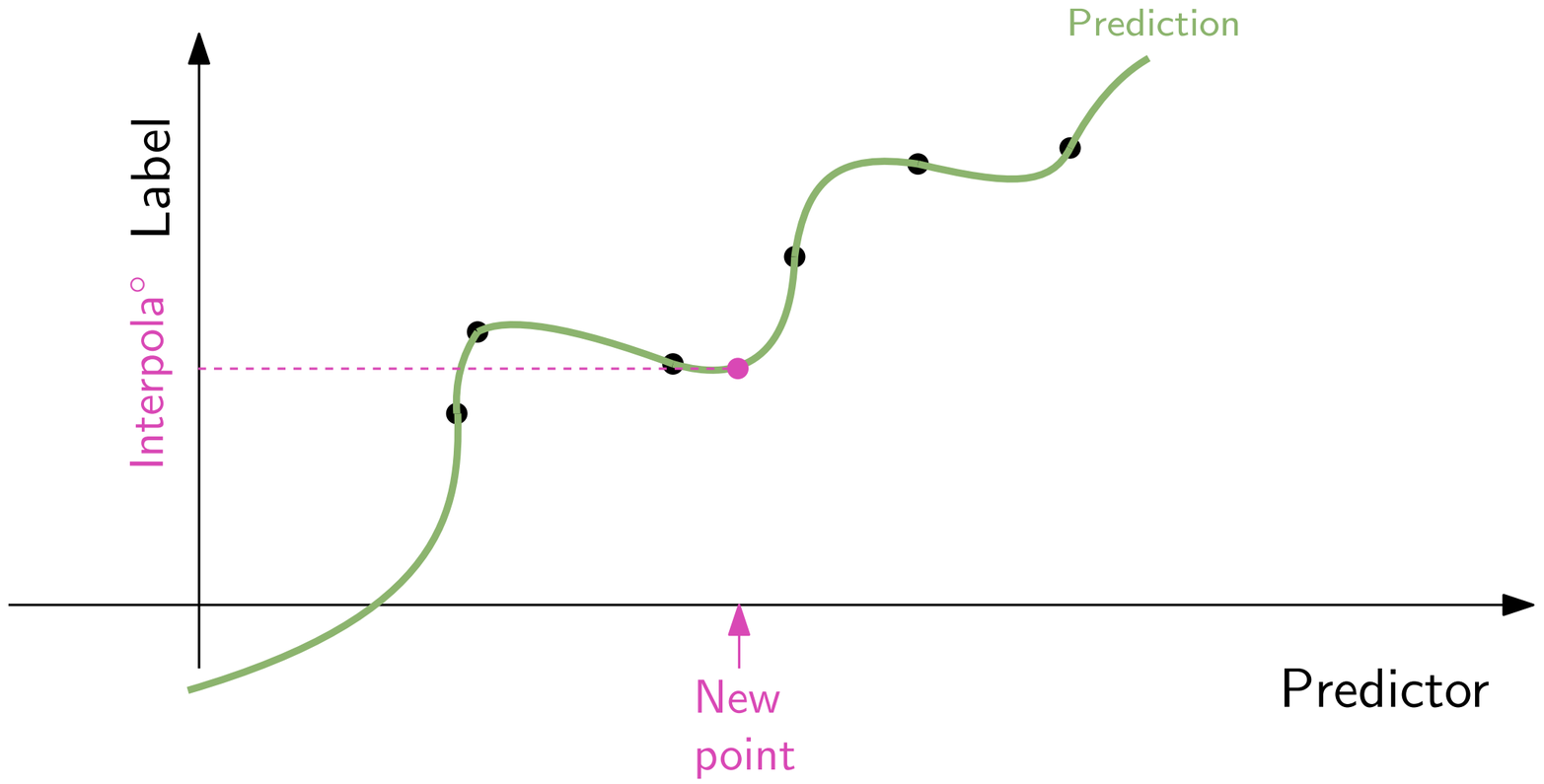


Statistical Learning STA4042

INTERPOLATE, EXTRAOPLATE & OVERFIT

SCHOOL OF DATA SCIENCE

Interpolation and Extrapolation



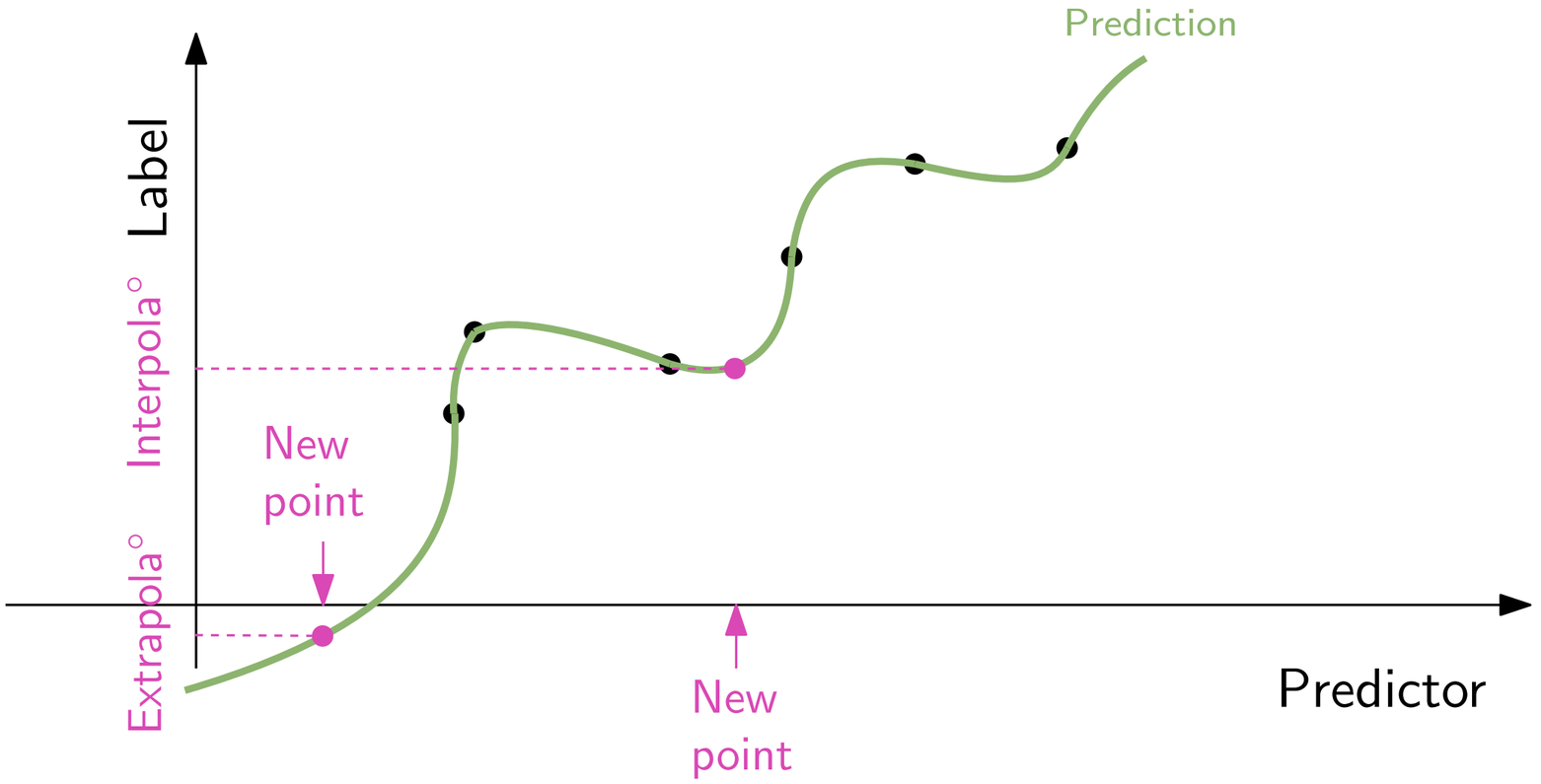
Interpolation: predict between points

Statistical Learning STA4042

INTERPOLATE, EXTRAOPLATE & OVERFIT

SCHOOL OF DATA SCIENCE

Interpolation and Extrapolation



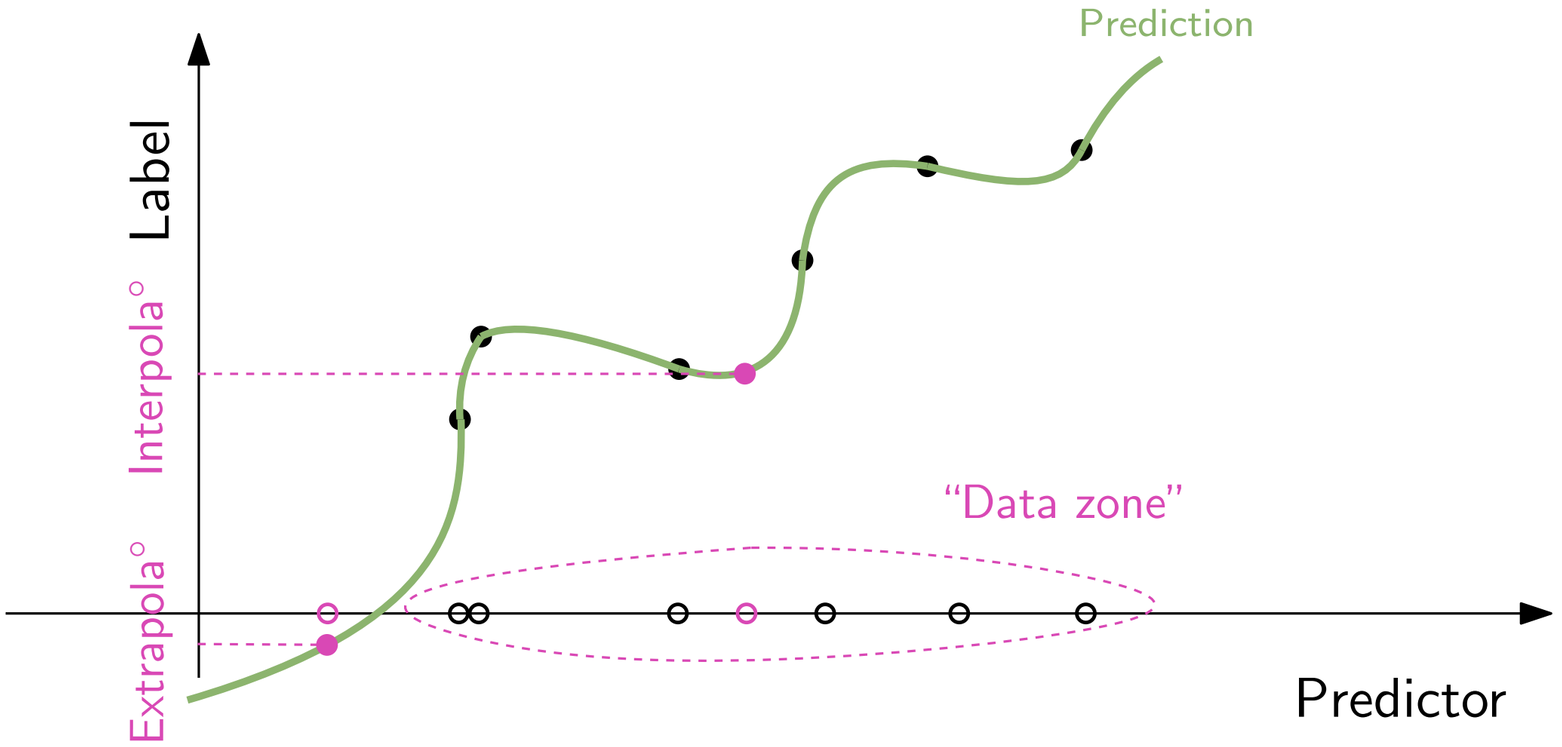
Interpolation: predict between points

Statistical Learning STA4042

INTERPOLATE, EXTRAOPLATE & OVERFIT

SCHOOL OF DATA SCIENCE

Interpolation and Extrapolation



Interpolation: predict between points

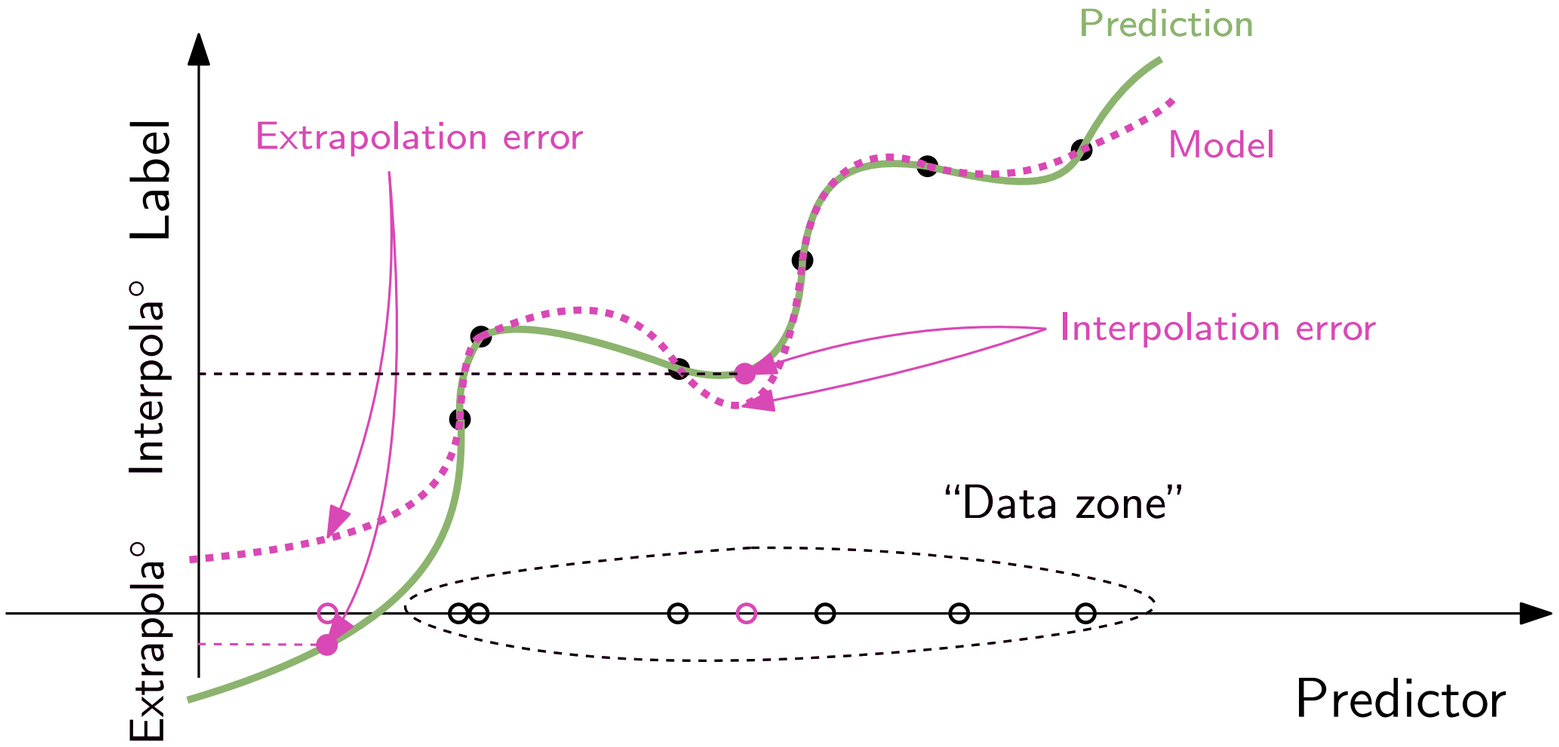
Extrapolation: predict outside of the data zone

Statistical Learning STA4042

INTERPOLATE, EXTRAOPLATE & OVERFIT

SCHOOL OF DATA SCIENCE

Interpolation and Extrapolation



Interpolation: predict between points

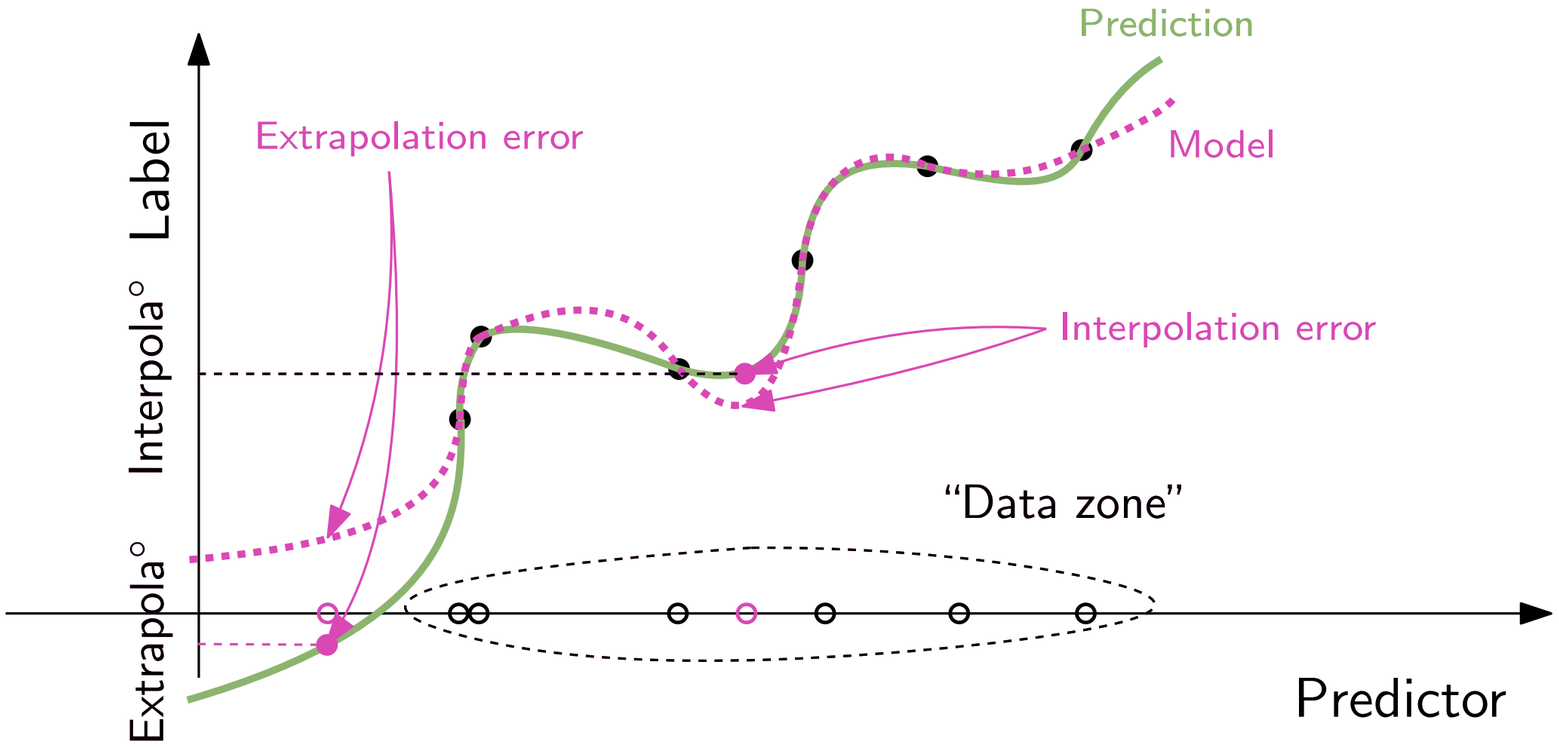
Extrapolation: predict outside of the data zone

Statistical Learning STA4042

INTERPOLATE, EXTRAOPLATE & OVERFIT

SCHOOL OF DATA SCIENCE

Interpolation and Extrapolation



Interpolation: predict between points

Extrapolation: predict outside of the data zone

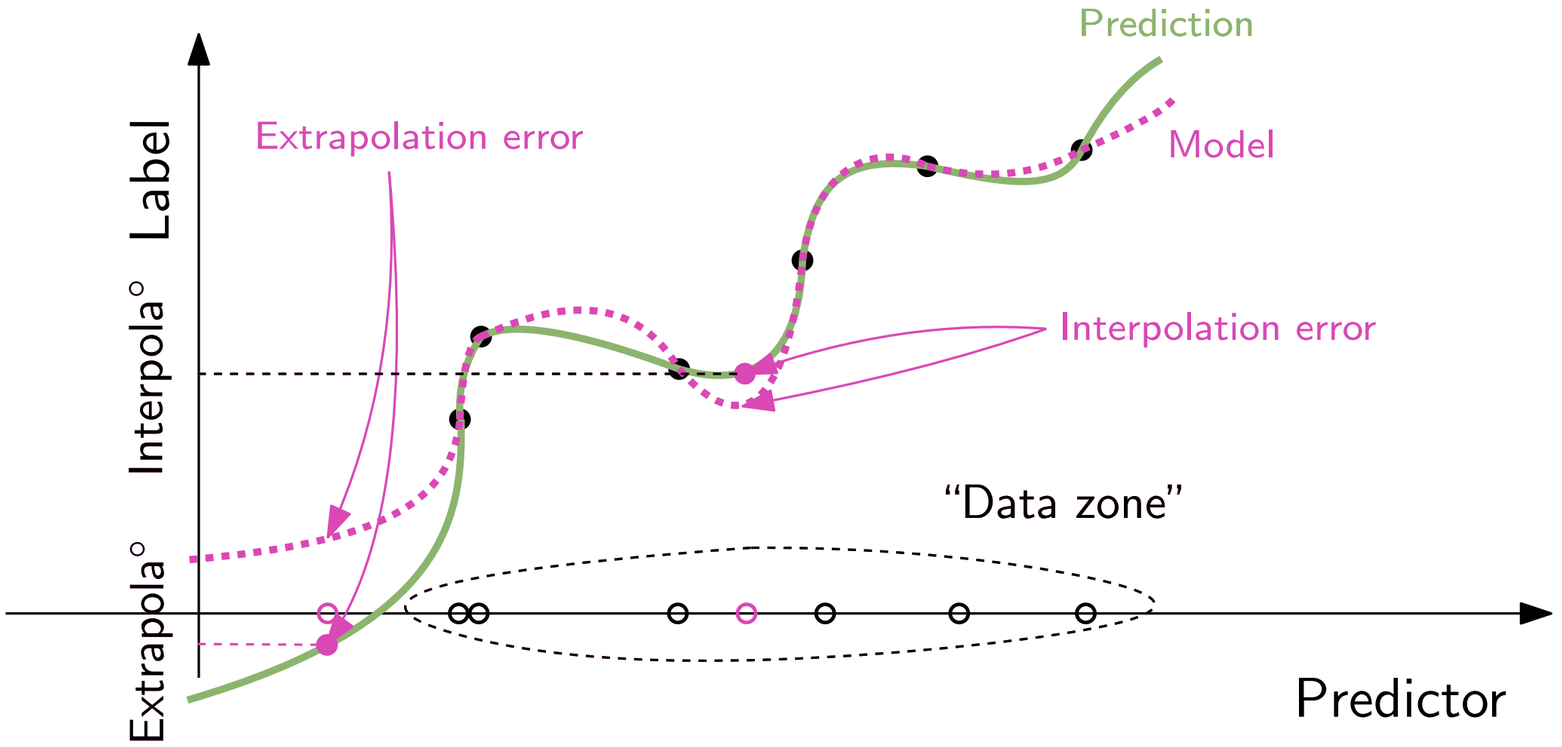
- Geometric opposition not relevant in machine learning (except in "transfer learning") because no points arrive outside of the data zone

Statistical Learning STA4042

INTERPOLATE, EXTRAOPLATE & OVERFIT

SCHOOL OF DATA SCIENCE

Interpolation and Extrapolation



Interpolation: predict between points

Extrapolation: predict outside of the data zone

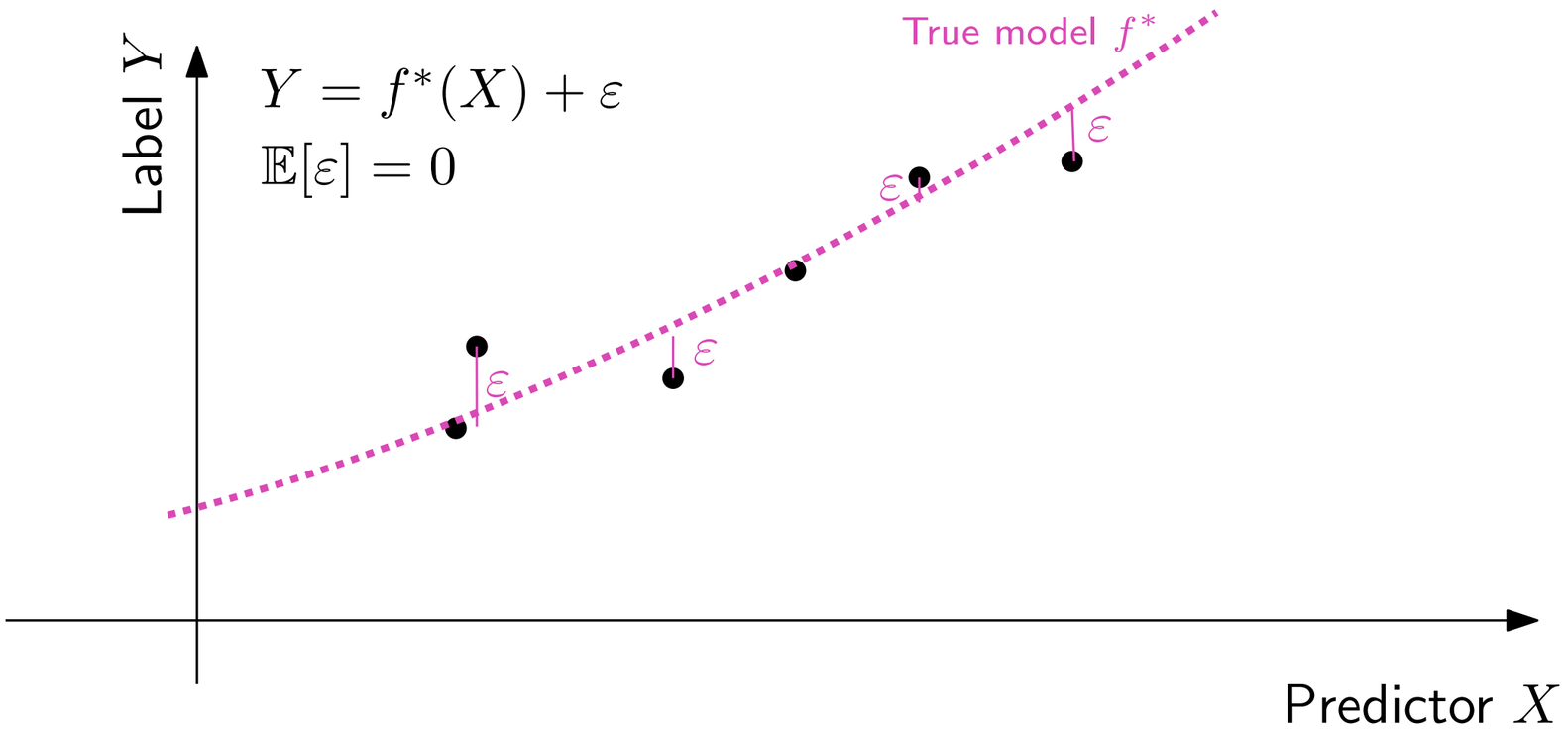
- Geometric opposition not relevant in machine learning (except in "transfer learning") because no points arrive outside of the data zone
- In statistical learning no model fits the data because of intrinsic noise

Statistical Learning STA4042

INTERPOLATE, EXTRAOPLATE & OVERFIT

SCHOOL OF DATA SCIENCE

Overfitting



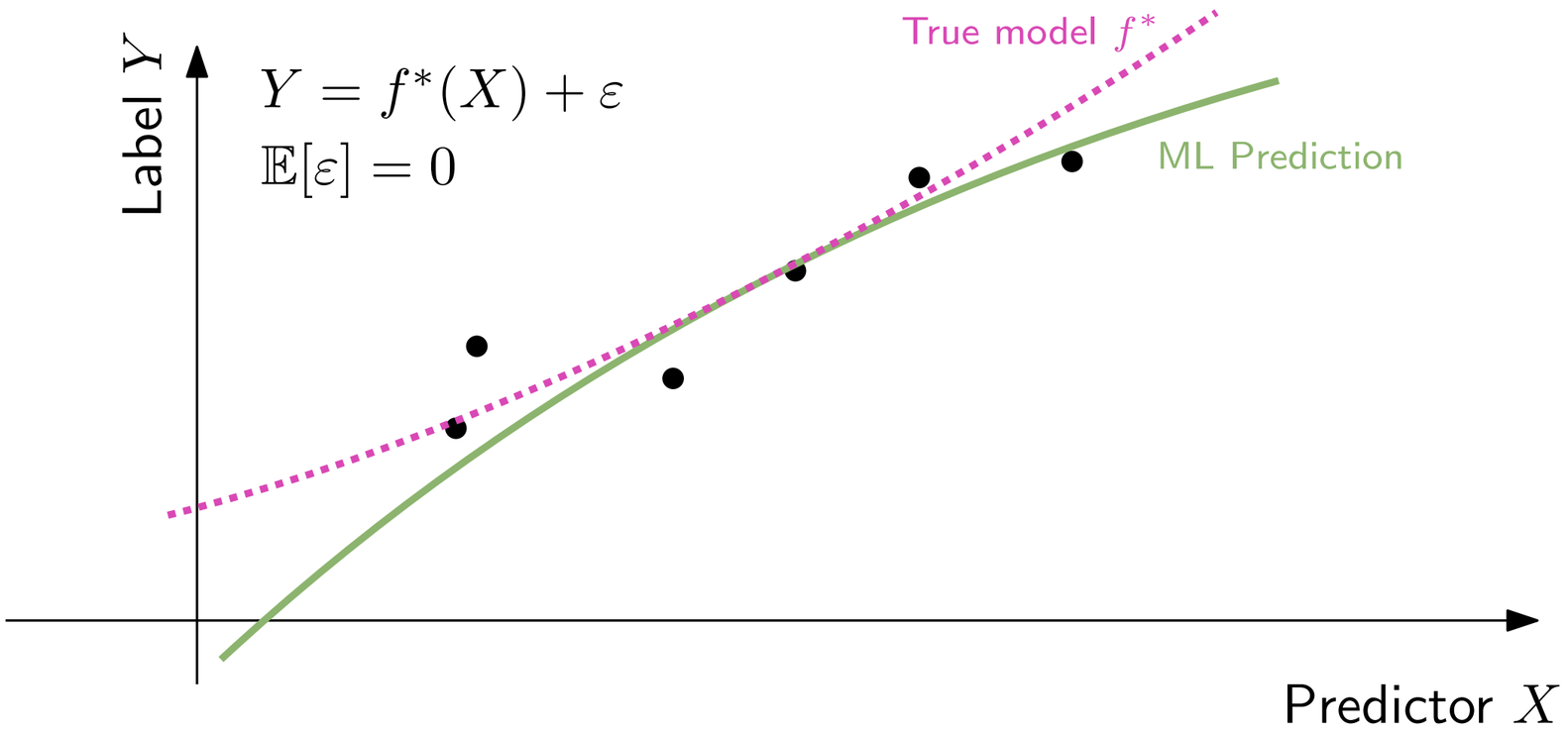
In satistical learning, 0 error generally unreachable because of ε.

Statistical Learning STA4042

INTERPOLATE, EXTRAOPLATE & OVERFIT

SCHOOL OF DATA SCIENCE

Overfitting



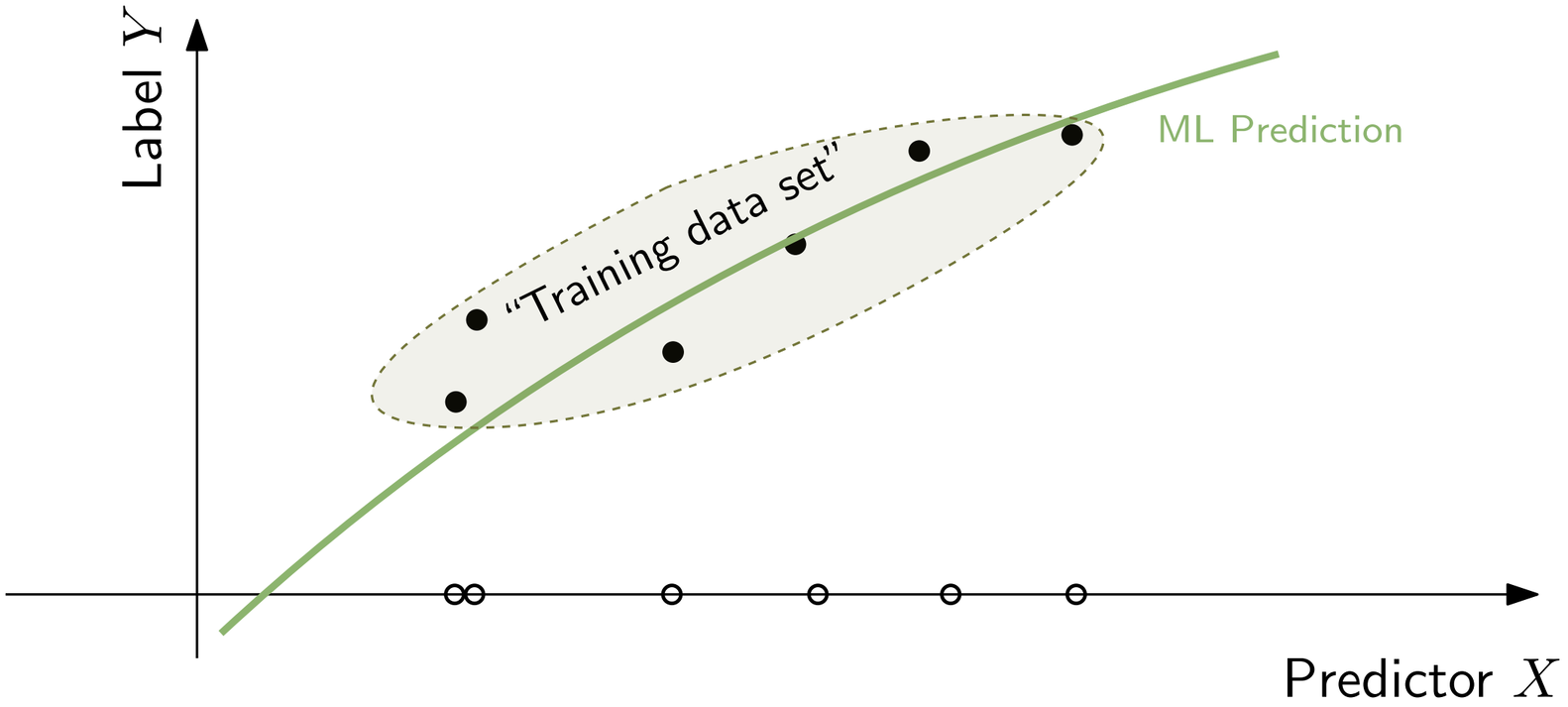
In satistical learning, 0 error generally unreachable because of ε.

Statistical Learning STA4042

INTERPOLATE, EXTRAOPLATE & OVERFIT

SCHOOL OF DATA SCIENCE

Overfitting



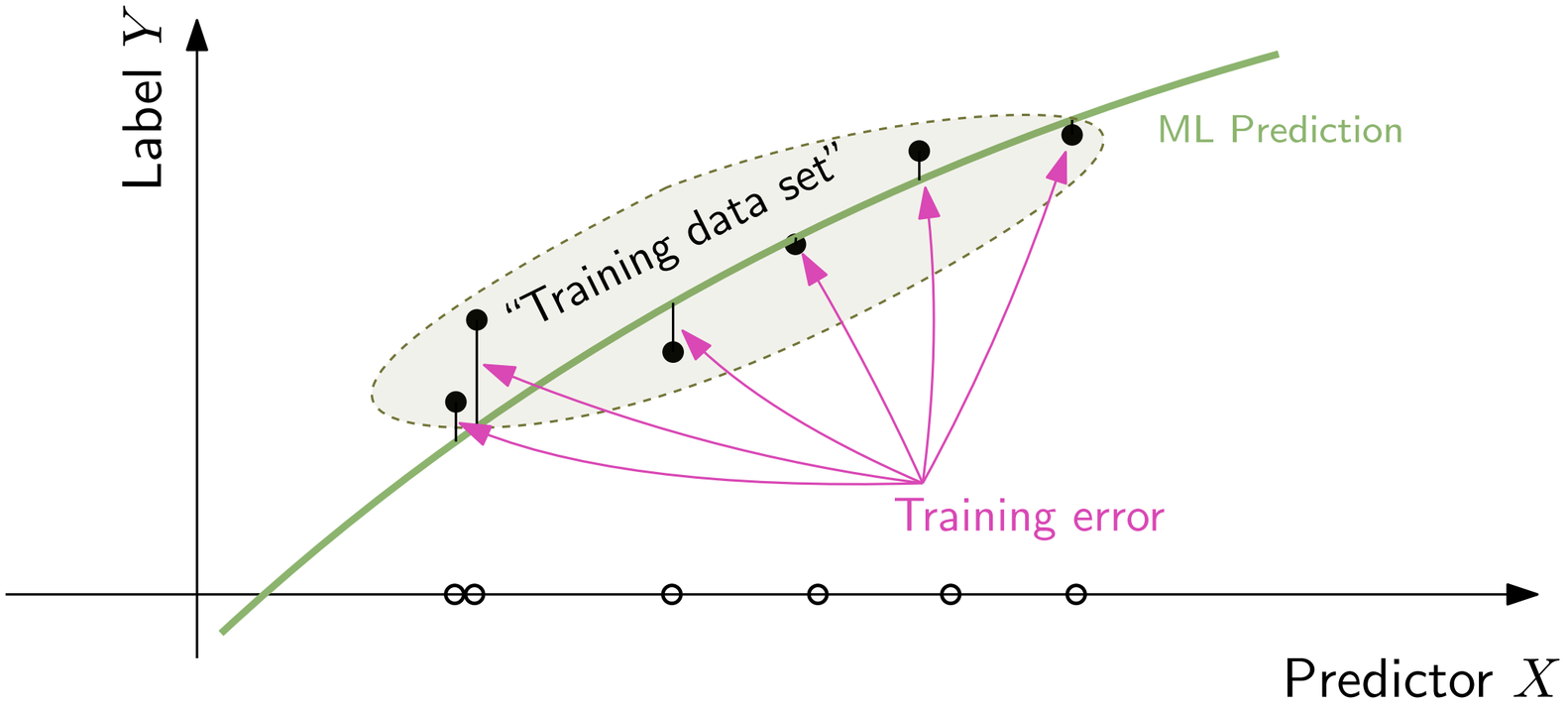
In satistical learning, 0 error generally unreachable because of ε.

Statistical Learning STA4042

INTERPOLATE, EXTRAOPLATE & OVERFIT

SCHOOL OF DATA SCIENCE

Overfitting



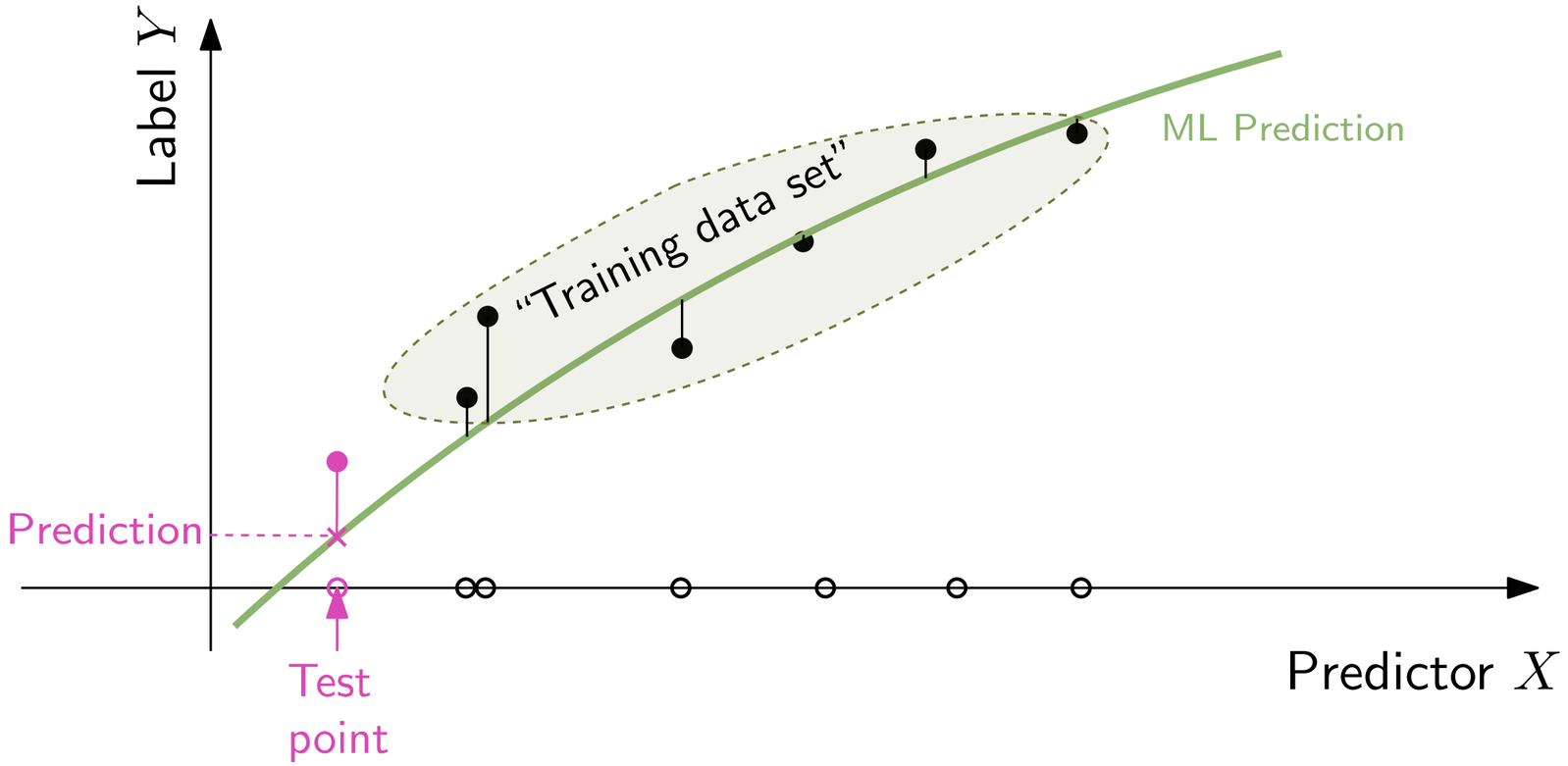
In satistical learning, 0 error generally unreachable because of ε.

Statistical Learning STA4042

INTERPOLATE, EXTRAOPLATE & OVERFIT

SCHOOL OF DATA SCIENCE

Overfitting



In satistical learning, 0 error generally unreachable because of ε.

Question of overfitting:

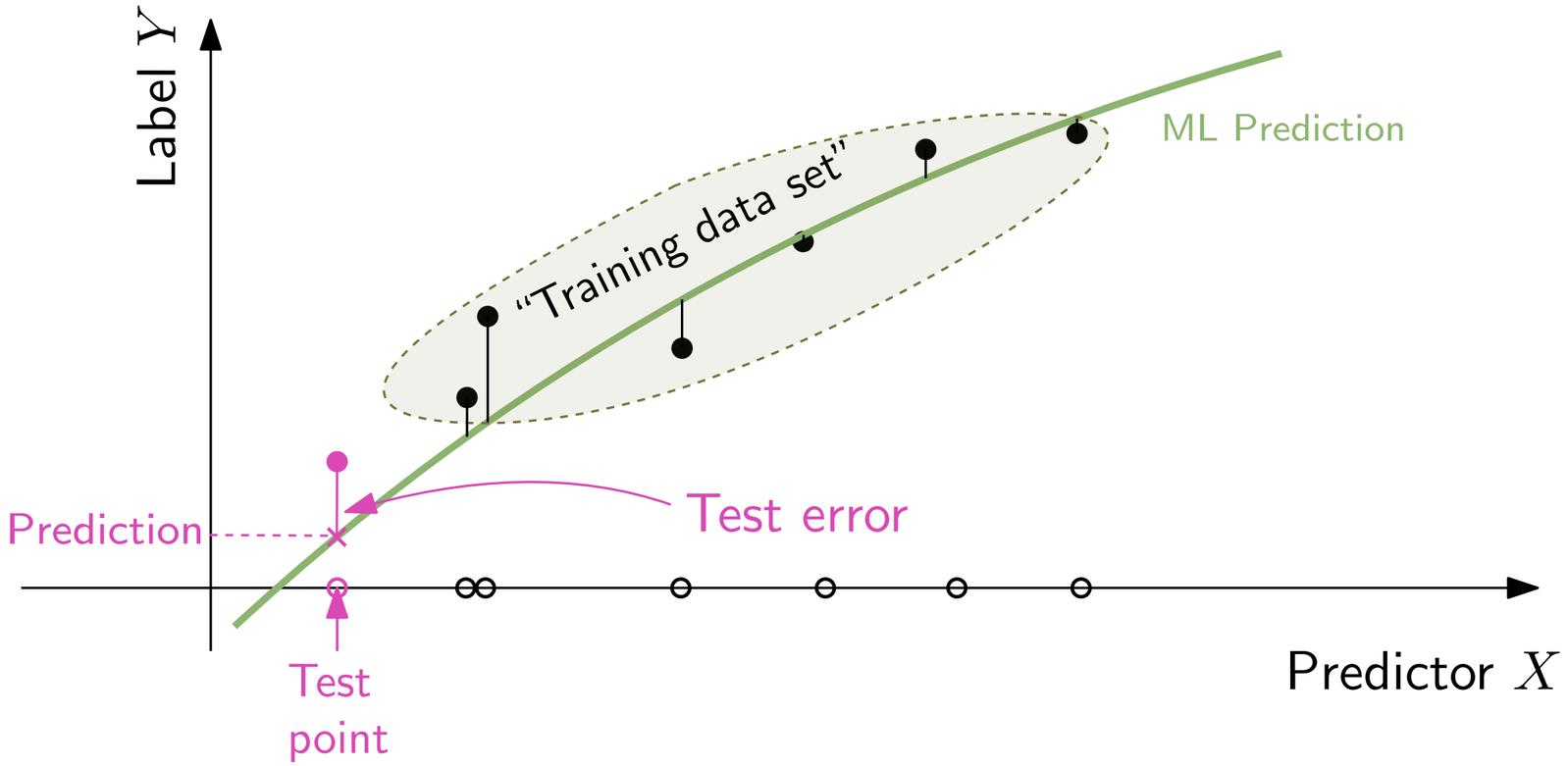
Is the prediction error on test data comparable to the prediction error on the train dataset?

Statistical Learning STA4042

INTERPOLATE, EXTRAOPLATE & OVERFIT

SCHOOL OF DATA SCIENCE

Overfitting



In satistical learning, 0 error generally unreachable because of ε.

Question of overfitting:

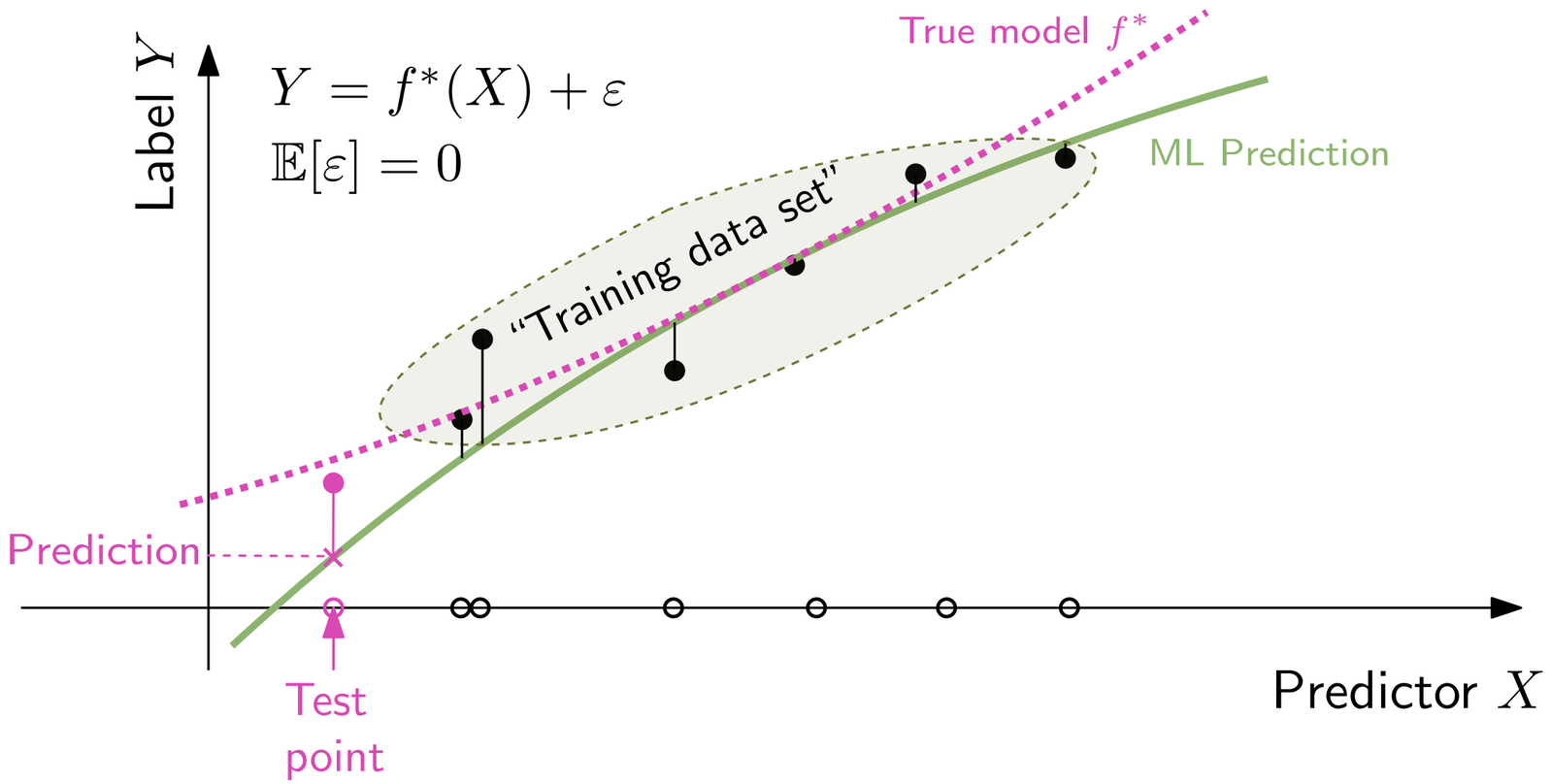
Is the prediction error on test data comparable to the prediction error on the train dataset?

Statistical Learning STA4042

INTERPOLATE, EXTRAOPLATE & OVERFIT

SCHOOL OF DATA SCIENCE

Overfitting



In satistical learning, 0 error generally unreachable because of ε.

Question of overfitting:

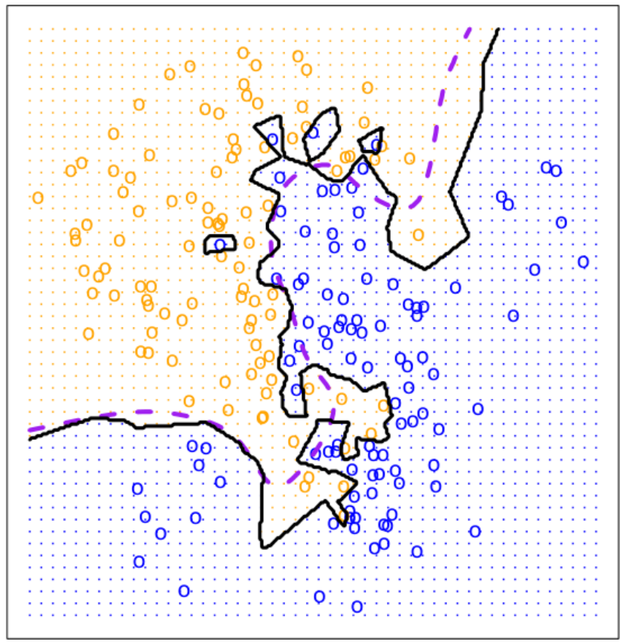
Is the prediction error on test data comparable to the prediction error on the train dataset?

True model minimizes in expectation the test error

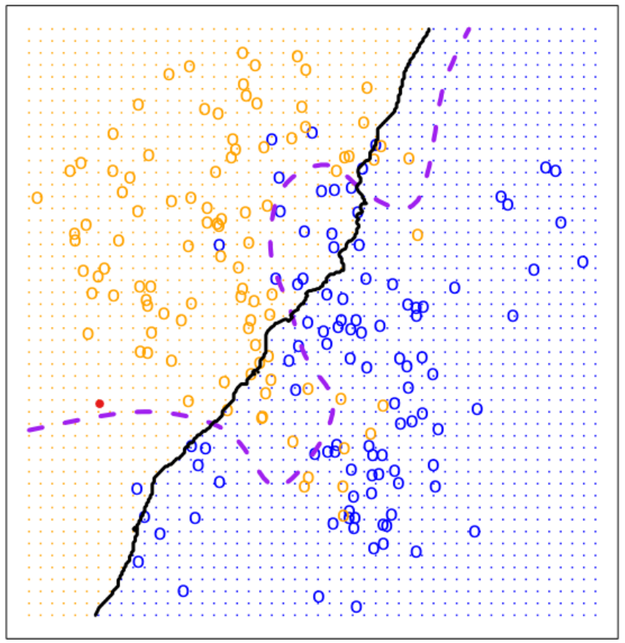
Example with k-nearest neighbors

Trade-off on the number of neighbors k:

$k = 1$



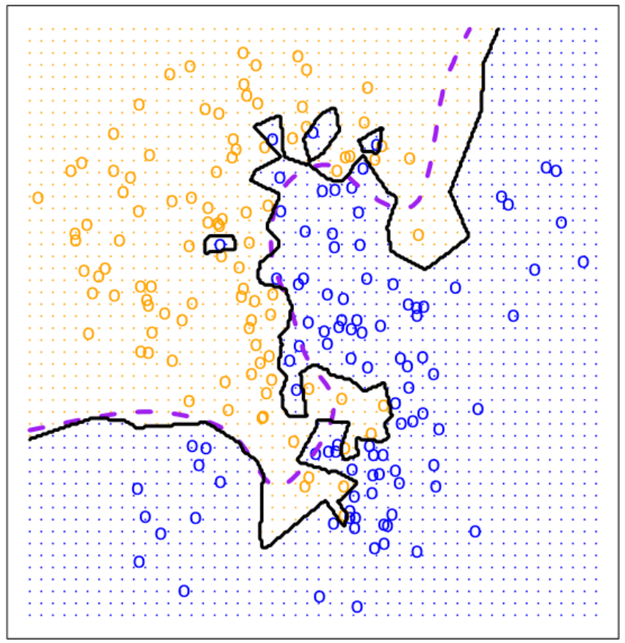
$k = 100$



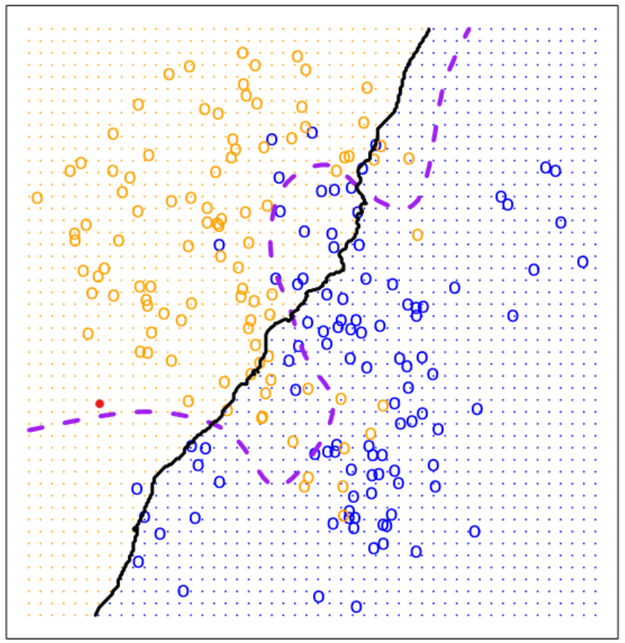
Example with k-nearest neighbors

Trade-off on the number of neighbors k:

$k = 1$



$k = 100$

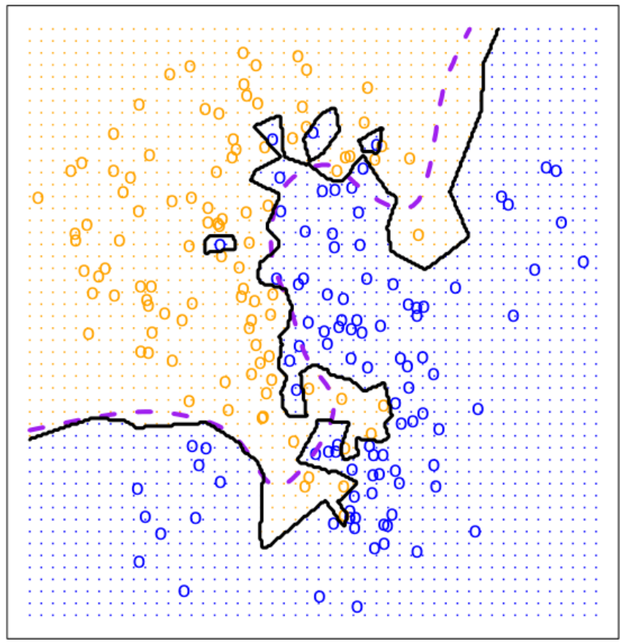


Overfitting

Example with k-nearest neighbors

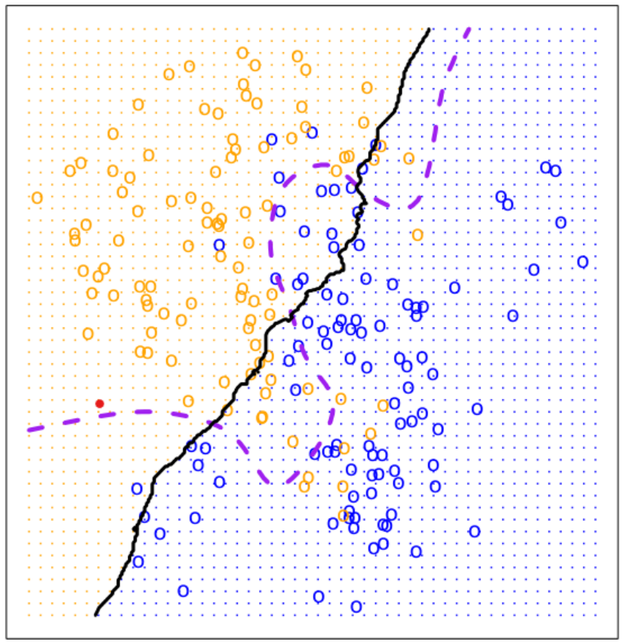
Trade-off on the number of neighbors k:

$k = 1$



Overfitting

$k = 100$

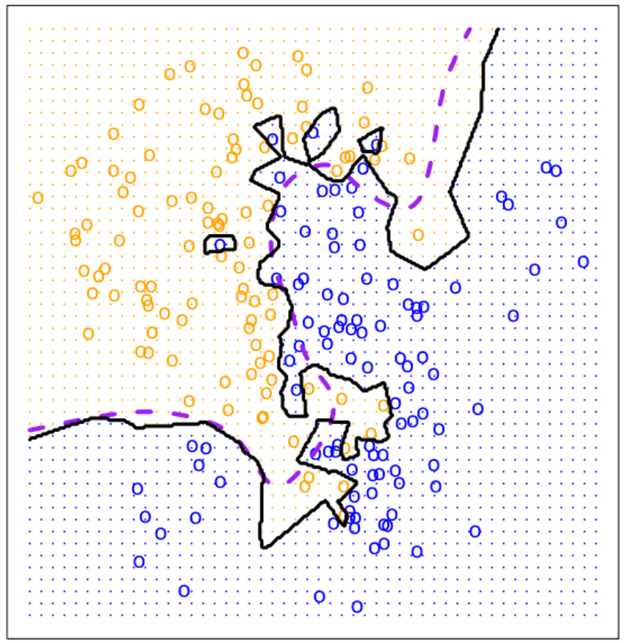


Bad interpolation

Example with k-nearest neighbors

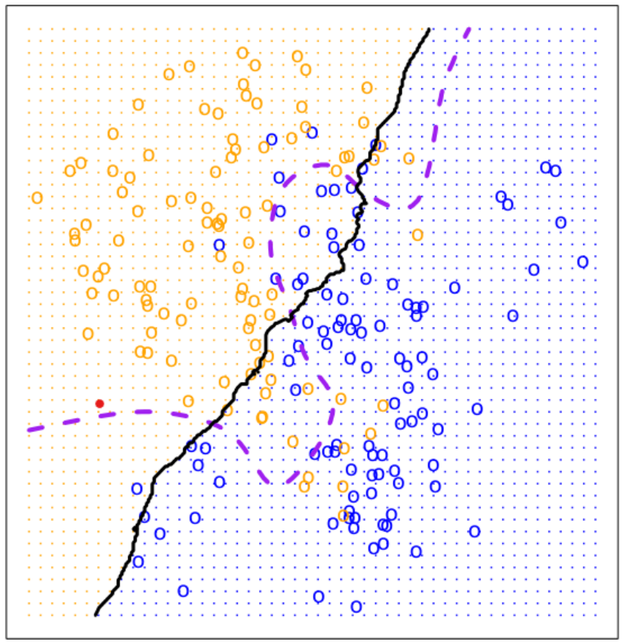
Trade-off on the number of neighbors k:

$k = 1$



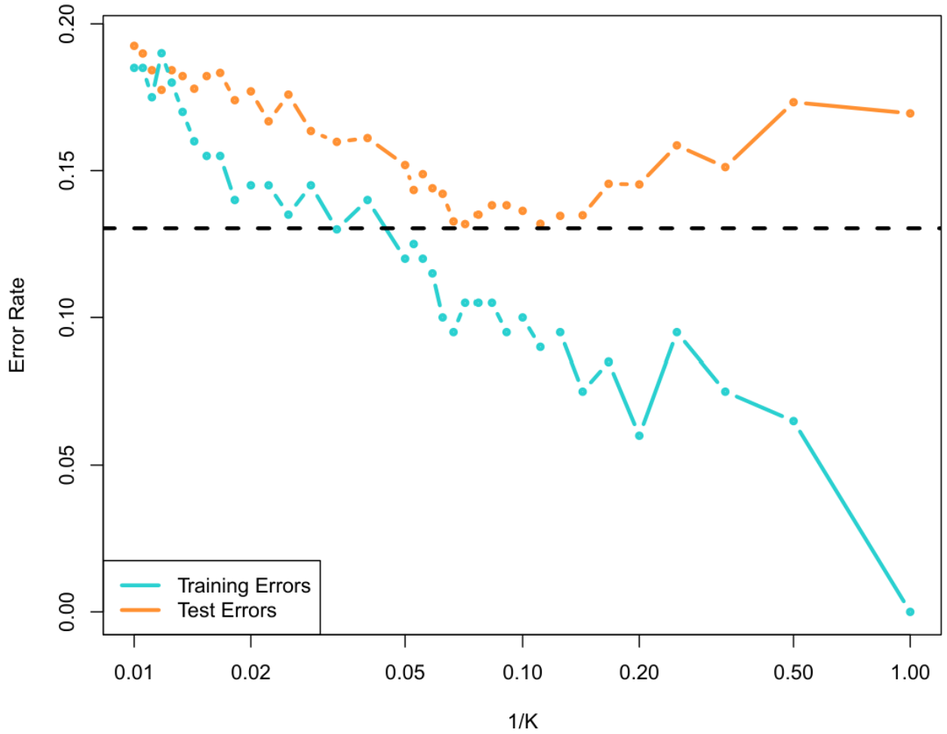
Overfitting

$k = 100$

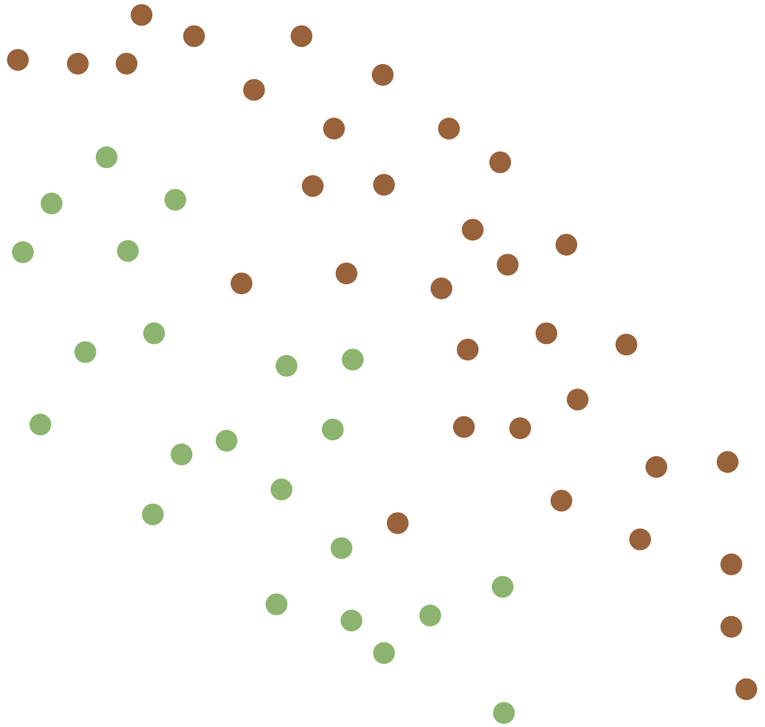


Bad interpolation

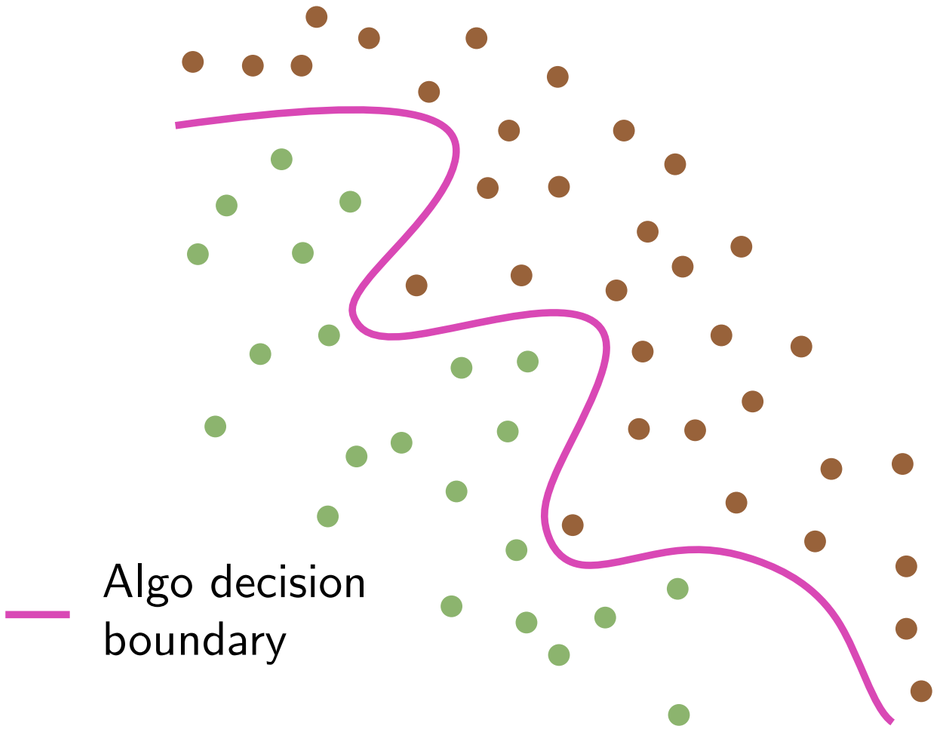
Look for a trade-off



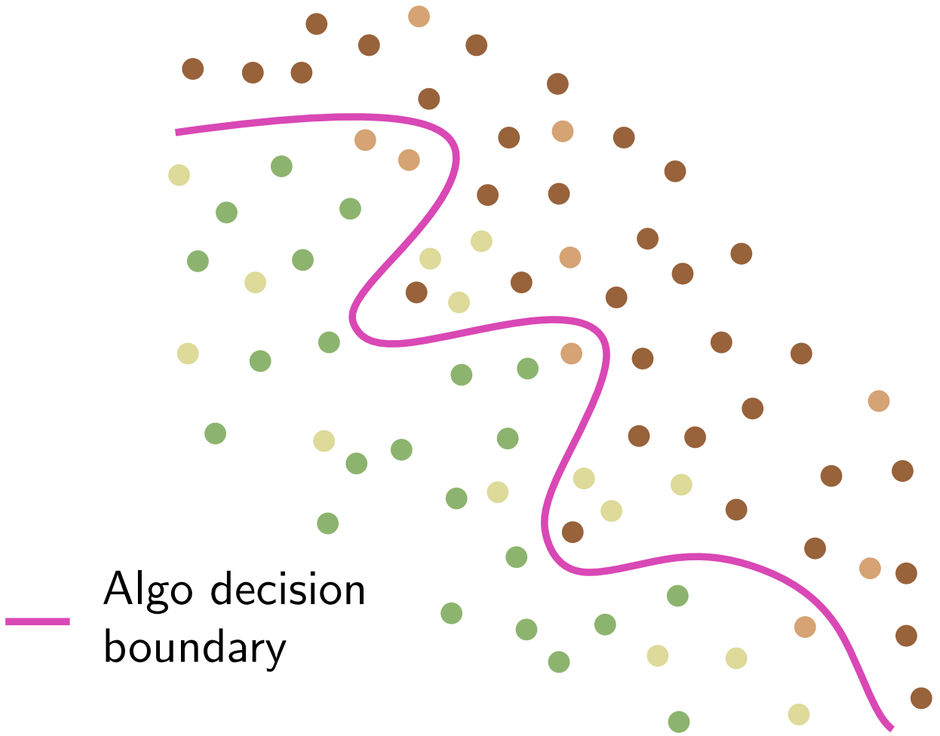
Overfitting: Geometrical interpretation



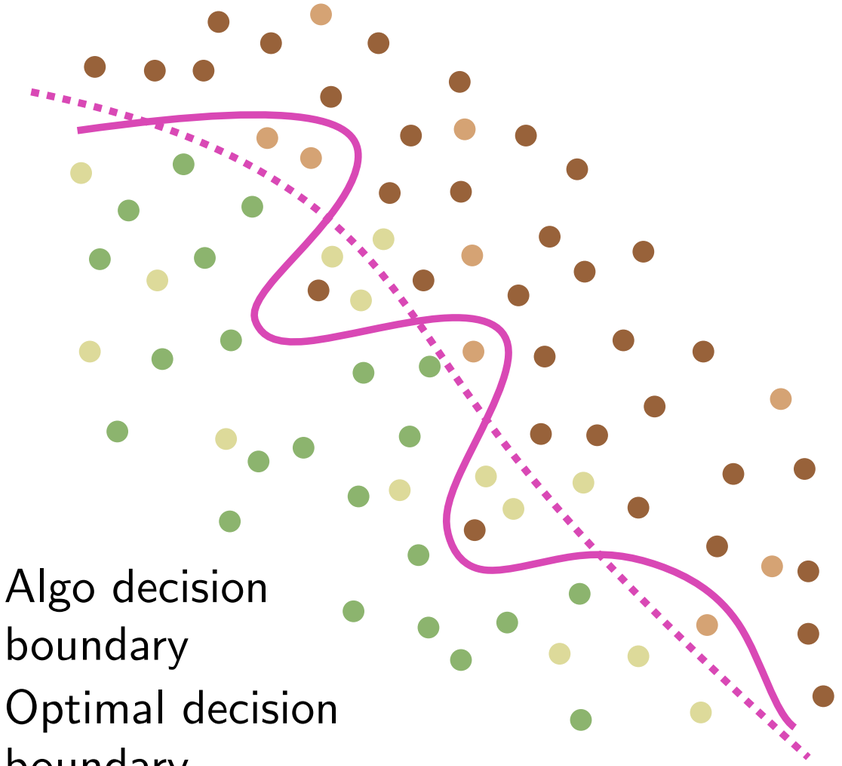
Overfitting: Geometrical interpretation



Overfitting: Geometrical interpretation



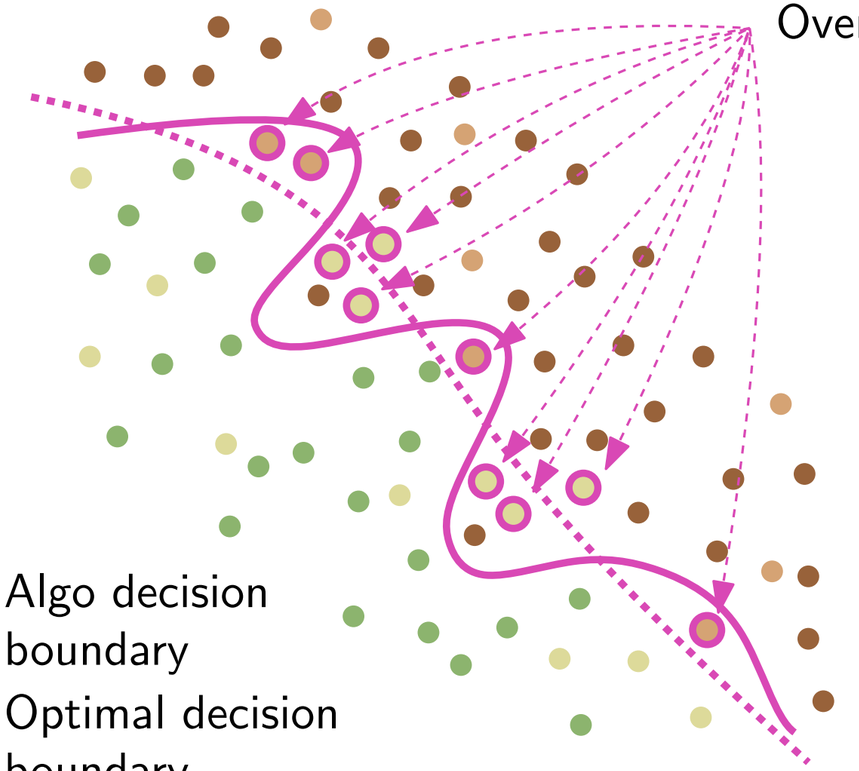
Overfitting: Geometrical interpretation



Algo decision boundary

Optimal decision boundary

Overfitting: Geometrical interpretation



Remarks:

- Algo boundary too complicated $\rightarrow$ need to simplify the model

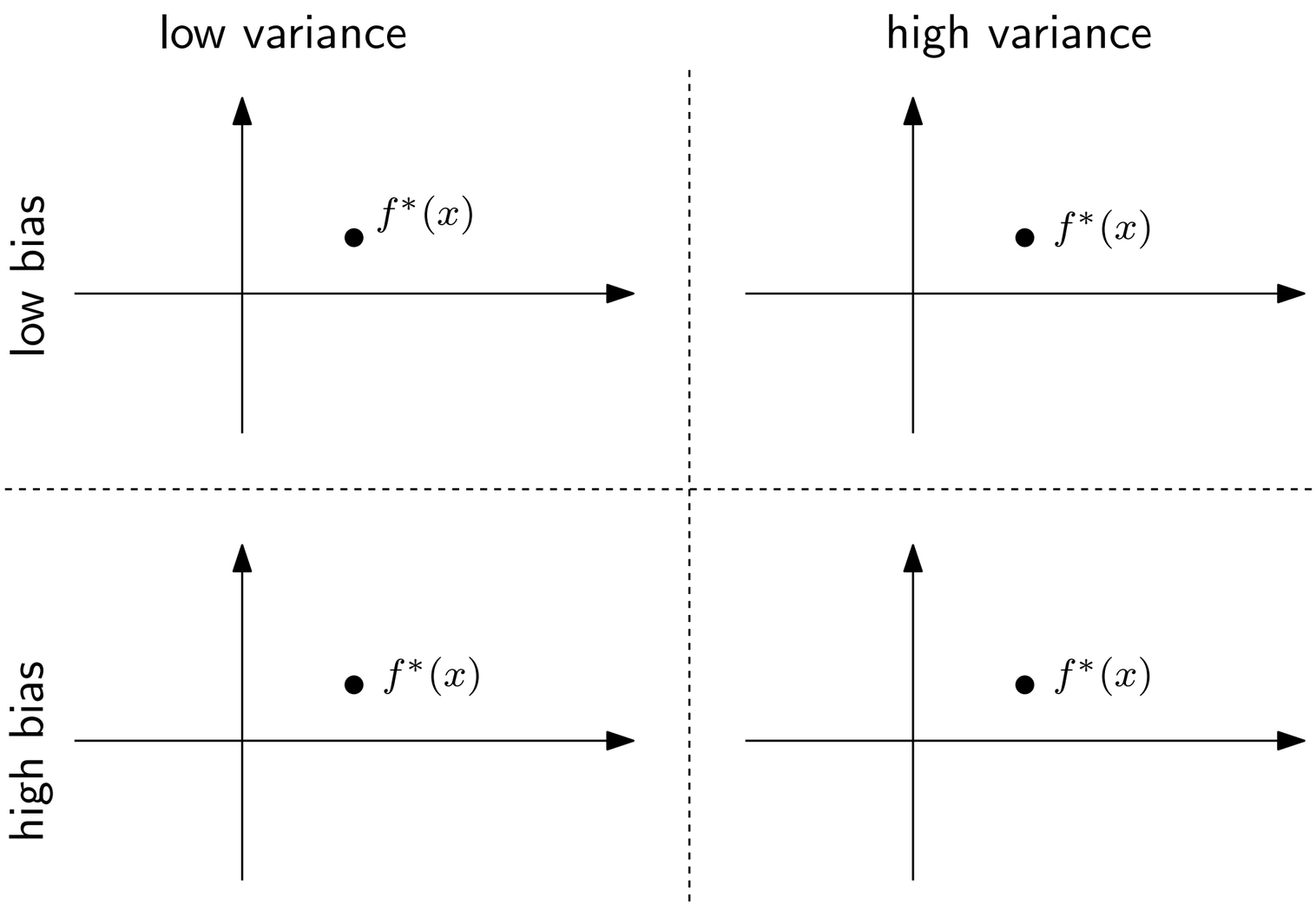
  $=$ Too flexible, need a more interpretable model

- Increase the regularity of the "hypothesis" or "prediction" h_w.

  In general setting, requires to bound ω

  $\longrightarrow$ Regularization

Overfitting: Statistical interpretation with Bias-variance tradeoff



Given a test data $x \in \mathbb{R}^p$ evaluate prediction $f_D(x) \approx f^*(x)$ as a random variable depending on the data set D

Depends on two quantities:

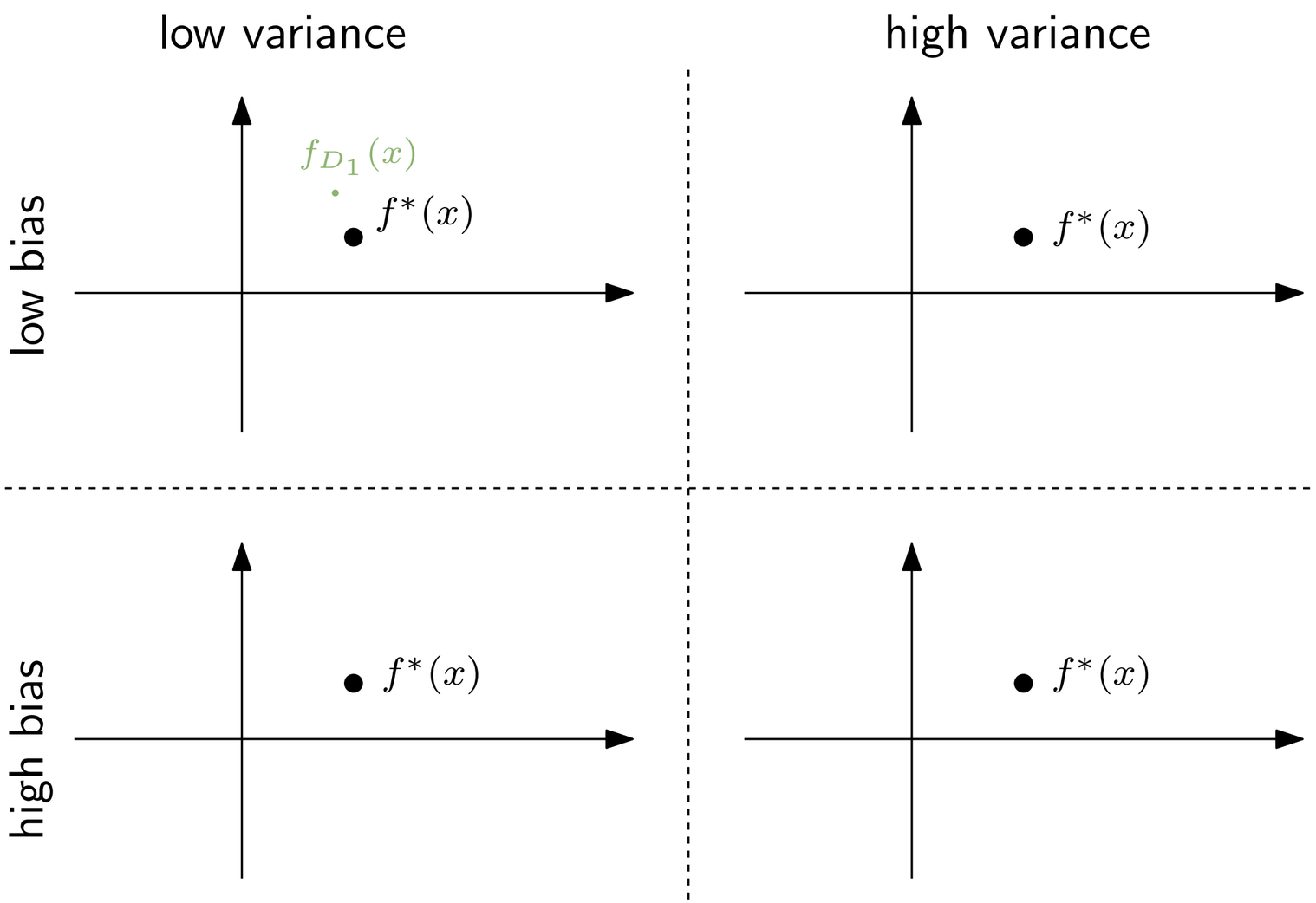
- **Bias:**

$$\mathbb{E}_D[f_D(X) - f^*(X)]$$

- **Variance:**

$$\mathbb{E}_D[(f_D(X) - \mathbb{E}_D[f_D(X)])^2]$$

Overfitting: Statistical interpretation with Bias-variance tradeoff



Given a test data $x \in \mathbb{R}^p$ evaluate prediction $f_D(x) \approx f^*(x)$ as a random variable depending on the data set D

Depends on two quantities:

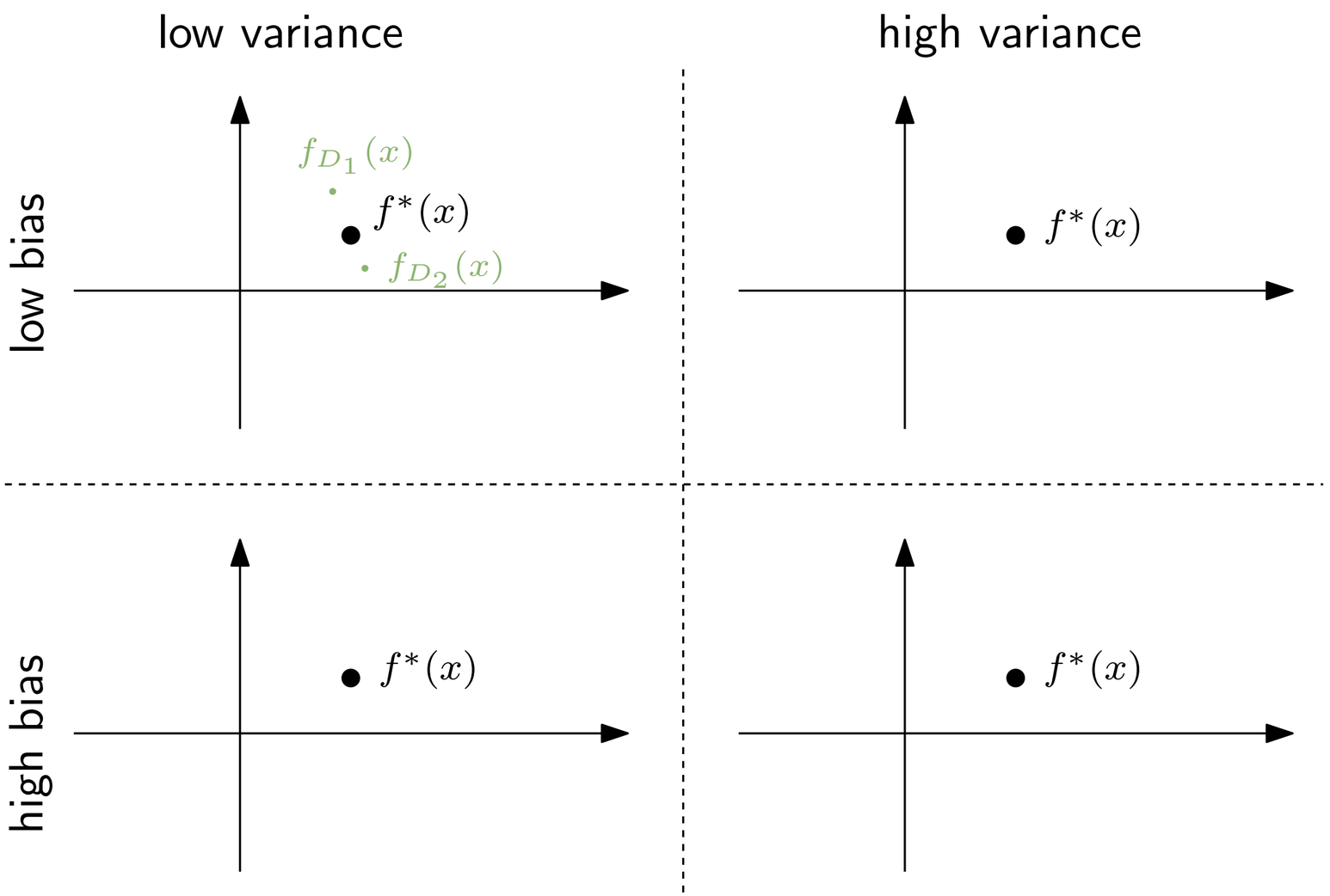
- **Bias:**

$$\mathbb{E}_D[f_D(X) - f^*(X)]$$

- **Variance:**

$$\mathbb{E}_D[(f_D(X) - \mathbb{E}_D[f_D(X)])^2]$$

Overfitting: Statistical interpretation with Bias-variance tradeoff



Given a test data $x \in \mathbb{R}^p$ evaluate prediction $f_D(x) \approx f^*(x)$ as a random variable depending on the data set D

Depends on two quantities:

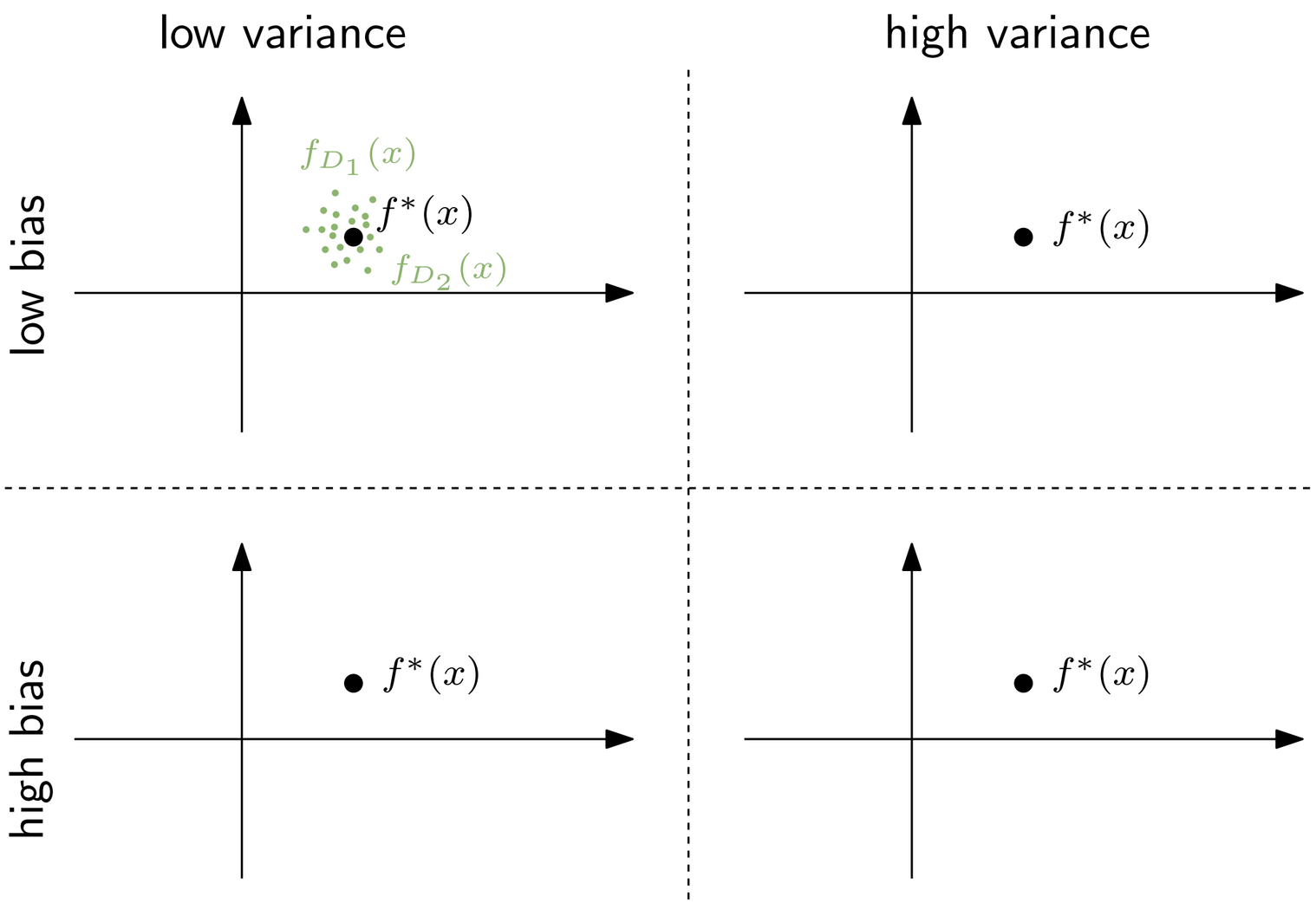
- **Bias:**

$$\mathbb{E}_D[f_D(X) - f^*(X)]$$

- **Variance:**

$$\mathbb{E}_D[(f_D(X) - \mathbb{E}_D[f_D(X)])^2]$$

Overfitting: Statistical interpretation with Bias-variance tradeoff



Given a test data $x \in \mathbb{R}^p$ evaluate prediction $f_D(x) \approx f^*(x)$ as a random variable depending on the data set D

Depends on two quantities:

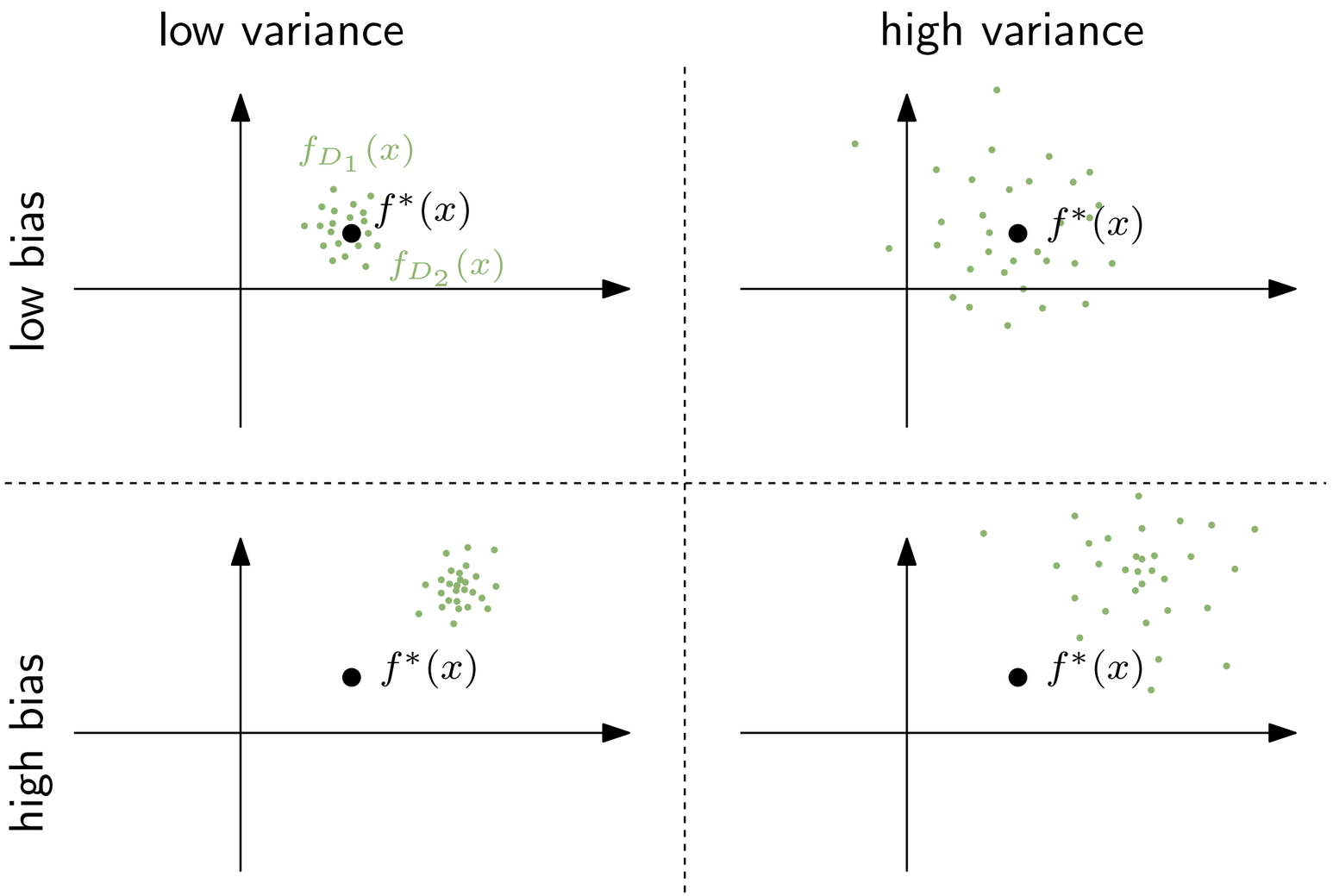
- **Bias:**

$$\mathbb{E}_D[f_D(X) - f^*(X)]$$

- **Variance:**

$$\mathbb{E}_D[(f_D(X) - \mathbb{E}_D[f_D(X)])^2]$$

Overfitting: Statistical interpretation with Bias-variance tradeoff



Given a test data $x \in \mathbb{R}^p$ evaluate prediction $f_D(x) \approx f^*(x)$ as a random variable depending on the data set D

Depends on two quantities:

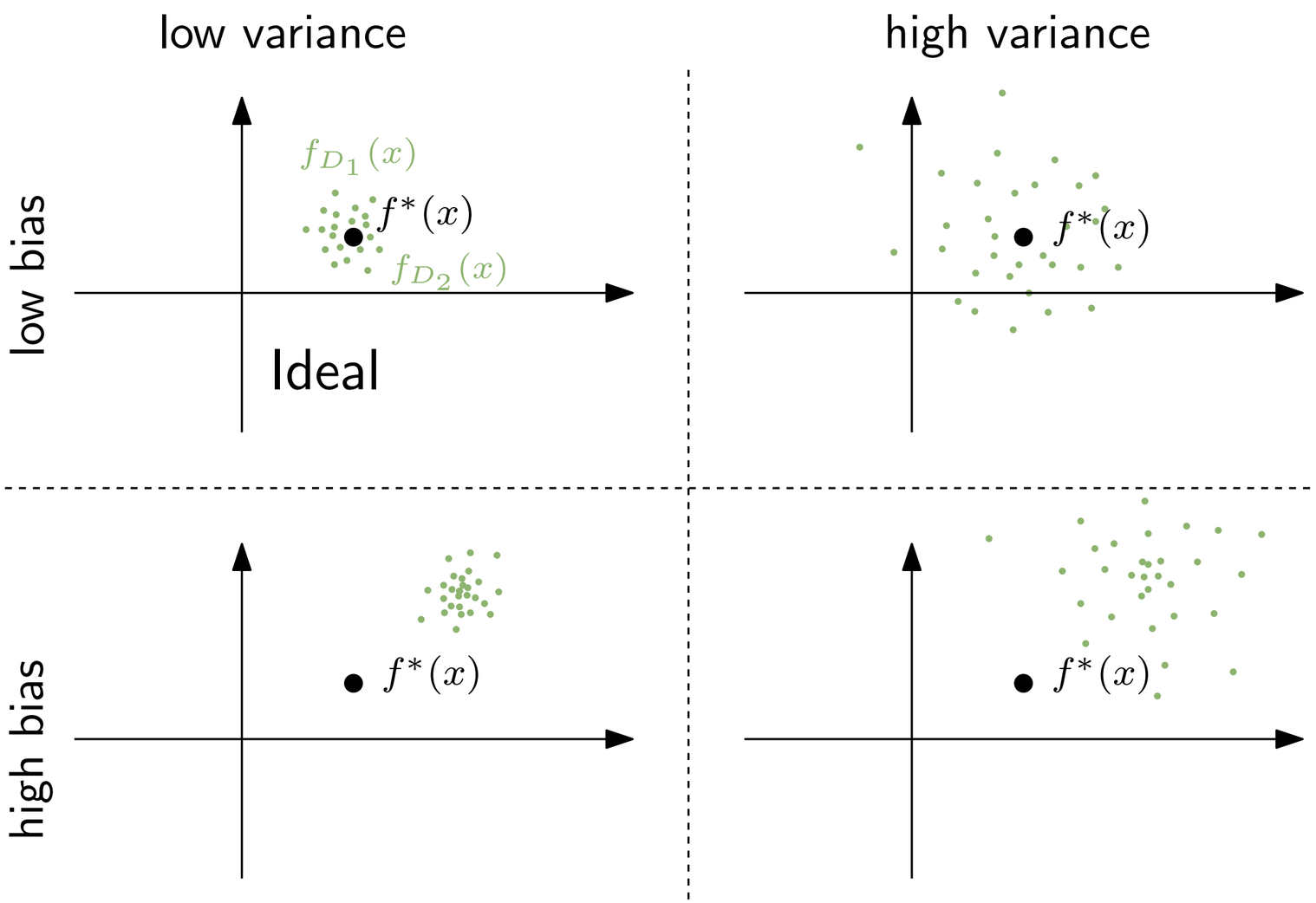
- **Bias:**

$$\mathbb{E}_D[f_D(X) - f^*(X)]$$

- **Variance:**

$$\mathbb{E}_D[(f_D(X) - \mathbb{E}_D[f_D(X)])^2]$$

Overfitting: Statistical interpretation with Bias-variance tradeoff



Given a test data $x \in \mathbb{R}^p$ evaluate prediction $f_D(x) \approx f^*(x)$ as a random variable depending on the data set D

Depends on two quantities:

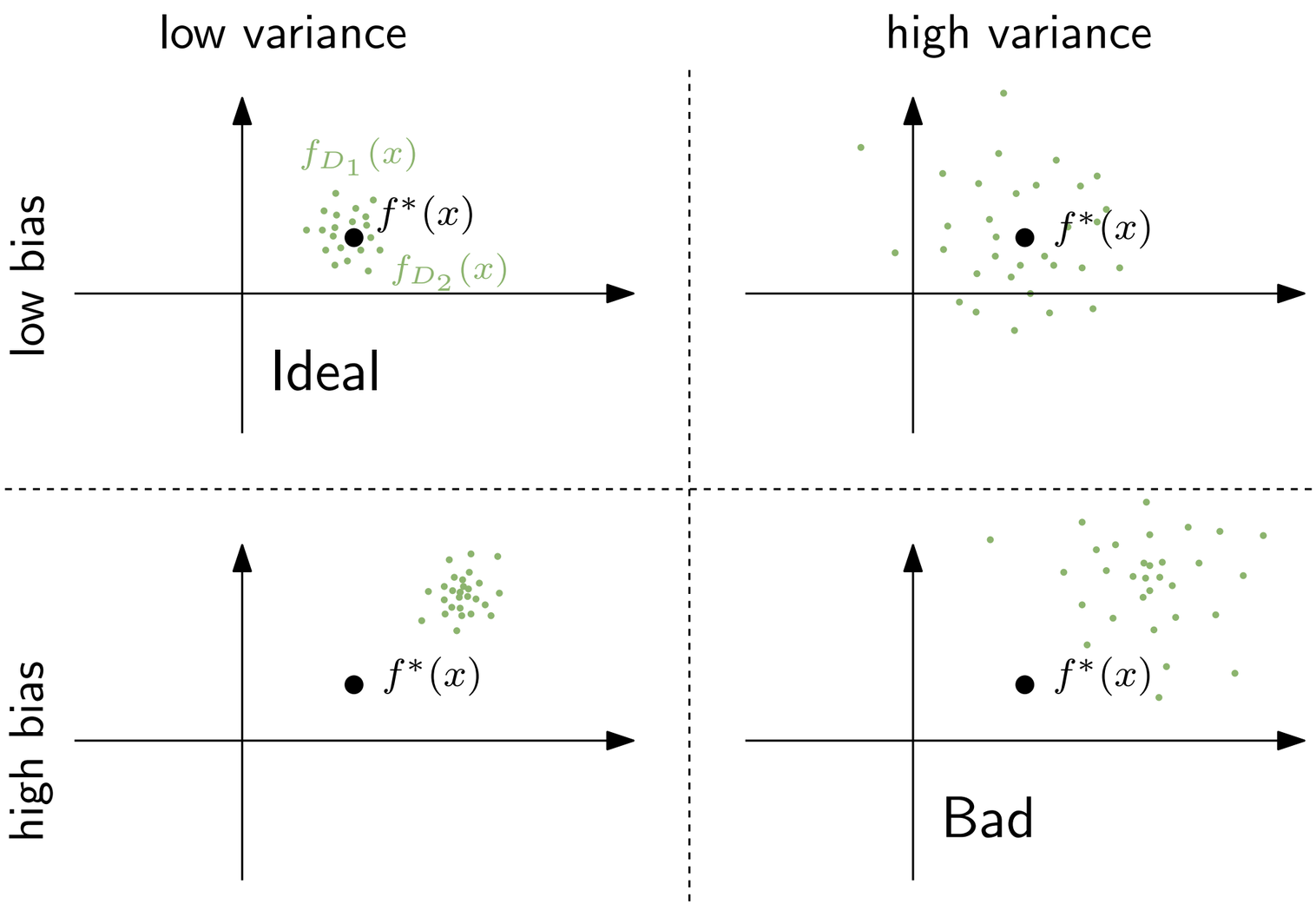
- **Bias:**

$$\mathbb{E}_D[f_D(X) - f^*(X)]$$

- **Variance:**

$$\mathbb{E}_D[(f_D(X) - \mathbb{E}_D[f_D(X)])^2]$$

Overfitting: Statistical interpretation with Bias-variance tradeoff



Given a test data $x \in \mathbb{R}^p$ evaluate prediction $f_D(x) \approx f^*(x)$ as a random variable depending on the data set D

Depends on two quantities:

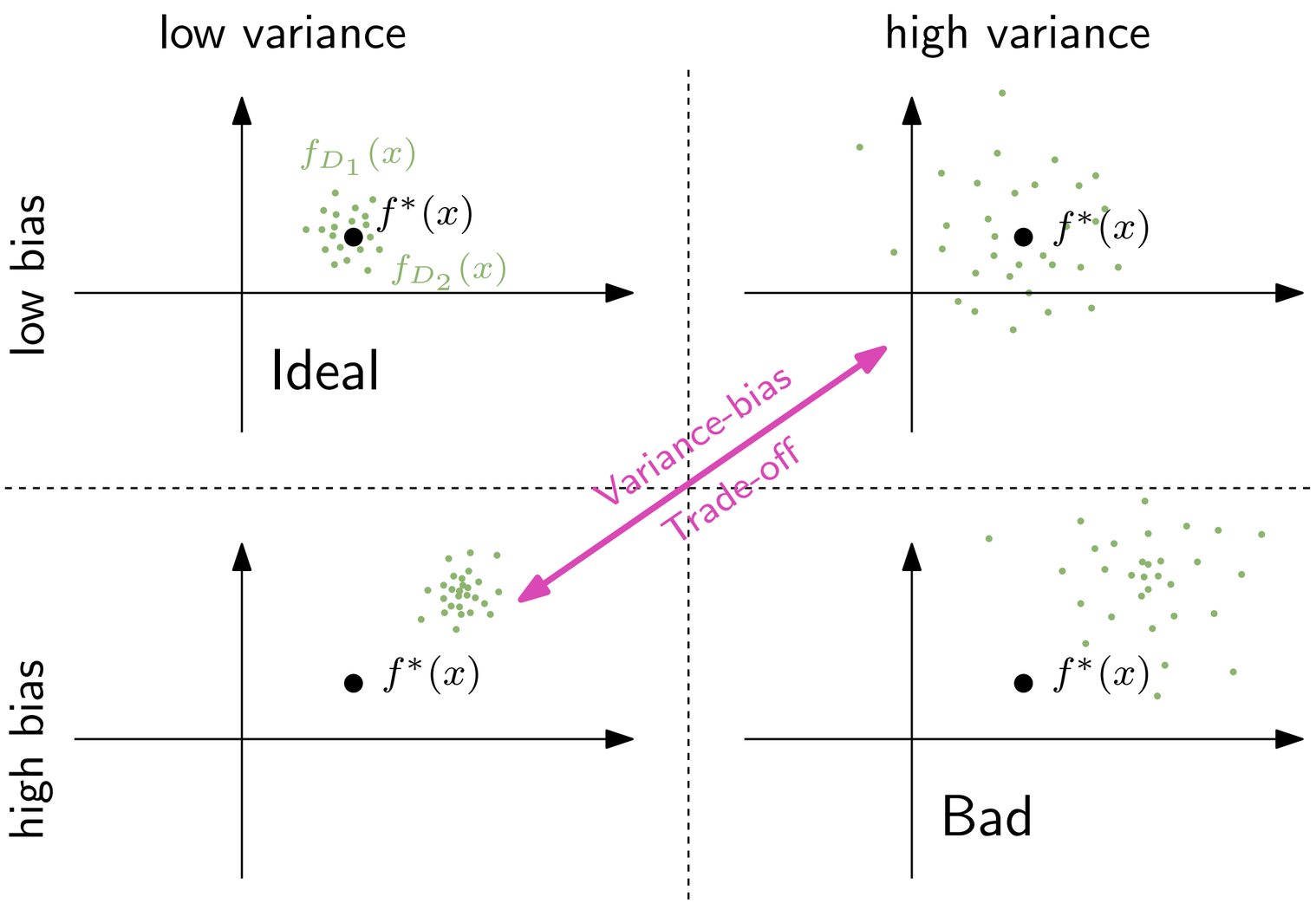
- **Bias:**

$$\mathbb{E}_D[f_D(X) - f^*(X)]$$

- **Variance:**

$$\mathbb{E}_D[(f_D(X) - \mathbb{E}_D[f_D(X)])^2]$$

Overfitting: Statistical interpretation with Bias-variance tradeoff



Given a test data $x \in \mathbb{R}^p$ evaluate prediction $f_D(x) \approx f^*(x)$ as a random variable depending on the data set D

Depends on two quantities:

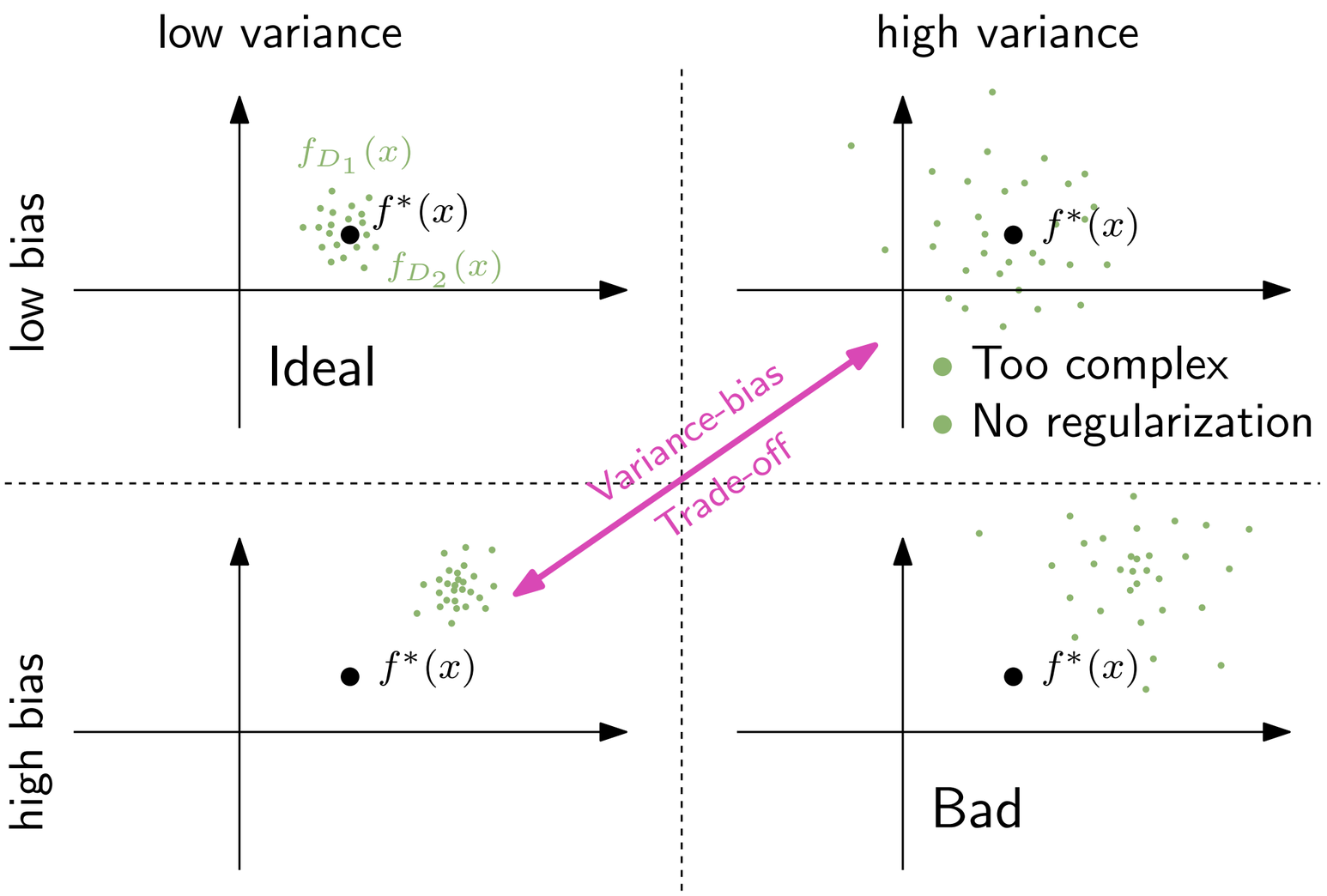
- **Bias:**

$$\mathbb{E}_D[f_D(X) - f^*(X)]$$

- **Variance:**

$$\mathbb{E}_D[(f_D(X) - \mathbb{E}_D[f_D(X)])^2]$$

Overfitting: Statistical interpretation with Bias-variance tradeoff



Given a test data $x \in \mathbb{R}^p$ evaluate prediction $f_D(x) \approx f^*(x)$ as a random variable depending on the data set D

Depends on two quantities:

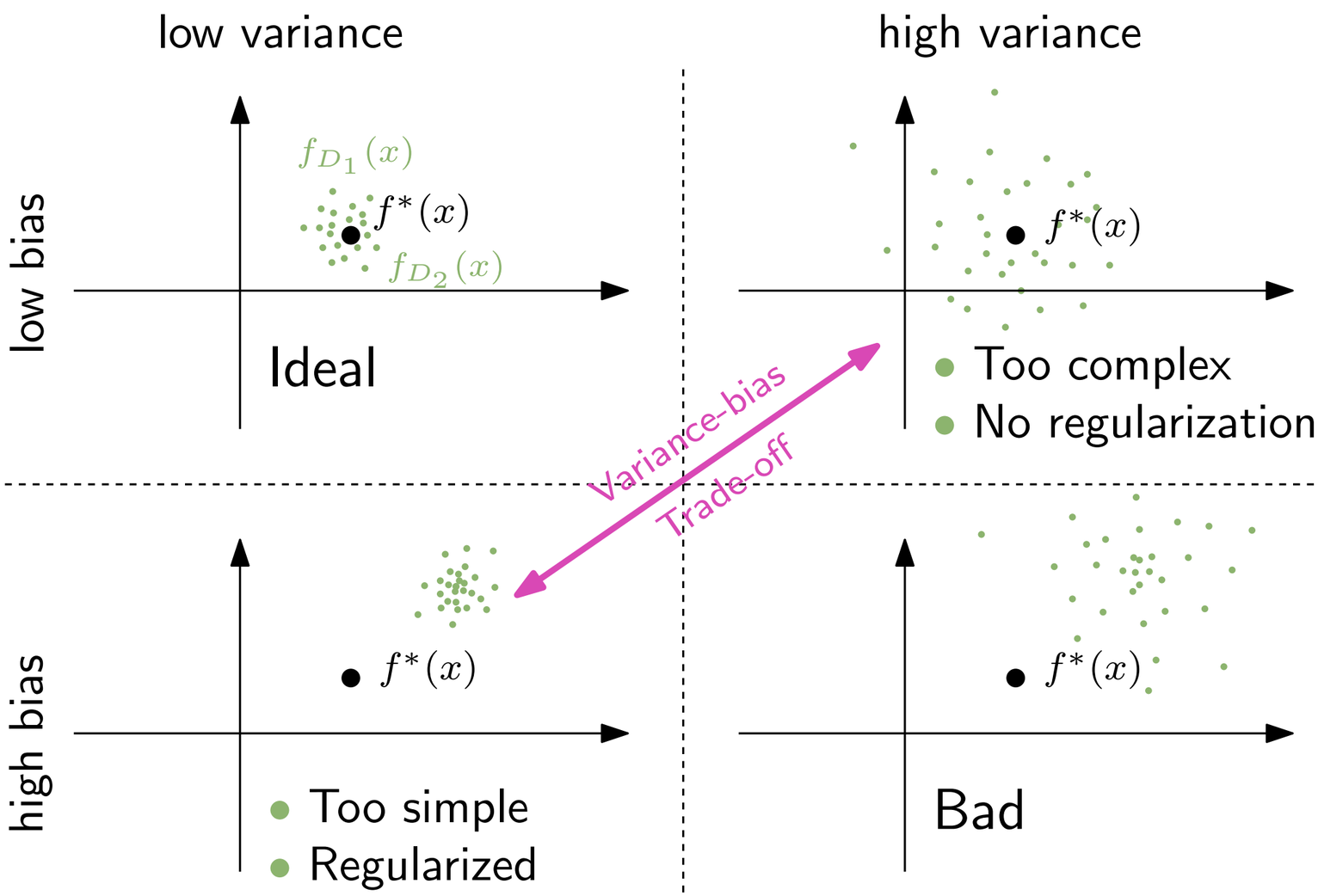
- **Bias:**

$$\mathbb{E}_D[f_D(X) - f^*(X)]$$

- **Variance:**

$$\mathbb{E}_D[(f_D(X) - \mathbb{E}_D[f_D(X)])^2]$$

Overfitting: Statistical interpretation with Bias-variance tradeoff



Given a test data $x \in \mathbb{R}^p$ evaluate prediction $f_D(x) \approx f^*(x)$ as a random variable depending on the data set D

Depends on two quantities:

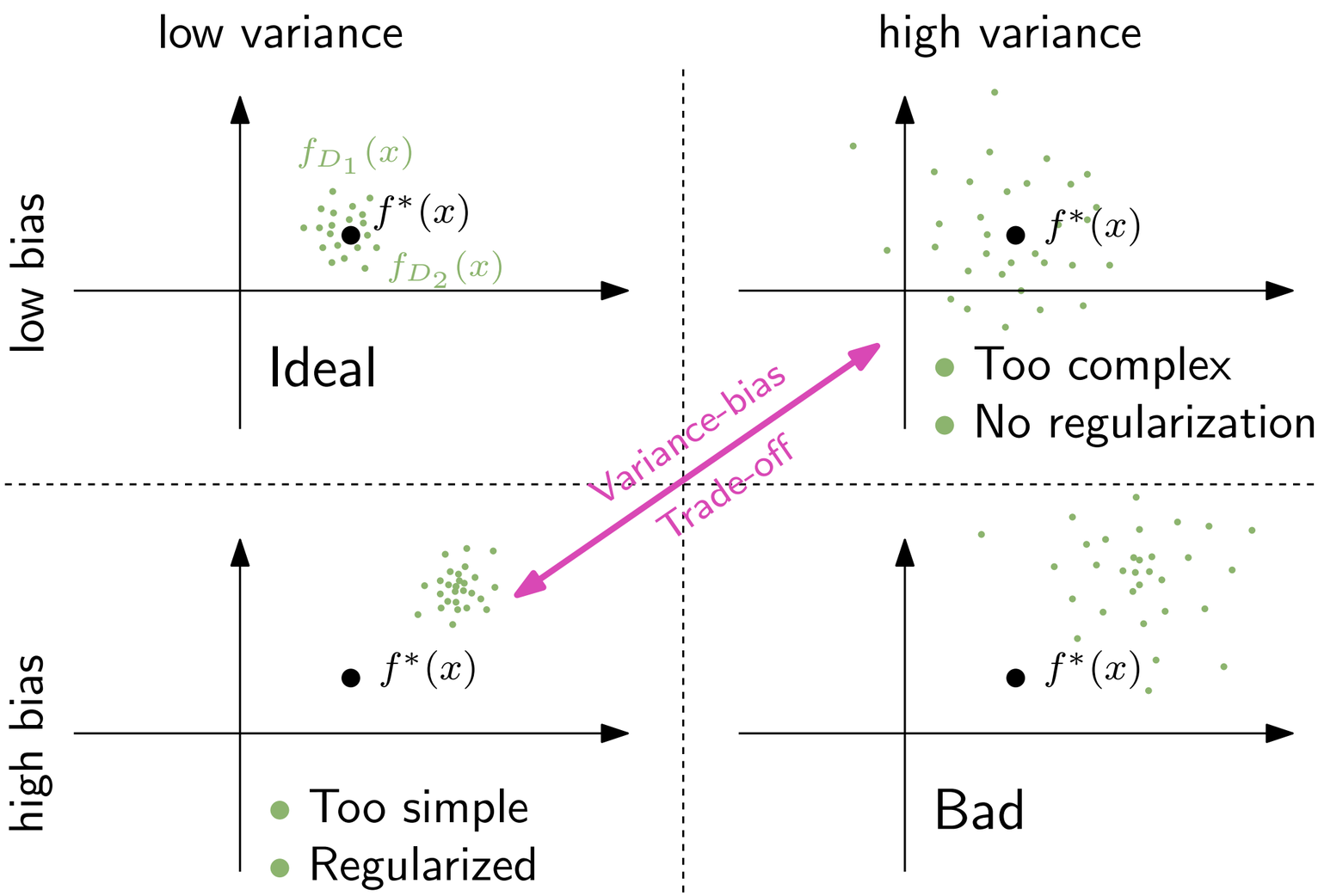
- **Bias:**

$$\mathbb{E}_D[f_D(X) - f^*(X)]$$

- **Variance:**

$$\mathbb{E}_D[(f_D(X) - \mathbb{E}_D[f_D(X)])^2]$$

Overfitting: Statistical interpretation with Bias-variance tradeoff



Given a test data $x \in \mathbb{R}^p$ evaluate prediction $f_D(x) \approx f^*(x)$ as a random variable depending on the data set D

Depends on two quantities:

- **Bias:**

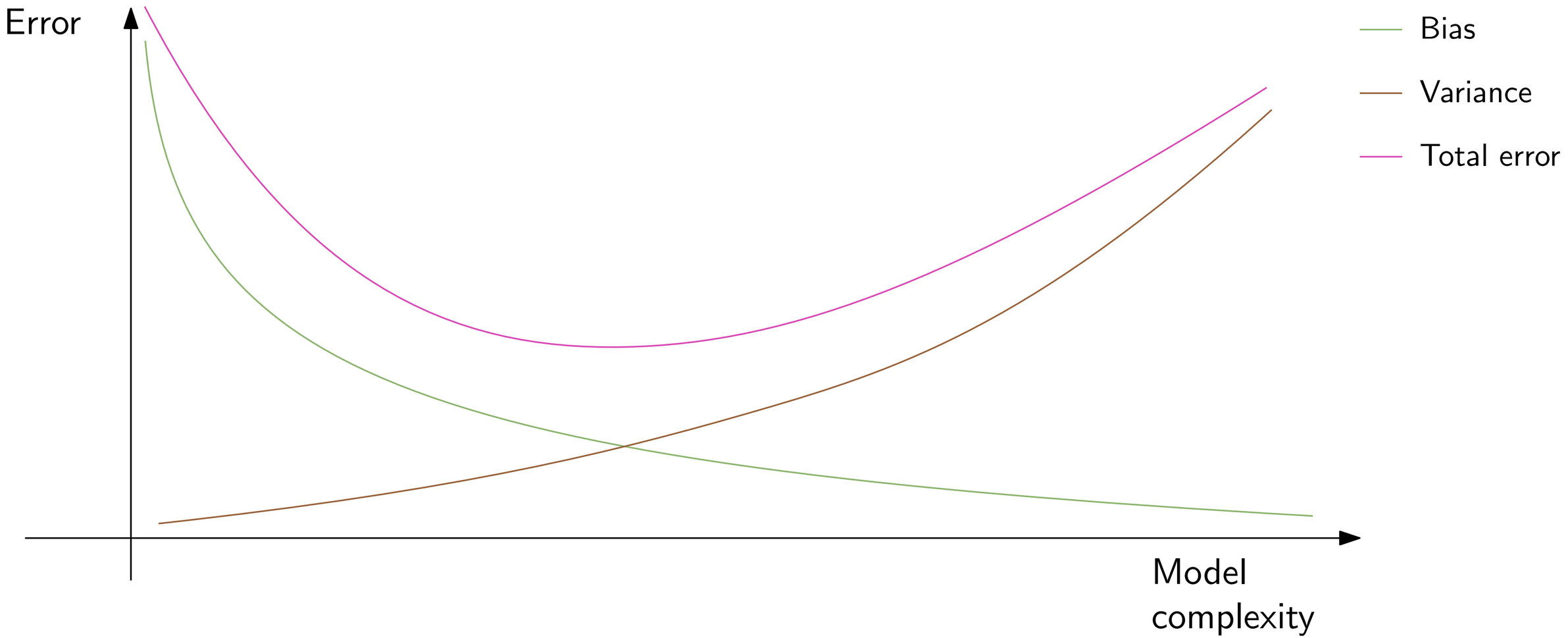
$$\mathbb{E}_D[f_D(X) - f^*(X)]$$

- **Variance:**

$$\mathbb{E}_D[(f_D(X) - \mathbb{E}_D[f_D(X)])^2]$$

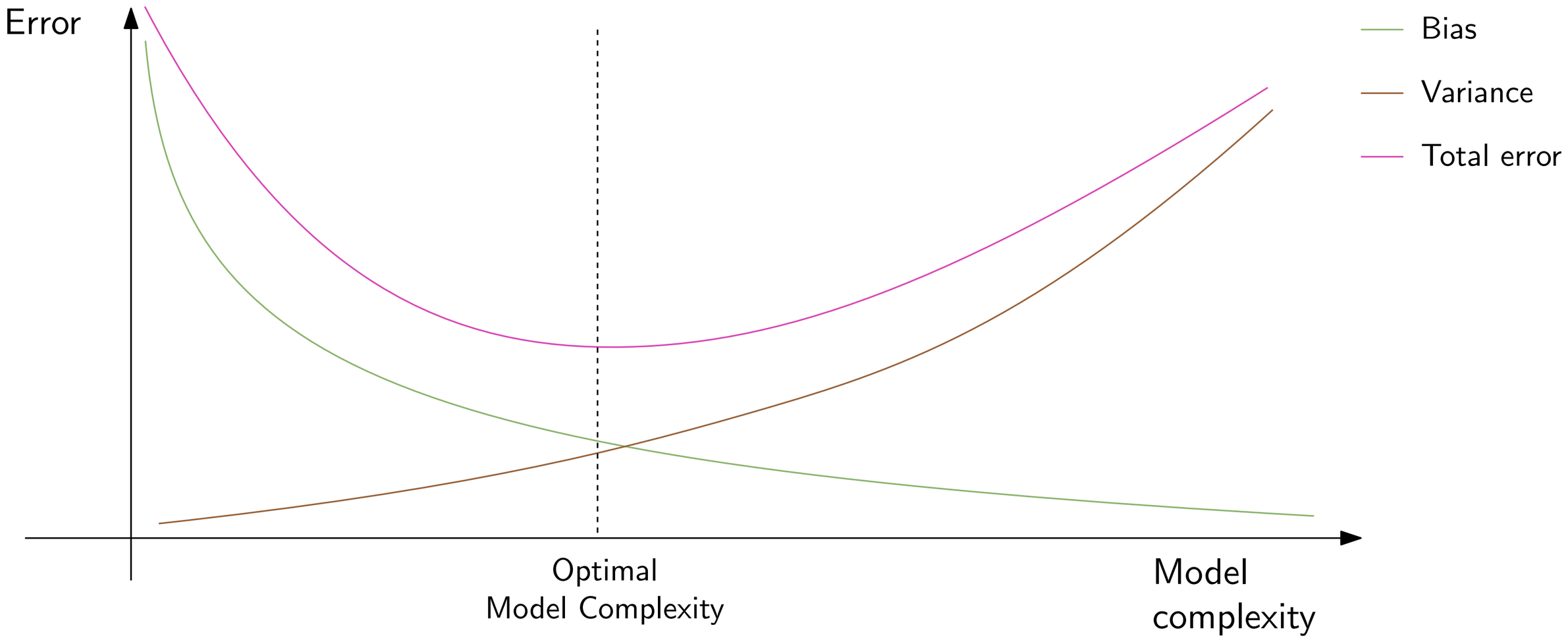
Overfitting: Statistical interpretation with Bias-variance tradeoff

A classical behavior:



Overfitting: Statistical interpretation with Bias-variance tradeoff

A classical behavior:



Overfitting: Statistical interpretation with Bias-variance tradeoff

Model: $Y = f^*(X) + \varepsilon$, with f^* deterministic and $\mathbb{E}[\varepsilon] = 0$

Given a data set $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, compare $f_D(X)$ with Y:

$$\mathbb{E}_{D,\varepsilon}[\|f_D(x) - Y\|^2] = \mathbb{E}_{D,\varepsilon}[\|f_D(x) - f^*(x) + f^*(x) - Y\|^2]$$

Overfitting: Statistical interpretation with Bias-variance tradeoff

Model: $Y = f^*(X) + \varepsilon$, with f^* determinist and $\mathbb{E}[\varepsilon] = 0$

Given a data set $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$, compare $f_D(X)$ with Y:

$$\mathbb{E}_{D,\varepsilon}[\|f_D(x) - Y\|^2] = \mathbb{E}_{D,\varepsilon}[\|f_D(x) - f^*(x) + f^*(x) - Y\|^2]$$

Lemma: Given two **independent** random vectors $X, Y \in \mathbb{R}^p$ such that $\mathbb{E}[X] = 0$ or $\mathbb{E}[Y] = 0$:

$$\mathbb{E}[\|X - Y\|^2] = \mathbb{E}[\|X\|^2] + \mathbb{E}[\|Y\|^2].$$

Proof: $\mathbb{E}[\|X - Y\|^2] = \mathbb{E}[\|X\|^2] + 2\mathbb{E}[X^T Y] + \mathbb{E}[\|Y\|^2] = \mathbb{E}[\|X\|^2] + \mathbb{E}[\|Y\|^2] + 2\mathbb{E}[X]^T \mathbb{E}[Y]$

Overfitting: Statistical interpretation with Bias-variance tradeoff

Model: $Y = f^*(X) + \varepsilon$, with f^* deterministe and $\mathbb{E}[\varepsilon] = 0$

Given a data set $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$, compare $f_D(X)$ with Y:

Independent

$$\mathbb{E}_{D,\varepsilon}[\|f_D(x) - Y\|^2] = \mathbb{E}_{D,\varepsilon}[\|f_D(x) - f^*(x) + \underbrace{f^*(x) - Y}_{\mathbb{E}_\varepsilon[\dots]=\mathbb{E}[\varepsilon]=0}\|^2]$$

Lemma: Given two **independent** random vectors $X, Y \in \mathbb{R}^p$ such that $\mathbb{E}[X] = 0$ or $\mathbb{E}[Y] = 0$:

$$\mathbb{E}[\|X - Y\|^2] = \mathbb{E}[\|X\|^2] + \mathbb{E}[\|Y\|^2].$$

Proof: $\mathbb{E}[\|X - Y\|^2] = \mathbb{E}[\|X\|^2] + 2\mathbb{E}[X^T Y] + \mathbb{E}[\|Y\|^2] = \mathbb{E}[\|X\|^2] + \mathbb{E}[\|Y\|^2] + 2\mathbb{E}[X]^T\mathbb{E}[Y]$

Overfitting: Statistical interpretation with Bias-variance tradeoff

Model: $Y = f^*(X) + \varepsilon$, with f^* deterministe and $\mathbb{E}[\varepsilon] = 0$

Given a data set $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$, compare $f_D(X)$ with Y:

$$\begin{aligned}\mathbb{E}_{D,\varepsilon}[\|f_D(x) - Y\|^2] &= \mathbb{E}_{D,\varepsilon}[\|f_D(x) - f^*(x) + f^*(x) - Y\|^2] \\ &= \mathbb{E}_{D,\varepsilon}[\|f_D(x) - f^*(x)\|^2] + \mathbb{E}_{D,\varepsilon}[\|f^*(x) - Y\|^2]\end{aligned}$$

Lemma: Given two **independent** random vectors $X, Y \in \mathbb{R}^p$ such that $\mathbb{E}[X] = 0$ or $\mathbb{E}[Y] = 0$:

$$\mathbb{E}[\|X - Y\|^2] = \mathbb{E}[\|X\|^2] + \mathbb{E}[\|Y\|^2].$$

Proof: $\mathbb{E}[\|X - Y\|^2] = \mathbb{E}[\|X\|^2] + 2\mathbb{E}[X^T Y] + \mathbb{E}[\|Y\|^2] = \mathbb{E}[\|X\|^2] + \mathbb{E}[\|Y\|^2] + 2\mathbb{E}[X]^T \mathbb{E}[Y]$

Overfitting: Statistical interpretation with Bias-variance tradeoff

Model: $Y = f^*(X) + \varepsilon$, with f^* deterministic and $\mathbb{E}[\varepsilon] = 0$

Given a data set $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$, compare $f_D(X)$ with Y:

$$\begin{aligned}
\mathbb{E}_{D,\varepsilon}[\|f_D(x) - Y\|^2] &= \mathbb{E}_{D,\varepsilon}[\|f_D(x) - f^*(x) + f^*(x) - Y\|^2] \\
&= \mathbb{E}_{D,\varepsilon}[\|f_D(x) - f^*(x)\|^2] + \mathbb{E}_{D,\varepsilon}[\|f^*(x) - Y\|^2] \\
&= \mathbb{E}_{D,\varepsilon}[\|f_D(x) - \mathbb{E}_D[f_D(x)] + \mathbb{E}_D[f_D(x)] - f^*(x)\|^2] + \mathbb{E}_\varepsilon[\|\varepsilon\|^2]
\end{aligned}$$

Lemma: Given two **independent** random vectors $X, Y \in \mathbb{R}^p$ such that $\mathbb{E}[X] = 0$ or $\mathbb{E}[Y] = 0$:

$$\mathbb{E}[\|X - Y\|^2] = \mathbb{E}[\|X\|^2] + \mathbb{E}[\|Y\|^2].$$

Proof: $\mathbb{E}[\|X - Y\|^2] = \mathbb{E}[\|X\|^2] + 2\mathbb{E}[X^T Y] + \mathbb{E}[\|Y\|^2] = \mathbb{E}[\|X\|^2] + \mathbb{E}[\|Y\|^2] + 2\mathbb{E}[X]^T \mathbb{E}[Y]$

Overfitting: Statistical interpretation with Bias-variance tradeoff

Model: $Y = f^*(X) + \varepsilon$, with f^* deterministist and $\mathbb{E}[\varepsilon] = 0$

Given a data set $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$, compare $f_D(X)$ with Y:

$$\begin{aligned}
\mathbb{E}_{D,\varepsilon}[\|f_D(x) - Y\|^2] &= \mathbb{E}_{D,\varepsilon}[\|f_D(x) - f^*(x) + f^*(x) - Y\|^2] \\
&= \mathbb{E}_{D,\varepsilon}[\|f_D(x) - f^*(x)\|^2] + \mathbb{E}_{D,\varepsilon}[\|f^*(x) - Y\|^2] \\
&= \mathbb{E}_{D,\varepsilon}[\| \underbrace{f_D(x) - \mathbb{E}_D[f_D(x)]}_{\mathbb{E}_D[\dots]=0} + \mathbb{E}_D[f_D(x)] - f^*(x)\|^2] + \mathbb{E}_\varepsilon[\|\varepsilon\|^2]
\end{aligned}$$

Independent

Lemma: Given two **independent** random vectors $X, Y \in \mathbb{R}^p$ such that $\mathbb{E}[X] = 0$ or $\mathbb{E}[Y] = 0$:

$$\mathbb{E}[\|X - Y\|^2] = \mathbb{E}[\|X\|^2] + \mathbb{E}[\|Y\|^2].$$

Proof: $\mathbb{E}[\|X - Y\|^2] = \mathbb{E}[\|X\|^2] + 2\mathbb{E}[X^T Y] + \mathbb{E}[\|Y\|^2] = \mathbb{E}[\|X\|^2] + \mathbb{E}[\|Y\|^2] + 2\mathbb{E}[X]^T\mathbb{E}[Y]$

Overfitting: Statistical interpretation with Bias-variance tradeoff

Model: $Y = f^*(X) + \varepsilon$, with f^* determinist and $\mathbb{E}[\varepsilon] = 0$

Given a data set $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$, compare $f_D(X)$ with Y:

$$\begin{aligned}
\mathbb{E}_{D,\varepsilon}[\|f_D(x) - Y\|^2] &= \mathbb{E}_{D,\varepsilon}[\|f_D(x) - f^*(x) + f^*(x) - Y\|^2] \\
&= \mathbb{E}_{D,\varepsilon}[\|f_D(x) - f^*(x)\|^2] + \mathbb{E}_{D,\varepsilon}[\|f^*(x) - Y\|^2] \\
&= \mathbb{E}_{D,\varepsilon}[\|f_D(x) - \mathbb{E}_D[f_D(x)] + \mathbb{E}_D[f_D(x)] - f^*(x)\|^2] + \mathbb{E}_\varepsilon[\|\varepsilon\|^2] \\
&= \mathbb{E}_{D,\varepsilon}[\|f_D(x) - \mathbb{E}_D[f_D(x)]\|^2] + \mathbb{E}_{D,\varepsilon}[\|\mathbb{E}_D[f_D(x)] - f^*(x)\|^2] + \mathbb{E}_\varepsilon[\|\varepsilon\|^2]
\end{aligned}$$

Lemma: Given two **independent** random vectors $X, Y \in \mathbb{R}^p$ such that $\mathbb{E}[X] = 0$ or $\mathbb{E}[Y] = 0$:

$$\mathbb{E}[\|X - Y\|^2] = \mathbb{E}[\|X\|^2] + \mathbb{E}[\|Y\|^2].$$

Proof: $\mathbb{E}[\|X - Y\|^2] = \mathbb{E}[\|X\|^2] + 2\mathbb{E}[X^T Y] + \mathbb{E}[\|Y\|^2] = \mathbb{E}[\|X\|^2] + \mathbb{E}[\|Y\|^2] + 2\mathbb{E}[X]^T \mathbb{E}[Y]$

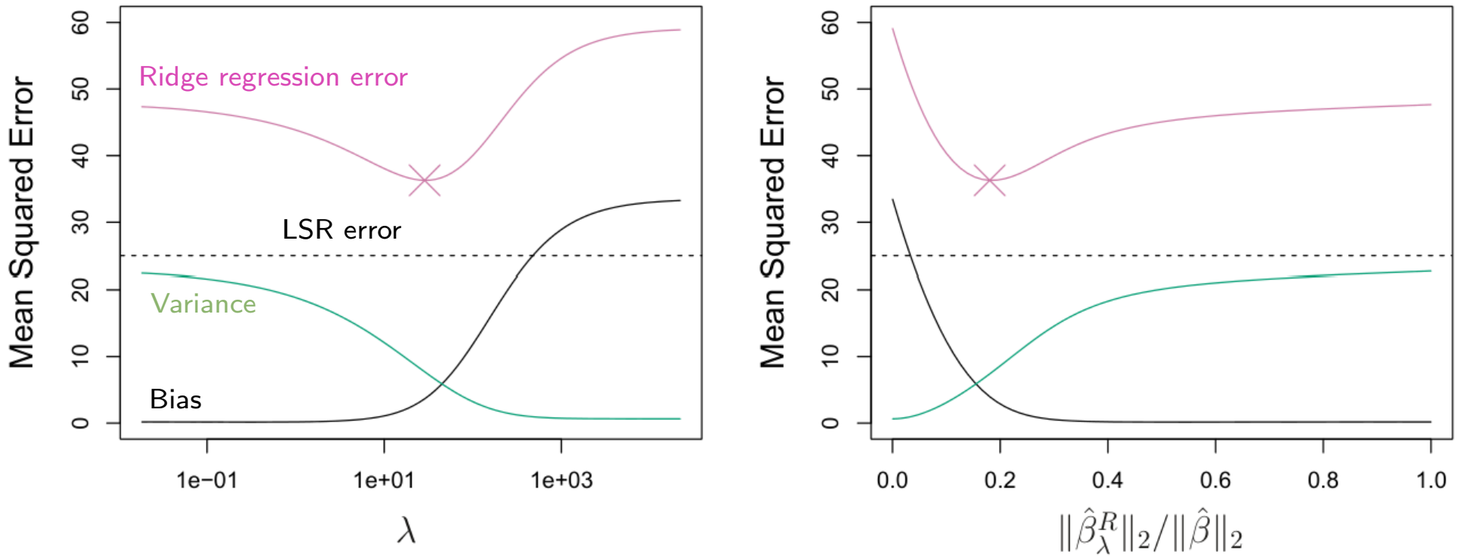
Overfitting: Statistical interpretation with Bias-variance tradeoff

Model: $Y = f^*(X) + \varepsilon$

Goal: Given a data set D, find the best prediction f_D: $f_D(X) \approx Y$.

Error: $\mathbb{E}_{D,\varepsilon}[(f_D(X) - Y)^2] = \underbrace{(\mathbb{E}_D[f_D(X) - f^*(X)])^2}_{\text{“Bias”}} + \underbrace{\mathbb{E}_D[(f_D(X) - \mathbb{E}_D[f_D(X)])^2]}_{\text{“Variance”}} + \mathbb{E}_\varepsilon[\epsilon^2]$

Comparison of Ridge regression and LSR depending on λ



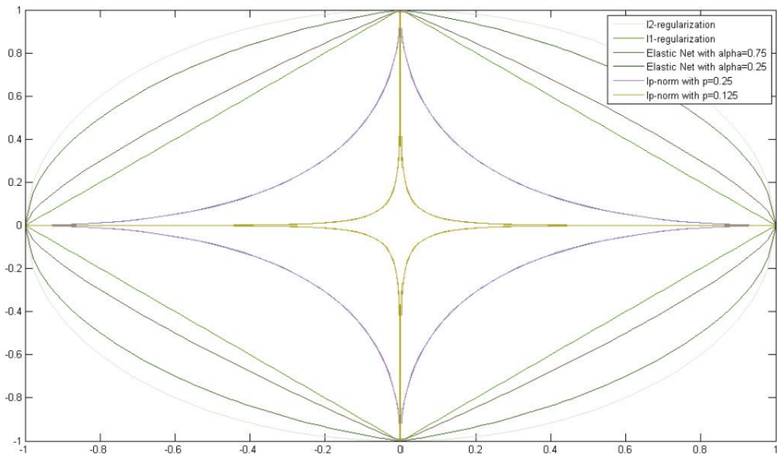
In linear regression look for $f(X) = \beta^T X$:

Least square regression:
Min $\frac{1}{n}\sum_{i=1}^n \|\beta^T x_i - y_i\|^2$

Ridge regression:
Min $\frac{1}{n}\sum_{i=1}^n \|\beta^T x_i - y_i\|^2 + \lambda\|\beta\|_2^2$

Regularization

Example with regression task: Minimize $\frac{1}{n}\sum_{i=1}^{n} l(h_w(x_i) - y_i) + \lambda r(w) \qquad w \in \mathcal{R}^q$



$r(z) = 1$ for different reguarization

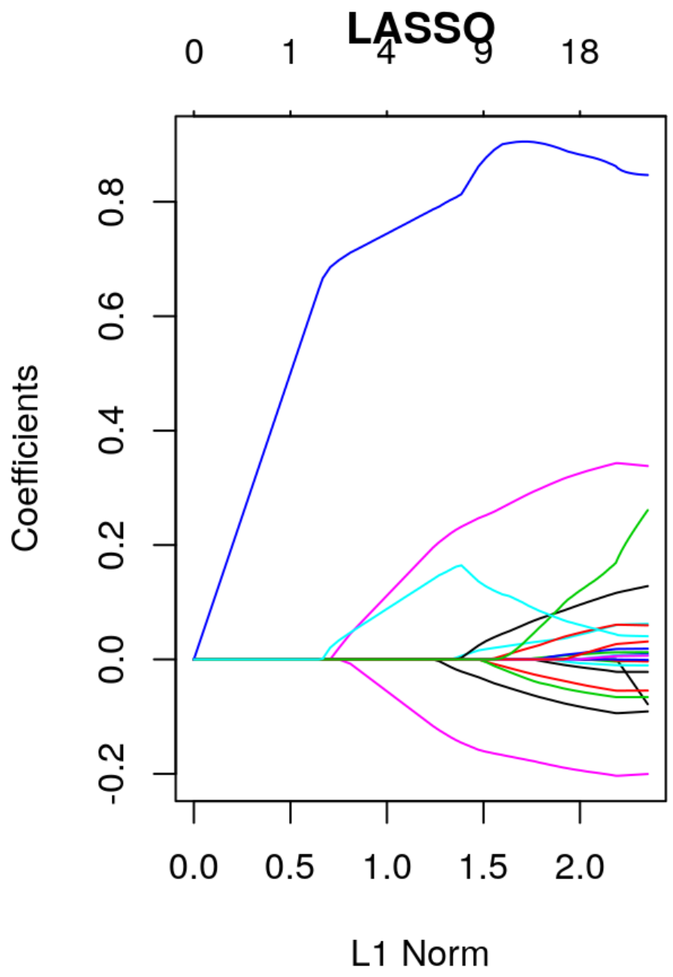
ℓ_2 **regularization:** $r(w) = \|w\|^2 = \|w\|_2^2 = \sum_{i=1}^{n} w_i^2$
Strictly Convex, Differentiable but "Dense Solution" (relies on all features to some degree).

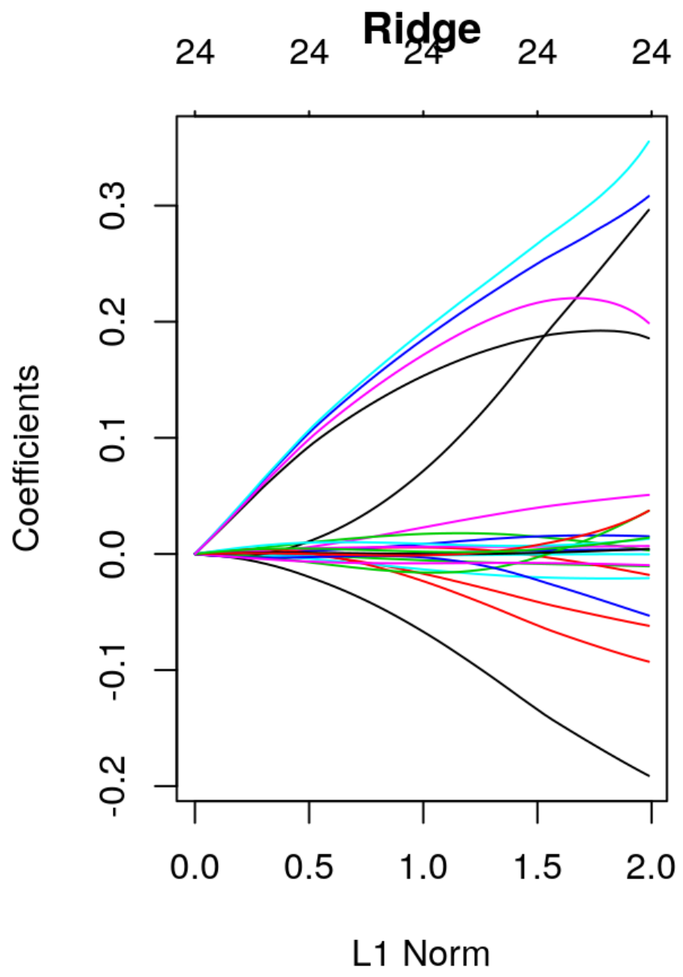
ℓ_1 **regularization:** $r(w) = \|w\|_1 = \sum_{i=1}^{q} |w_i|$
Convex (but not strictly), not differentiable at 0 (the point which minimization is intended to bring us to).
Selects the features "Sparse Solutions".

ℓ_p **regularization:** $r(w) = \|w\|_p = (\sum_{i=1}^{n} |w_i|^p)^{\frac{1}{p}}$, $0 \leq p \leq 1$
Rarely used: very sparse, initialisation dependent (during minimization procedure), non convex...

Elastic regularization: $r(z) = \alpha\|z\|_2^2 + \beta\|z\|_1$

ℓ_2 vs. ℓ_1 regularization





Ridge regression:
Min $\frac{1}{n}\sum_{i=1}^{n} \|w^T x_i - y_i\|^2 + \|w\|_2^2$

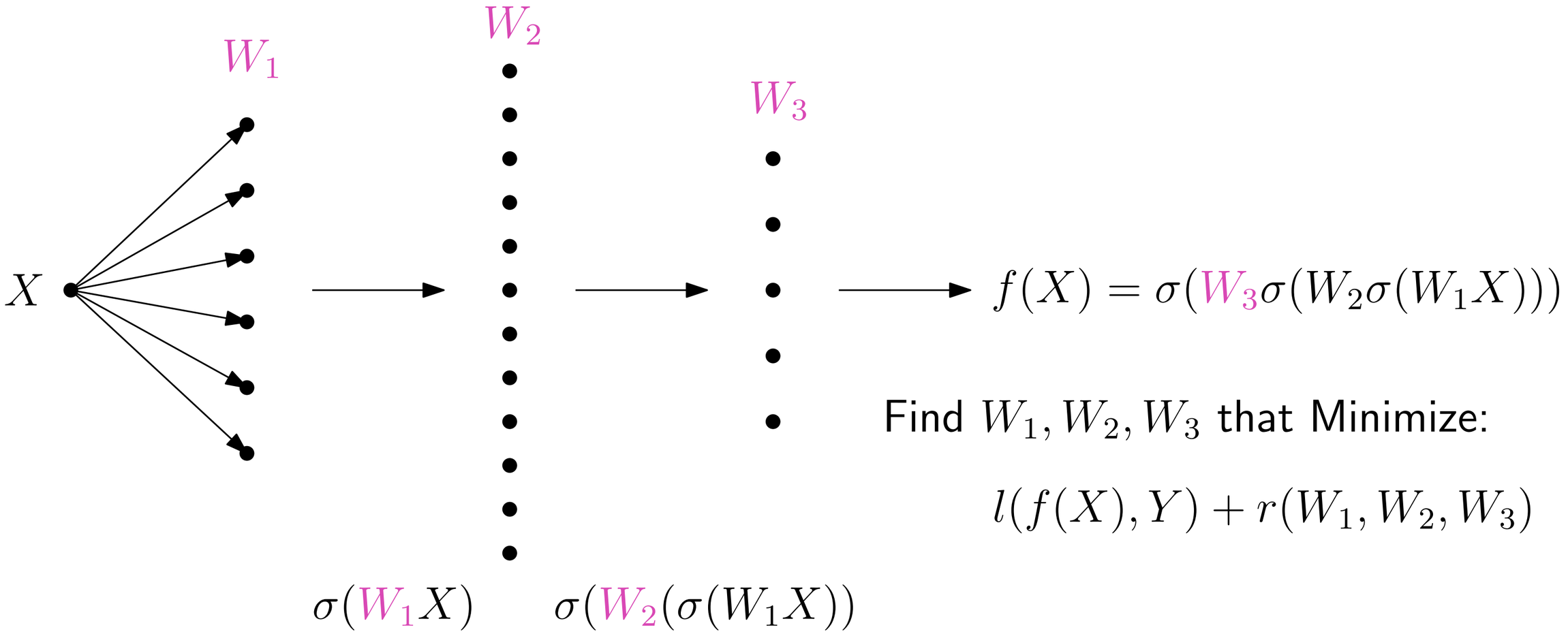
Lasso:
Min $\frac{1}{n}\sum_{i=1}^{n} \|w^T x_i - y_i\|^2 + \|w\|_1$

$\rightarrow$ Ridge collects the contributions of most of the predictors

$\rightarrow$ Lasso selects the most important coefficients

Deal with Overfitting for Neural-network

$f(X) = \sigma(W_l \sigma(W_{l-1} \cdots \sigma(W_1 X) \cdots)$ (Sequence of linear and non-linear transformations)

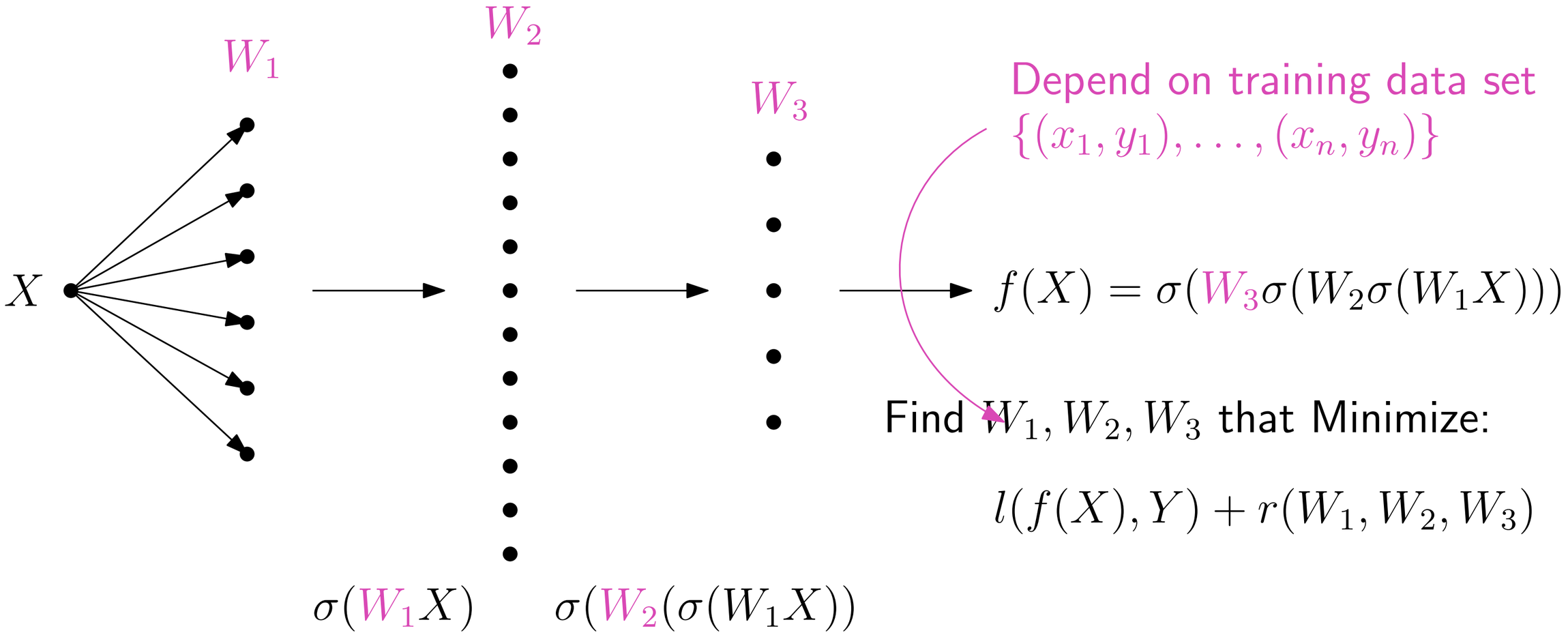


Find W_1, W_2, W_3 that Minimize:

$$l(f(X), Y) + r(W_1, W_2, W_3)$$

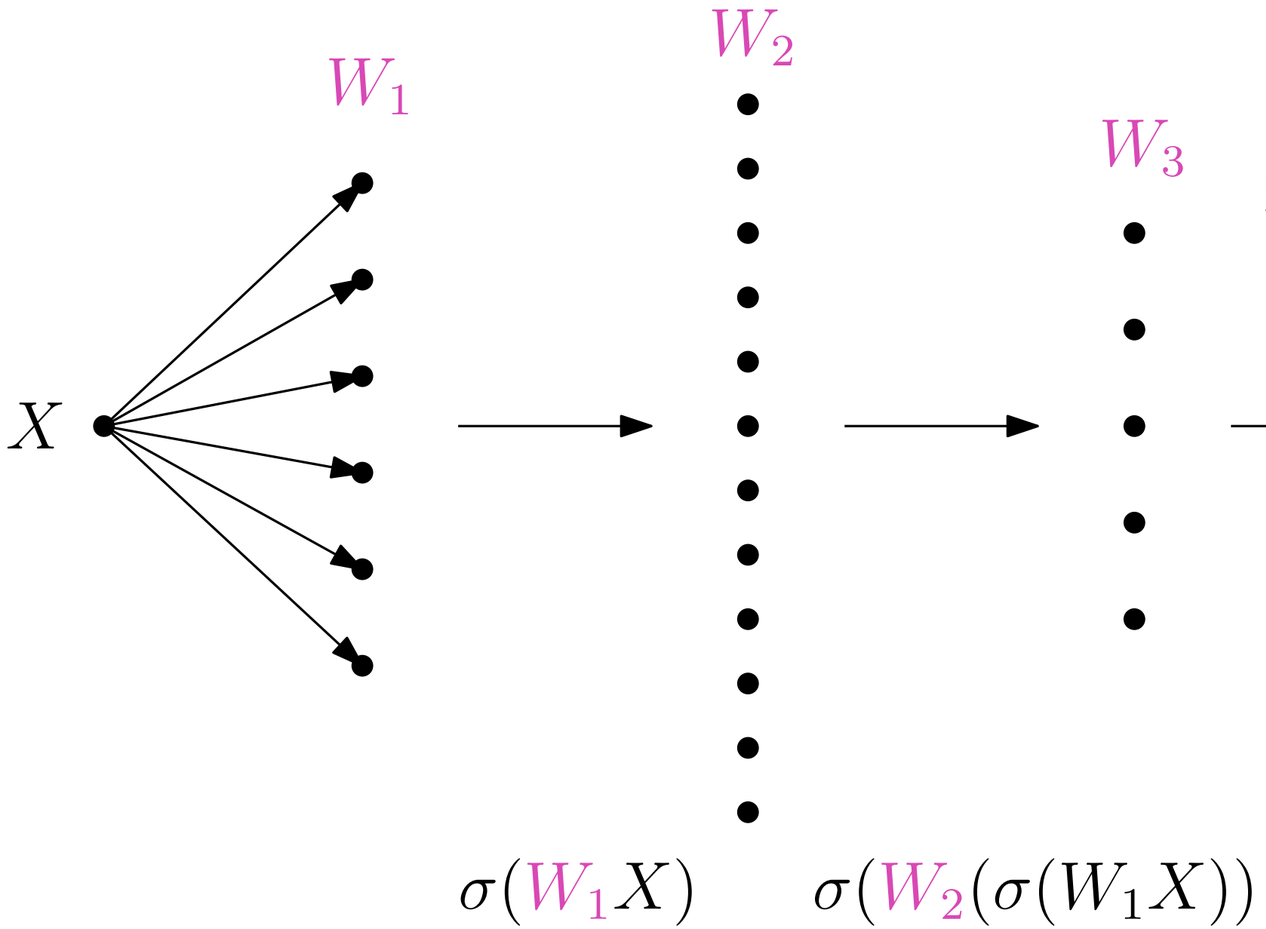
Deal with Overfitting for Neural-network

$f(X) = \sigma(W_l \sigma(W_{l-1} \cdots \sigma(W_1 X) \cdots)$ (Sequence of linear and non-linear transformations)



Deal with Overfitting for Neural-network

$f(X) = \sigma(W_l \sigma(W_{l-1} \cdots \sigma(W_1 X) \cdots)$ (Sequence of linear and non-linear transformations)



"Activation functions":
ReLu, Sigmoid, Tanh, Leaky ReLu, ELU, SELU...

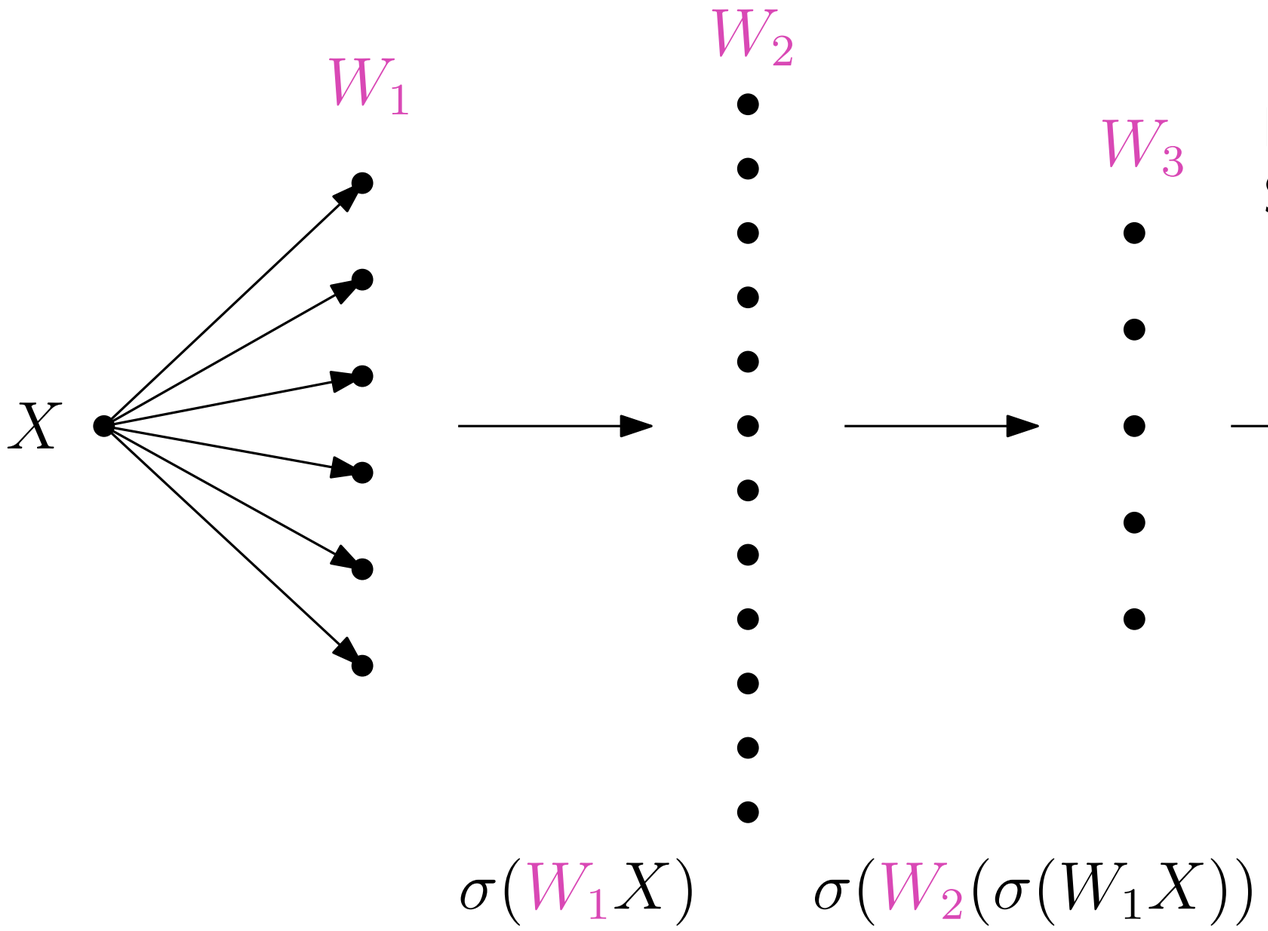
$$f(X) = \sigma(W_3 \sigma(W_2 \sigma(W_1 X)))$$

Find W_1, W_2, W_3 that Minimize:

$$l(f(X), Y) + r(W_1, W_2, W_3)$$

Deal with Overfitting for Neural-network

$f(X) = \sigma(W_l \sigma(W_{l-1} \cdots \sigma(W_1 X) \cdots)$ (Sequence of linear and non-linear transformations)



"Activation functions":
ReLu, Sigmoid, Tanh, Leaky ReLu, ELU, SELU...

$$f(X) = \sigma(W_3 \sigma(W_2 \sigma(W_1 X)))$$

Find W_1, W_2, W_3 that Minimize:

$$l(f(X), Y) + r(W_1, W_2, W_3)$$

NB: Usually, back-propagation procedure only approximate local minima

Deal with Overfitting for Neural-network

$$f(X) = \sigma(W_l \sigma(W_{l-1} \cdots \sigma(W_0 X) \cdots))$$

General idea: Reduce the dependence to the training data set D, reduce variability of output

Deal with Overfitting for Neural-network

$$f(X) = \sigma(W_l \sigma(W_{l-1} \cdots \sigma(W_0 X) \cdots))$$

General idea: Reduce the dependence to the training data set D, reduce variability of output

- Simplify the model

Deal with Overfitting for Neural-network

$$f(X) = \sigma(W_l \sigma(W_{l-1} \cdots \sigma(W_0 X) \cdots))$$

General idea: Reduce the dependence to the training data set D, reduce variability of output

- Simplify the model
- Increase regularization

Deal with Overfitting for Neural-network

$$f(X) = \sigma(W_l \sigma(W_{l-1} \cdots \sigma(W_0 X) \cdots))$$

General idea: Reduce the dependence to the training data set D, reduce variability of output

- Simplify the model
- Increase regularization
- Increase the number of training dataset when possible

Deal with Overfitting for Neural-network

$$f(X) = \sigma(W_l \sigma(W_{l-1} \cdots \sigma(W_0 X) \cdots))$$

General idea: Reduce the dependence to the training data set D, reduce variability of output

- Simplify the model
- Increase regularization
- Increase the number of training dataset when possible
- Early stop

Deal with Overfitting for Neural-network

$$f(X) = \sigma(W_l \sigma(W_{l-1} \cdots \sigma(W_0 X) \cdots))$$

General idea: Reduce the dependence to the training data set D, reduce variability of output

- Simplify the model
- Increase regularization
- Increase the number of training dataset when possible
- Early stop
- Add drop-out

Cross-validation to choose the best parameters

Typical problem: chose the correct regularization parameter λ in:

$$\text{Minimize } \frac{1}{n}\sum_{i=1}^{n} L(h_w(x_i), y_i) + \lambda\rho(w),\ w \in \mathbb{R}^p.$$

Cross validation is a general idea that work for multiple purpose in supervised learning.

Idea: Do multiple tests to find the best hyperparameters for the method

Cross-validation to choose the best parameters

Typical problem: chose the correct regularization parameter λ in:

$$\text{Minimize } \frac{1}{n}\sum_{i=1}^{n} L(h_w(x_i), y_i) + \lambda\rho(w),\ w \in \mathbb{R}^p.$$

Cross validation is a general idea that work for multiple purpose in supervised learning.

Idea: Do multiple tests to find the best hyperparameters for the method

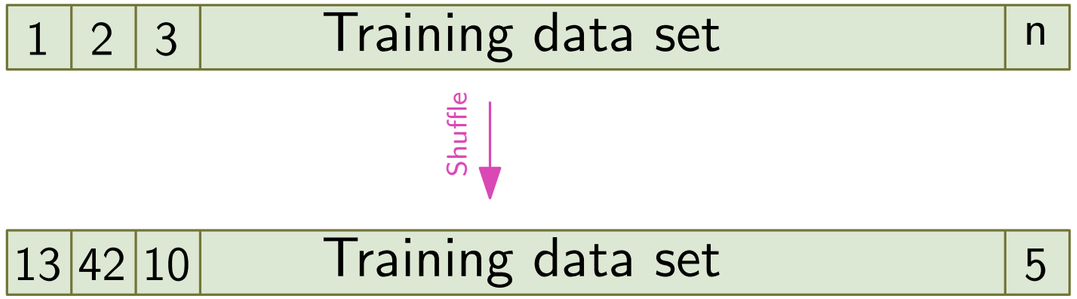
Cross-validation to choose the best parameters

Typical problem: chose the correct regularization parameter λ in:

$$\text{Minimize } \frac{1}{n}\sum_{i=1}^{n} L(h_w(x_i), y_i) + \lambda\rho(w),\ w \in \mathbb{R}^p.$$

Cross validation is a general idea that work for multiple purpose in supervised learning.

Idea: Do multiple tests to find the best hyperparameters for the method



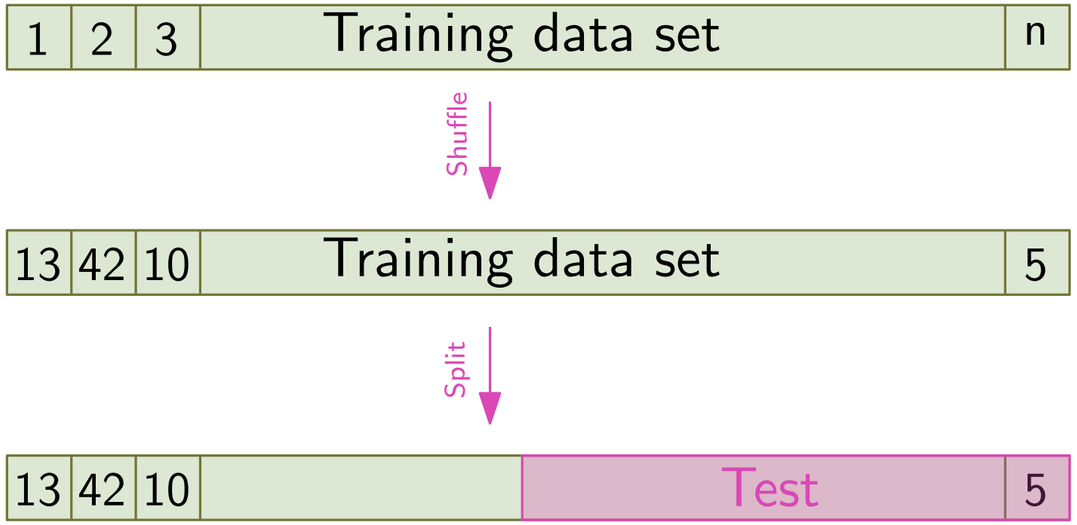
Cross-validation to choose the best parameters

Typical problem: chose the correct regularization parameter λ in:

$$\text{Minimize } \frac{1}{n}\sum_{i=1}^{n} L(h_w(x_i), y_i) + \lambda\rho(w),\ w \in \mathbb{R}^p.$$

Cross validation is a general idea that work for multiple purpose in supervised learning.

Idea: Do multiple tests to find the best hyperparameters for the method



Cross-validation to choose the best parameters

Typical problem: chose the correct regularization parameter λ in:

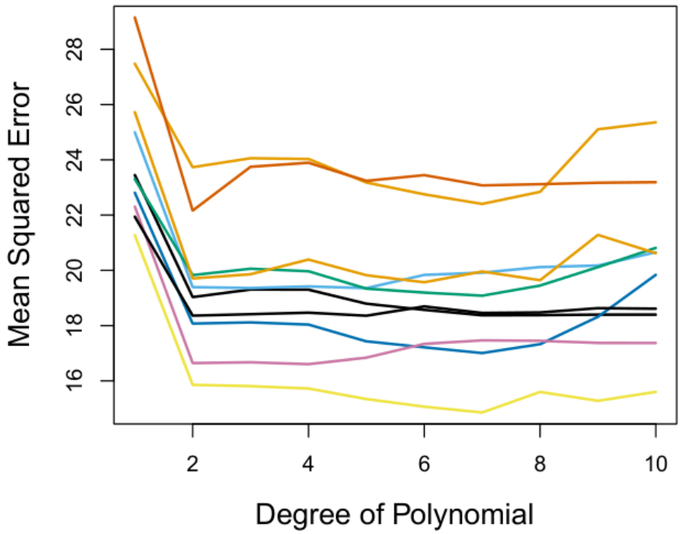
$$\text{Minimize } \frac{1}{n}\sum_{i=1}^{n} L(h_w(x_i), y_i) + \lambda\rho(w),\ w \in \mathbb{R}^p.$$

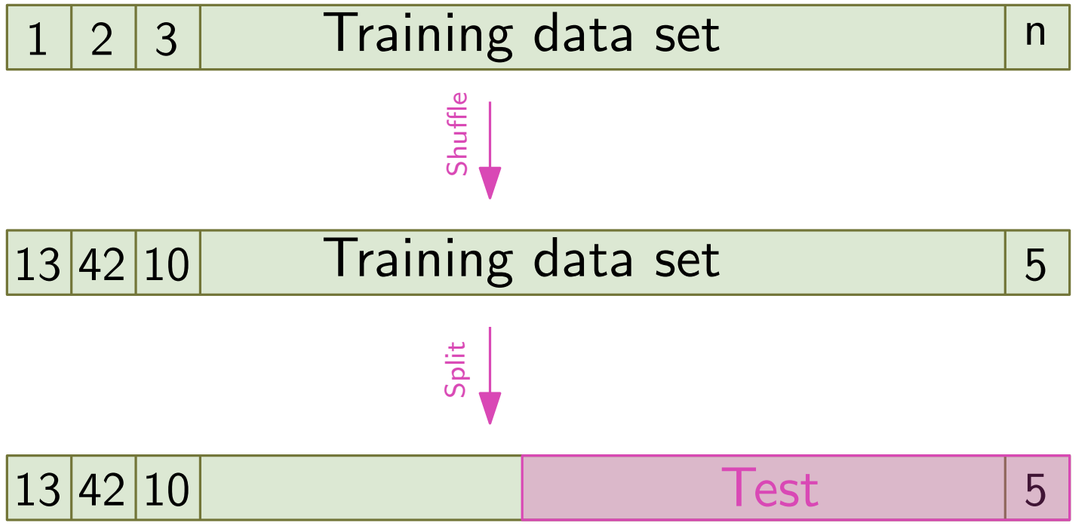
Cross validation is a general idea that work for multiple purpose in supervised learning.

Idea: Do multiple tests to find the best hyperparameters for the method

Example: for a regression task, try different polynomial degrees to fit the data.

Different colors = test errors of different split





Cross-validation to choose the best parameters

Typical problem: chose the correct regularization parameter λ in:

$$\text{Minimize } \frac{1}{n}\sum_{i=1}^{n} L(h_w(x_i), y_i) + \lambda\rho(w),\ w \in \mathbb{R}^p.$$

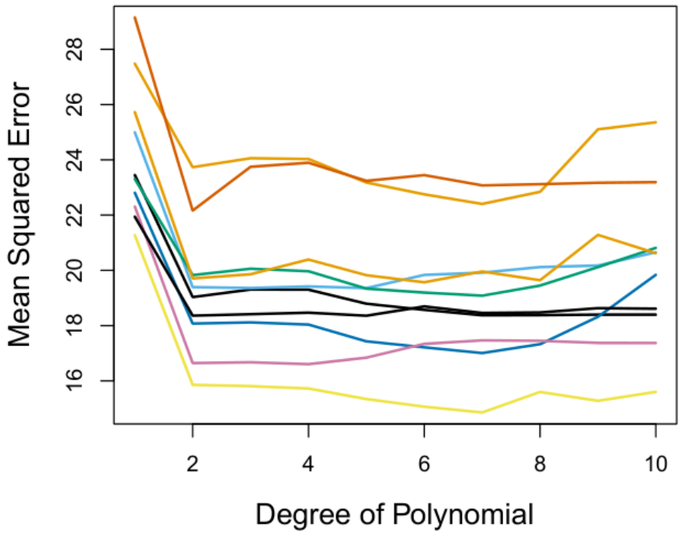
Cross validation is a general idea that work for multiple purpose in supervised learning.

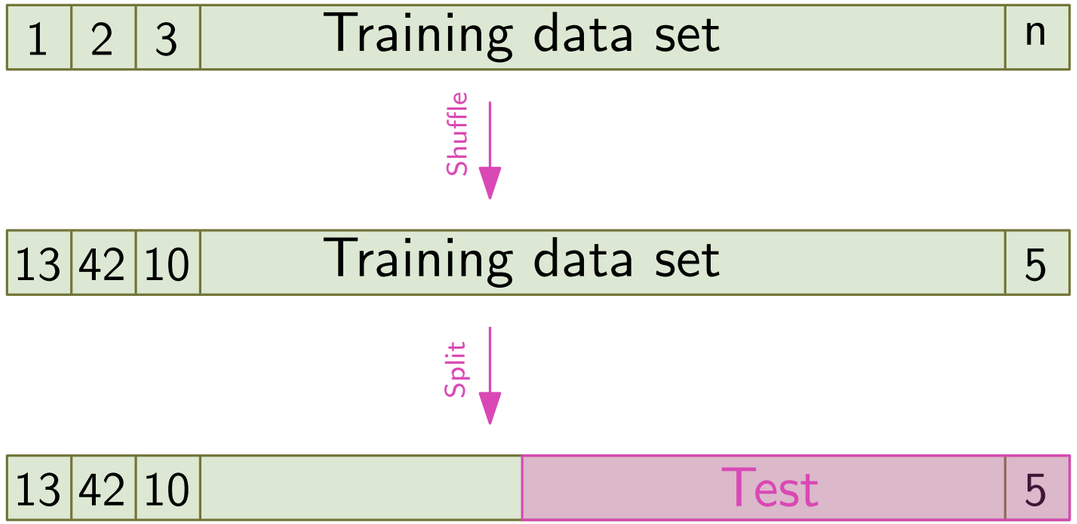
Idea: Do multiple tests to find the best hyperparameters for the method

Example: for a regression task, try different polynomial degrees to fit the data.

Different colors = test errors of different split

→ Variability because of split





Cross-validation to choose the best parameters

Typical problem: chose the correct regularization parameter λ in:

$$\text{Minimize } \frac{1}{n}\sum_{i=1}^{n} L(h_w(x_i), y_i) + \lambda\rho(w),\ w \in \mathbb{R}^p.$$

Improvement: k-fold cross validation

Cross-validation to choose the best parameters

Typical problem: chose the correct regularization parameter λ in:

$$\text{Minimize } \frac{1}{n}\sum_{i=1}^{n} L(h_w(x_i), y_i) + \lambda\rho(w),\ w \in \mathbb{R}^p.$$

Improvement: k-fold cross validation

Cross-validation to choose the best parameters

Typical problem: chose the correct regularization parameter λ in:

$$\text{Minimize } \frac{1}{n}\sum_{i=1}^{n} L(h_w(x_i), y_i) + \lambda\rho(w),\ w \in \mathbb{R}^p.$$

Improvement: k-fold cross validation

Cross-validation to choose the best parameters

Typical problem: chose the correct regularization parameter λ in:

$$\text{Minimize } \frac{1}{n}\sum_{i=1}^{n} L(h_w(x_i), y_i) + \lambda\rho(w),\ w \in \mathbb{R}^p.$$

Improvement: k-fold cross validation

- Divide the data set in k parts

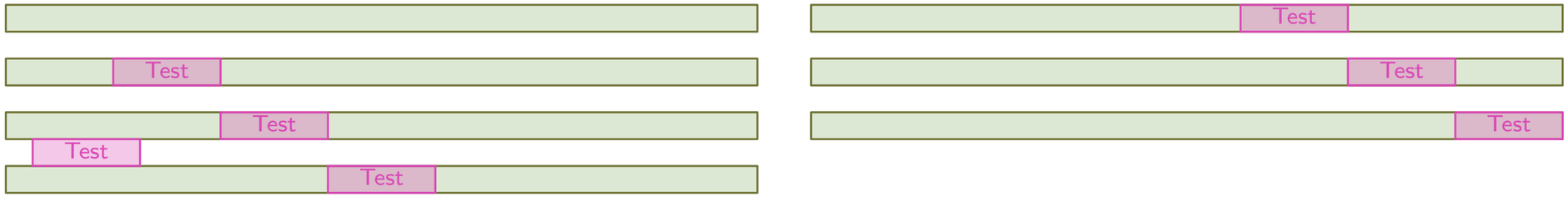
Cross-validation to choose the best parameters

Typical problem: chose the correct regularization parameter λ in:

$$\text{Minimize } \frac{1}{n}\sum_{i=1}^{n} L(h_w(x_i), y_i) + \lambda\rho(w),\ w \in \mathbb{R}^p.$$

Improvement: k-fold cross validation

- Divide the data set in k parts



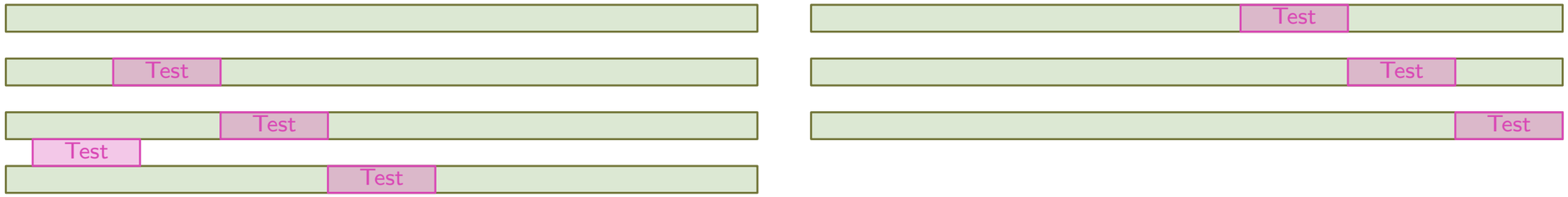
Cross-validation to choose the best parameters

Typical problem: chose the correct regularization parameter λ in:

$$\text{Minimize } \frac{1}{n}\sum_{i=1}^{n} L(h_w(x_i), y_i) + \lambda\rho(w),\ w \in \mathbb{R}^p.$$

Improvement: k-fold cross validation

- Divide the data set in k parts



- Try different λ for each setting

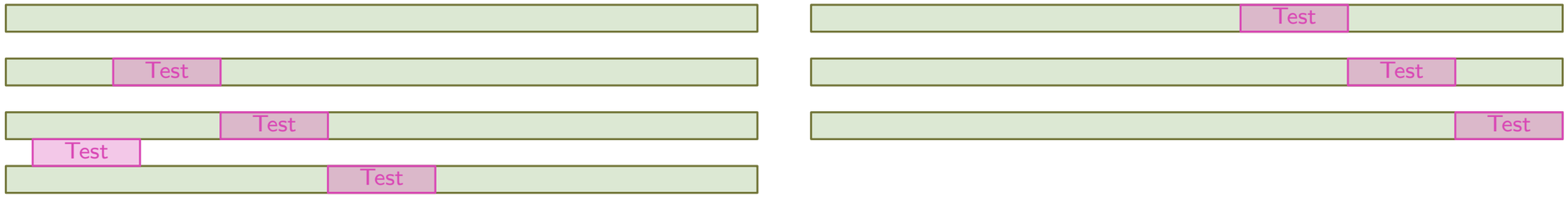
Cross-validation to choose the best parameters

Typical problem: chose the correct regularization parameter λ in:

$$\text{Minimize } \frac{1}{n}\sum_{i=1}^{n} L(h_w(x_i), y_i) + \lambda\rho(w),\ w \in \mathbb{R}^p.$$

Improvement: k-fold cross validation

- Divide the data set in k parts



- Try different λ for each setting
- Average the results and choose the one that maximizes the performances

Cross-validation to choose the best parameters

Typical problem: chose the correct regularization parameter λ in:

$$\text{Minimize } \frac{1}{n}\sum_{i=1}^{n} L(h_w(x_i), y_i) + \lambda\rho(w),\ w \in \mathbb{R}^p.$$

Improvement: k-fold cross validation