Lecture 08

Model Evaluation Part 1: Introduction to Overfitting and Underfitting

STAT 451: Machine Learning, Fall 2020

Sebastian Raschka

http://stat.wisc.edu/~sraschka/teaching/stat451-fs2020/

Where we are in this course

Part 1: Introduction

- L01 - Course overview, introduction to machine learning
- L02 - Introduction to Supervised Learning and k-Nearest Neighbors Classifiers

Part 2: Computational foundations

- L03 - Using Python
- L04 - Introduction to Python's scientific computing stack
- L05 - Data preprocessing and machine learning with scikit-learn

Part 3: Tree-based methods

- L06 - Decision trees
- L07 - Ensemble methods

Part 4: Model evaluation

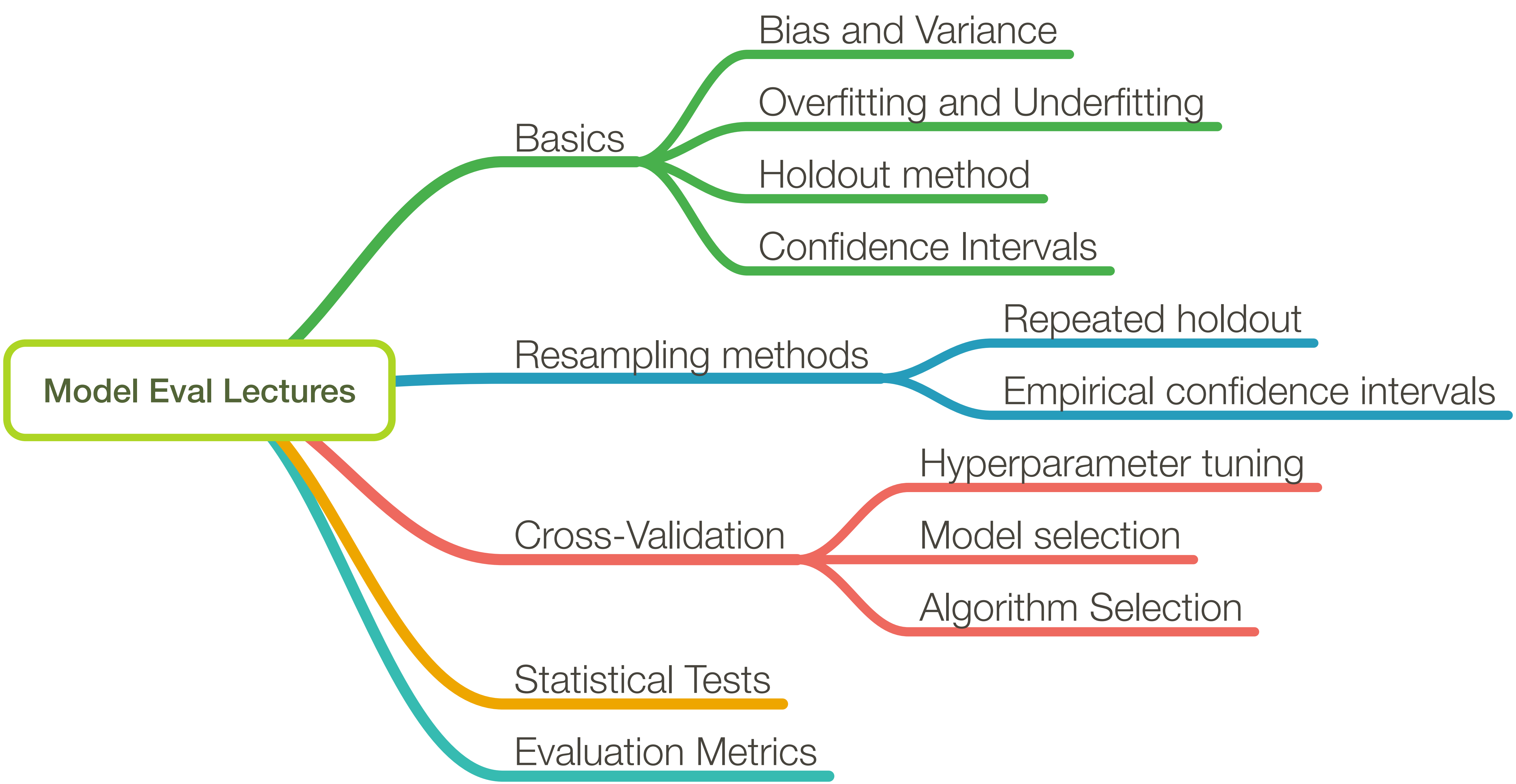
- Midterm exam
- L08 - Model evaluation 1 – overfitting
- L09 - Model evaluation 2 – confidence intervals
- L10 - Model evaluation 3 – cross-validation and model selection
- L11 - Model evaluation 4 – algorithm selection
- L12 - Model evaluation 5 – evaluation and performance metrics

Part 5: Dimensionality reduction and unsupervised learning

- L13 - Feature selection
- L14 - Feature extraction
- L15 - Clustering

Part 6: Bayesian learning

Overview



Model Eval Lectures
Basics
Bias and Variance
Overfitting and Underfitting
Holdout method
Confidence Intervals
Resampling methods
Repeated holdout
Empirical confidence intervals
Cross-Validation
Hyperparameter tuning
Model selection
Algorithm Selection
Statistical Tests
Evaluation Metrics

8.1 Overfitting and Underfitting

8.2 Intro to Bias-Variance Decomposition

8.3 Bias-Variance Decomposition of the Squared Error

8.4 Relationship between Bias-Variance Decomposition and Overfitting and Underfitting

8.5 Bias-Variance Decomposition of the 0/1 Loss

8.6 Other Forms of Bias

Overfitting and Underfitting

Overfitting and Underfitting

Generalization Performance

Want a model to "generalize" well to _________ data

(Want "high generalization accuracy" or "low generalization error")

Overfitting and Underfitting

Assumptions

- i.i.d. assumption: training and test examples are independent and identically distributed (drawn from the same joint probability distribution, P(**X**, y))
- For some random model that has not been fitted to the training set,
  we expect **the training error is** ______ _____ **the test error**
- The training error or accuracy provides
  an _______**imistically biased** estimate of the generalization performance

Overfitting and Underfitting

Model Capacity

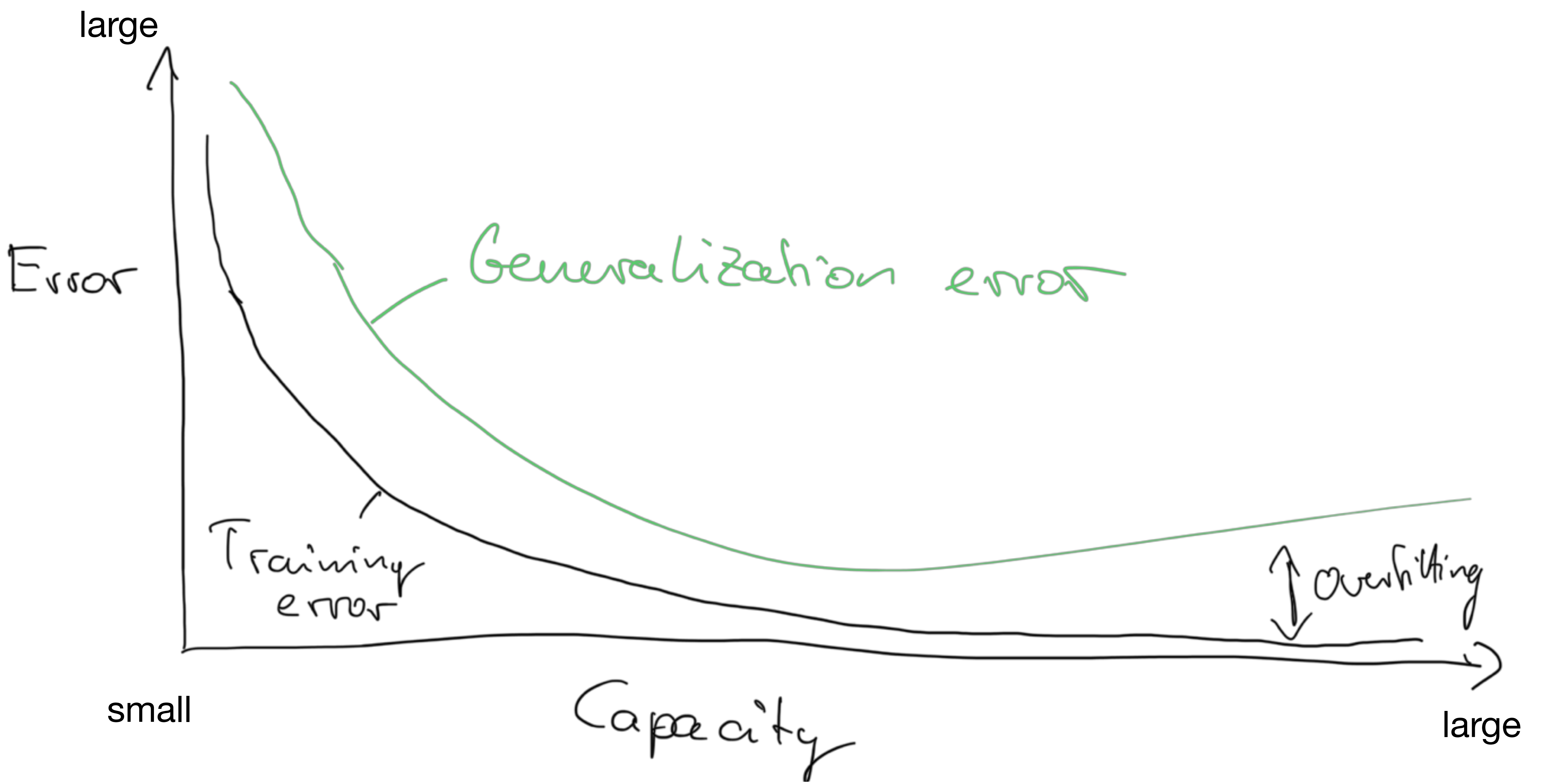
- Underfitting: both the training and test error are _________

- Overfitting: gap between training and test error
(where test error is larger)

- Large hypothesis space being searched by a learning algorithm

-> high tendency to ______fit

Overfitting and Underfitting



"[...] model has high bias/variance" -- What does that mean?

Google Scholar "model has high bias"

Articles About 70 results (**0.20** sec) My profile My library

Any time
Since 2020
Since 2019
Since 2016
Custom range...

Sort by relevance
Sort by date

include patents
include citations

Create alert

Evaluation of regression models: Model assessment, model selection and generalization error — [PDF] mdpi.com
F Emmert-Streib, M Dehmer - Machine learning and knowledge extraction, 2019 - mdpi.com
When performing a regression or classification analysis, one needs to specify a statistical model. This model should avoid the overfitting and underfitting of data, and achieve a low generalization error that characterizes its prediction performance. In order to identify such a model …
Cited by 14 Related articles All 3 versions

[PDF] A Comparative Simulation Study of ARIMA and Fuzzy Time Series Model for Forecasting Time Series Data — [PDF] academia.edu
HA Haji, K Sadik, AM Soleh - 2018 - academia.edu
… θ=0.9 for both σe 2 = 3 and σe 2 = 5 , which is to be expected. But for Yu **model has high bias** for that condition.The relationship between the bias and other forecasting accuracy measures is roughly linear for all methods. Furthermore, The largest bias for σe 2 = 5 is …
Related articles All 4 versions

[PDF] Prediction of Yelp Review Star Rating using Sentiment Analysis — [PDF] stanford.edu
C Li, J Zhang - 2014 - cs229.stanford.edu
… Final Report Figure 4: Ablative Analysis for 5-star Classification. As we can see, removing features may lead to higher mean square error, which supported our hypothesis that the resulted **model has high bias** and needs more features. 5.2 Recommendation Model …
Cited by 5 Related articles

[PDF] Automatic recognition of handwritten digits using multi-layer sigmoid neural network — [PDF] semanticscholar.org
SK Katungunya, X Ding… - International Journal of …, 2016 - pdfs.semanticscholar.org
… regularization parameter (λ). Regularization add a penalty term that depends on the characteristics of the parameters. If a **model has high bias**, decreasing the effect of regularization can lead to better results. A high variance …
Cited by 1 Related articles

[PDF] Overfitting vs. underfitting: A complete example — [PDF] pstu.ac.bd
W Koehrsen - Towards Data Science, 2018 - pstu.ac.bd
… model depends very little on the training data because it barely pays any attention to the points! Instead, the **model has high bias**, which means it makes a strong assumption Under t 1 degree

Google Scholar "model has high variance"

Articles About 113 results (0.20 sec) My profile My library

Any time
Since 2020
Since 2019
Since 2016
Custom range...

Sort by relevance
Sort by date

include patents
include citations

Create alert

Evaluation of regression models: Model assessment, model selection and generalization error [PDF] mdpi.com
F Emmert-Streib, M Dehmer - Machine learning and knowledge extraction, 2019 - mdpi.com
When performing a regression or classification analysis, one needs to specify a statistical model. This model should avoid the overfitting and underfitting of data, and achieve a low generalization error that characterizes its prediction performance. In order to identify such a model …
Cited by 14 Related articles All 3 versions

[HTML] Bias-variance decomposition of errors in data-driven land cover change modeling [HTML] springer.com
J Gao, AC Burnicki, JE Burt - Landscape Ecology, 2016 - Springer
… AdaBoosting is expected to noticeably reduce modeling error only if the base **model has high variance**; if the base model performs poorly, boosting may transform it into a worse model (Breiman 1996; Domingos 2000). Results. Interpreting error component maps …
Cited by 2 Related articles All 6 versions

A Novel Accurate and Fast Converging Deep Learning-Based Model for Electrical Energy Consumption Forecasting in a Smart Grid [PDF] mdpi.com
G Hafeez, KS Alimgeer, Z Wadud, Z Shafiq… - Energies, 2020 - mdpi.com
Energy consumption forecasting is of prime importance for the restructured environment of energy management in the electricity market. Accurate energy consumption forecasting is essential for efficient energy management in the smart grid (SG); however, the energy consumption …
Cited by 2 Related articles All 6 versions

[PDF] Model-based motion planning [PDF] umass.edu
B Burns, O Brock - Computer Science Department Faculty …, 2004 - scholarworks.umass.edu
… random. Cohn et al. [10] note that hill-climbing may also be used to find ˜x, but we have not found this to be necessary. The result is a sampling strategy that only queries sample points at which the **model has high variance**. A …
Cited by 11 Related articles All 11 versions

Signalling and the pricing of new issues [PDF] wiley.com
M Grinblatt, CY Hwang - The Journal of Finance, 1989 - Wiley Online Library
… Nanda (1988), 2 In Nanda's model, firms with high mean returns also have low variances. Since this **model has high-variance** low-mean firms issuing debt, high-mean firms are penalized by issuing debt that is perceived as being riskier than it really is

8.1 Overfitting and Underfitting

8.2 Intro to Bias-Variance Decomposition

8.3 Bias-Variance Decomposition of the Squared Error

8.4 Relationship between Bias-Variance Decomposition and Overfitting and Underfitting

8.5 Bias-Variance Decomposition of the 0/1 Loss

8.6 Other Forms of Bias

"[...] model has high bias/variance" -- What does that mean?

Bias-Variance Decomposition and Bias-Variance Trade-off

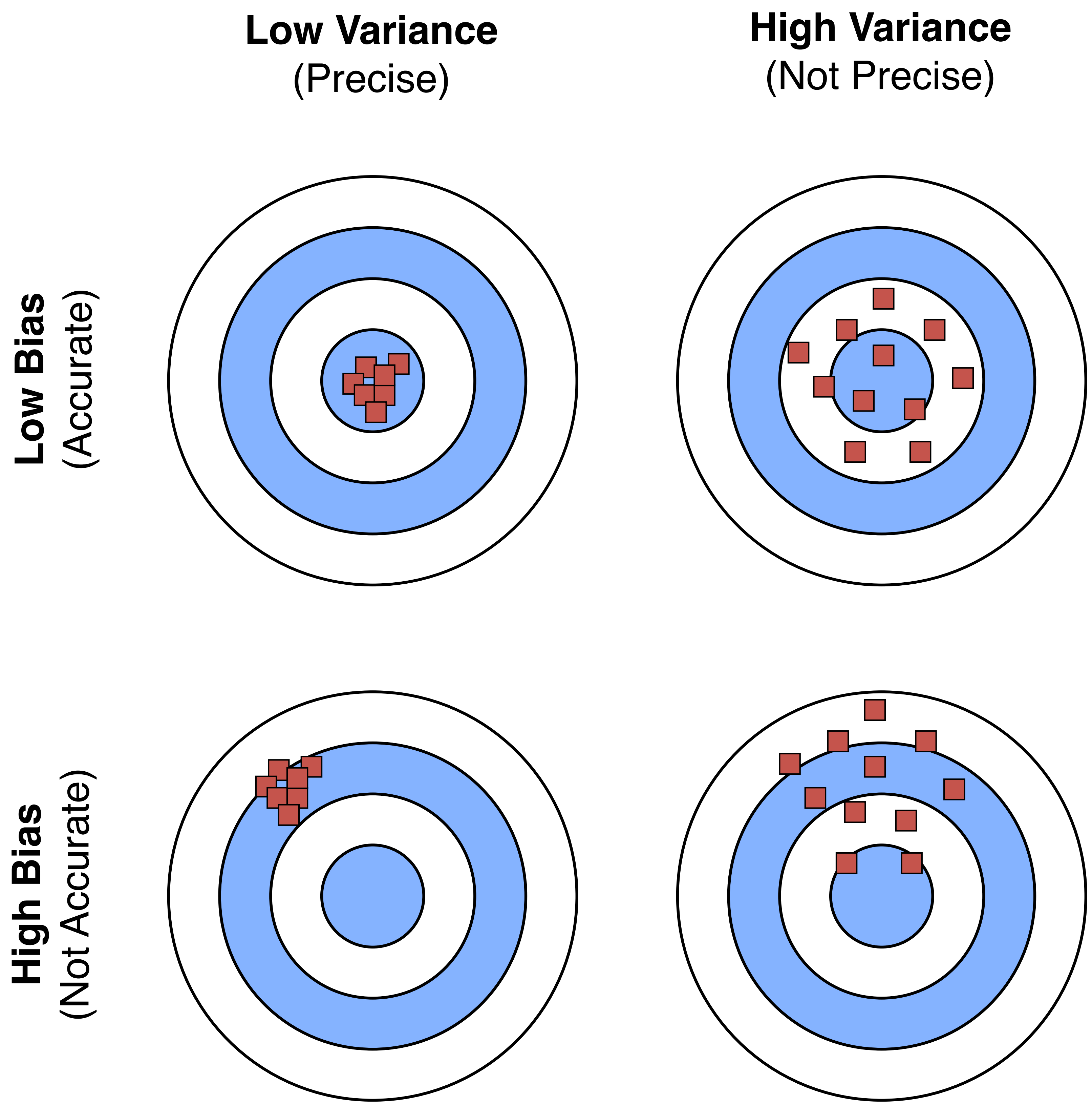
(and how it related to overfitting and underfitting)

Bias-Variance Decomposition

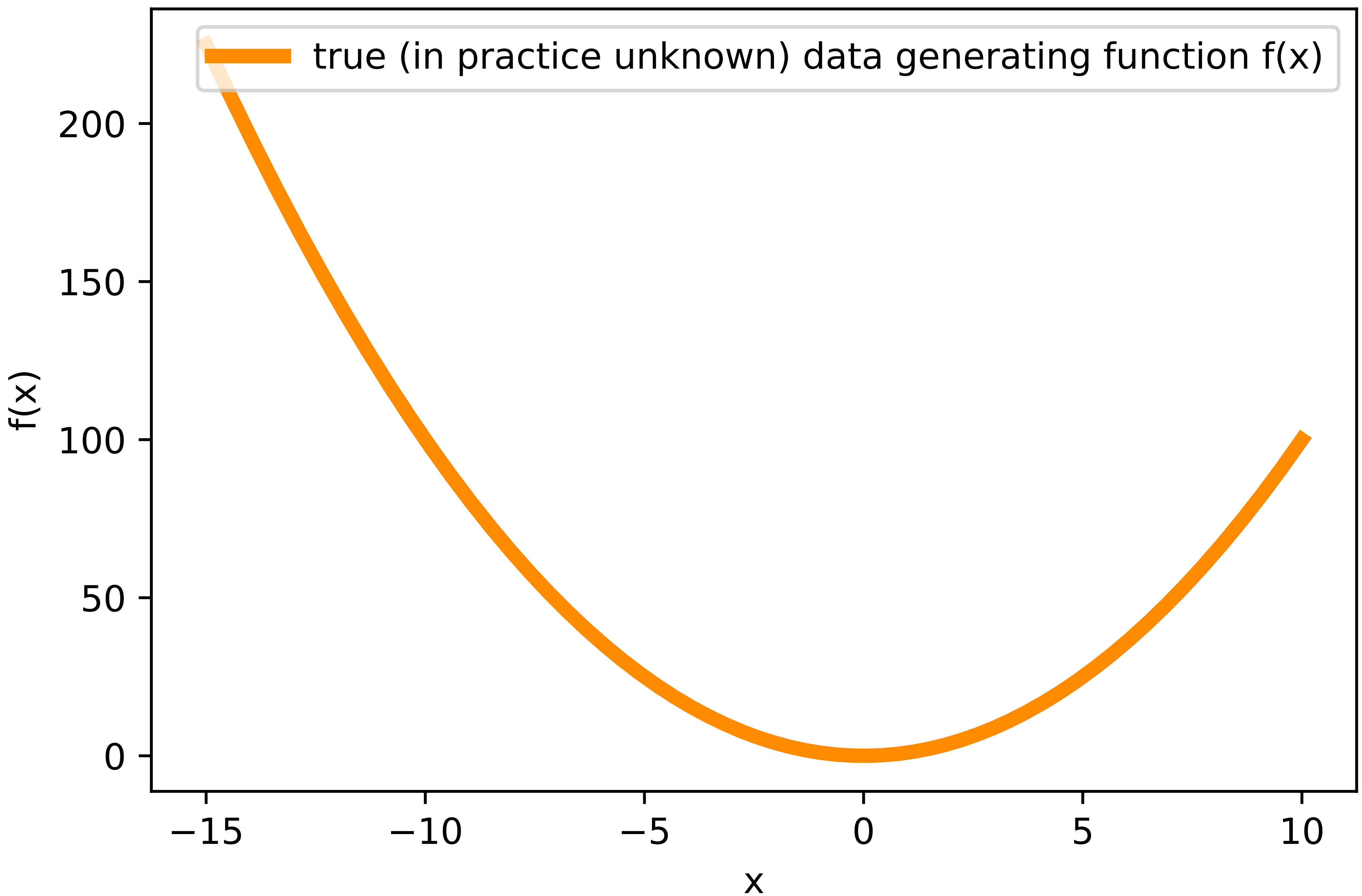
- Decomposition of the loss into bias and variance help us understand learning algorithms, concepts are related to underfitting and overfitting

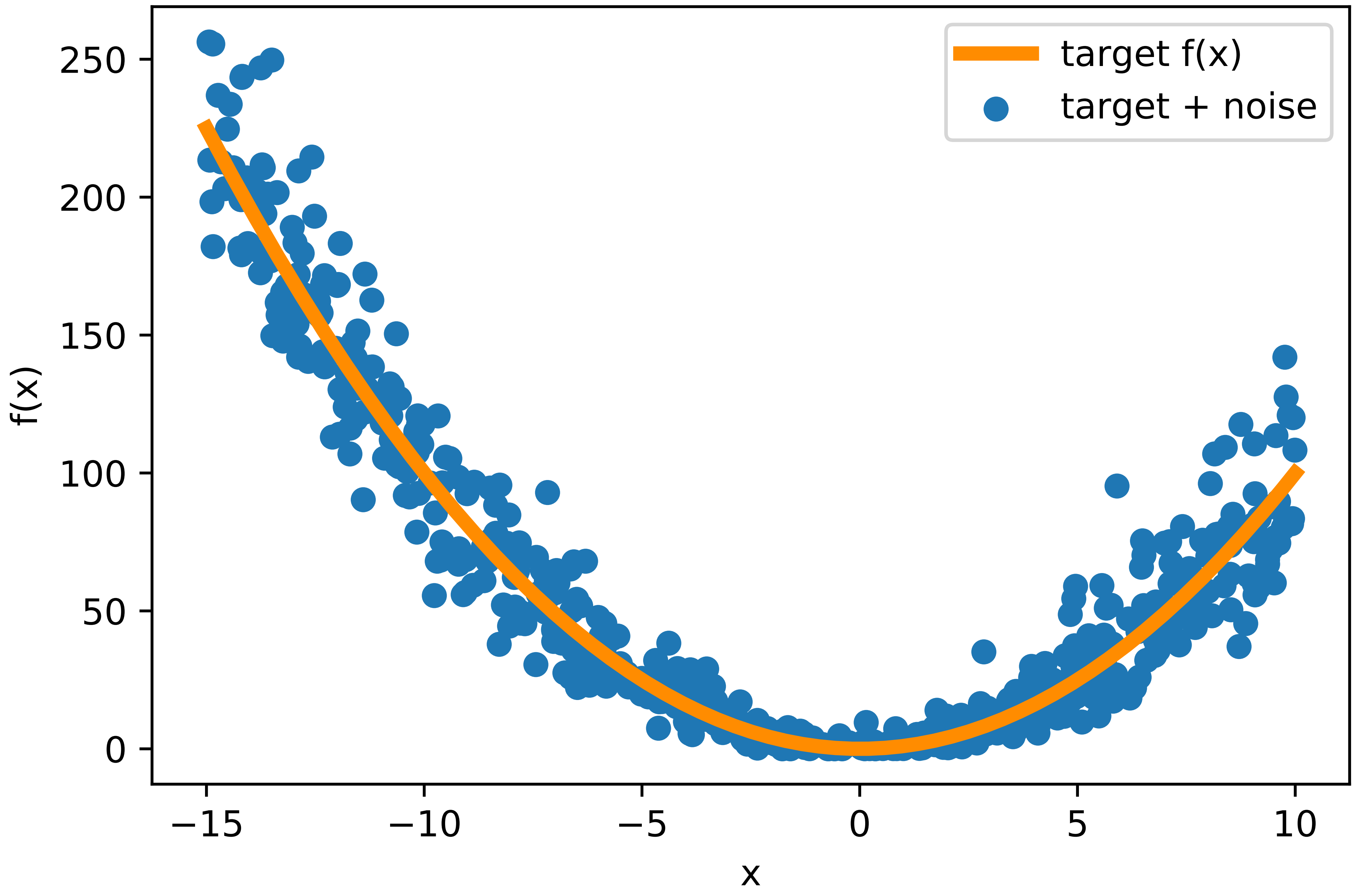
- Helps explain why ensemble methods (last lecture) might perform better than single models

Bias-Variance Intuition

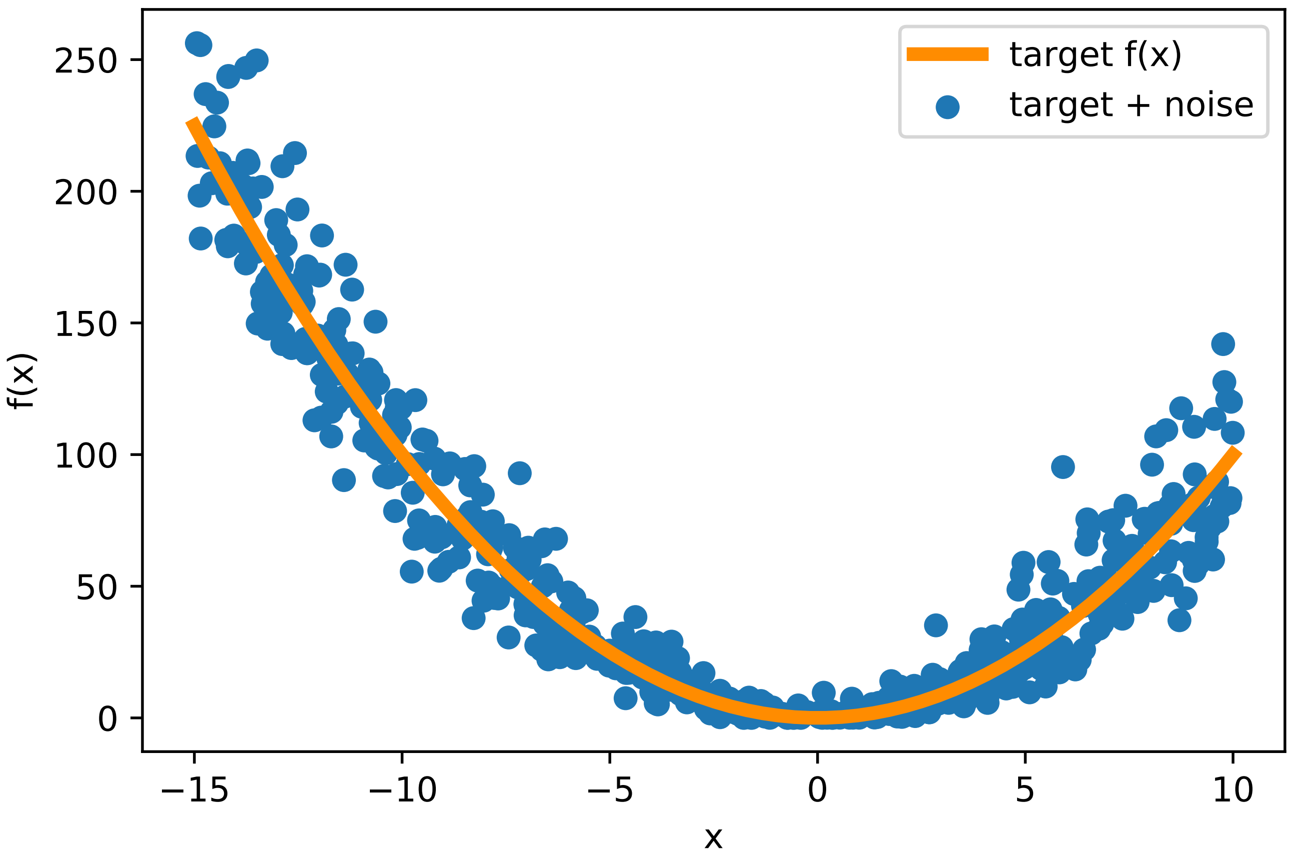


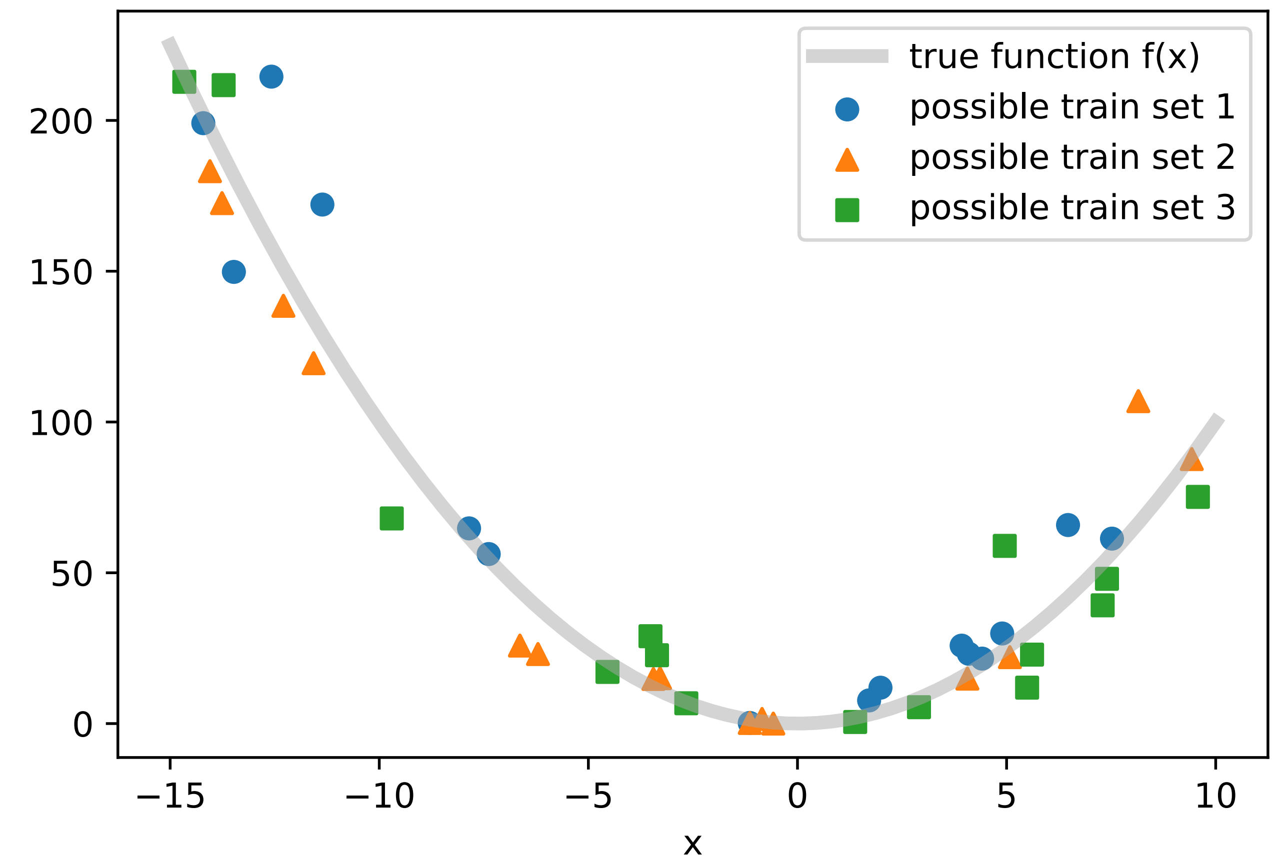
Bias-Variance Intuition



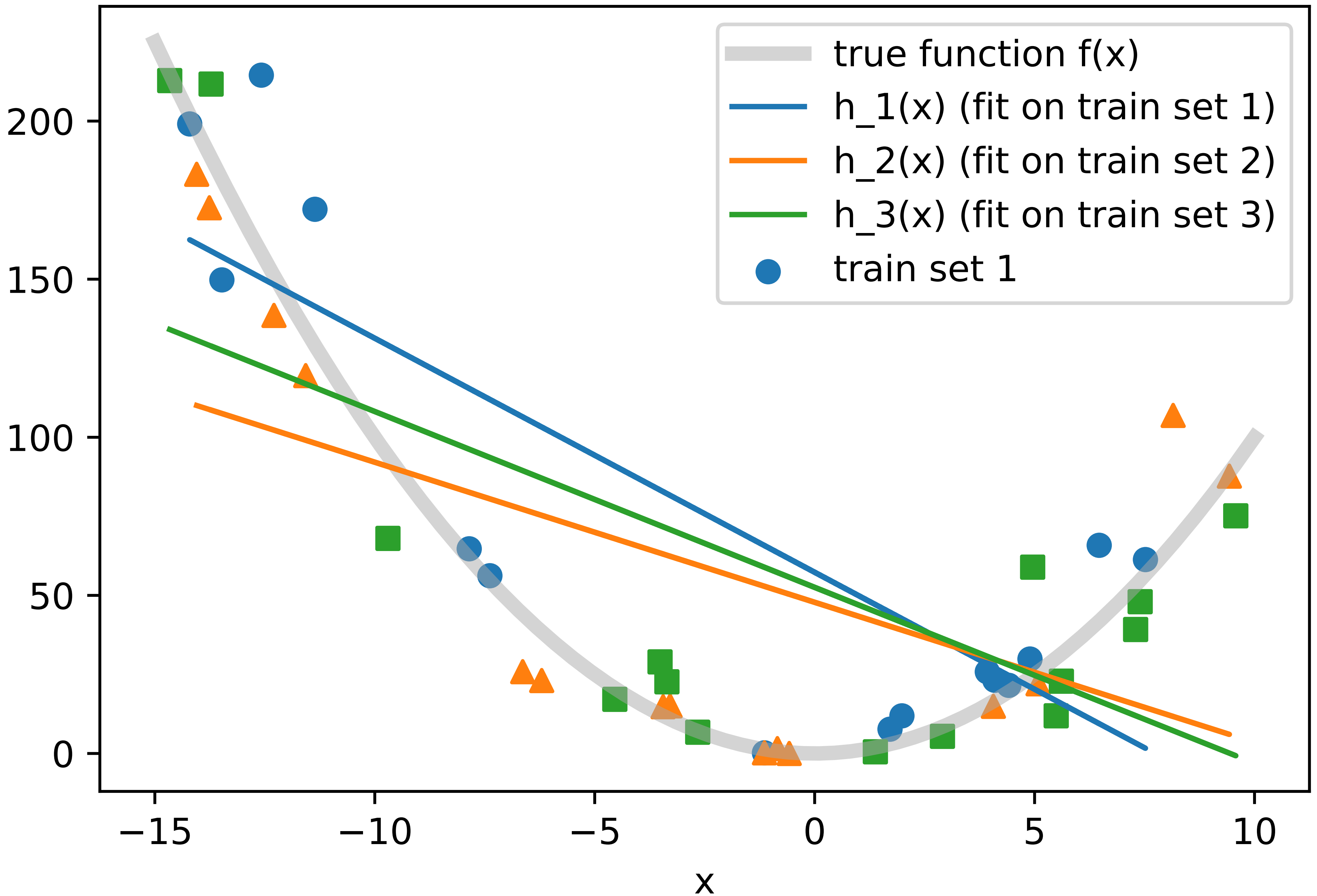

target f(x)
target + noise
250
200
150
100
50
0
f(x)
−15
−10
−5
0
5
10
x

Bias-Variance Intuition



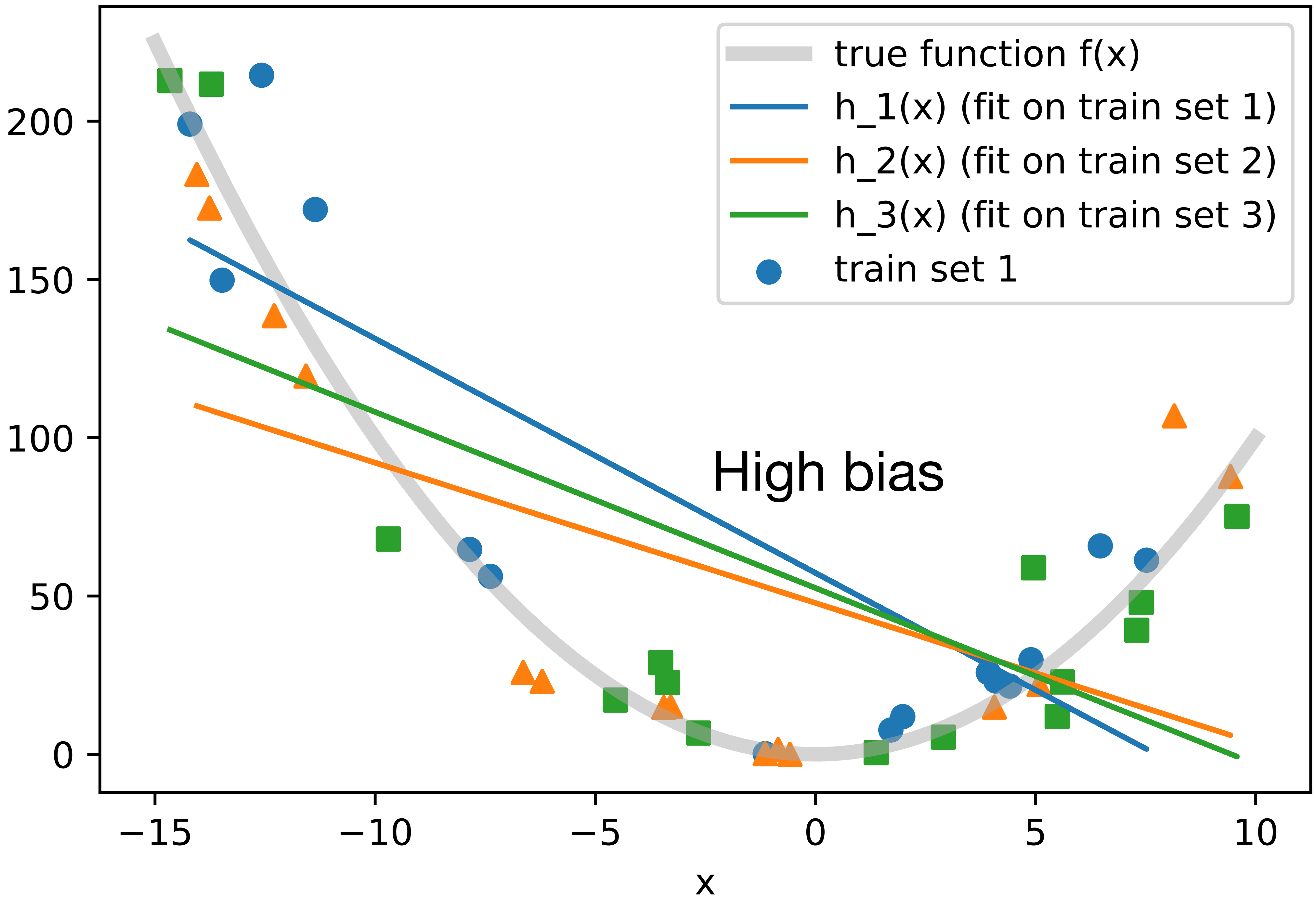


Bias-Variance Intuition



linear regression models

Bias-Variance Intuition

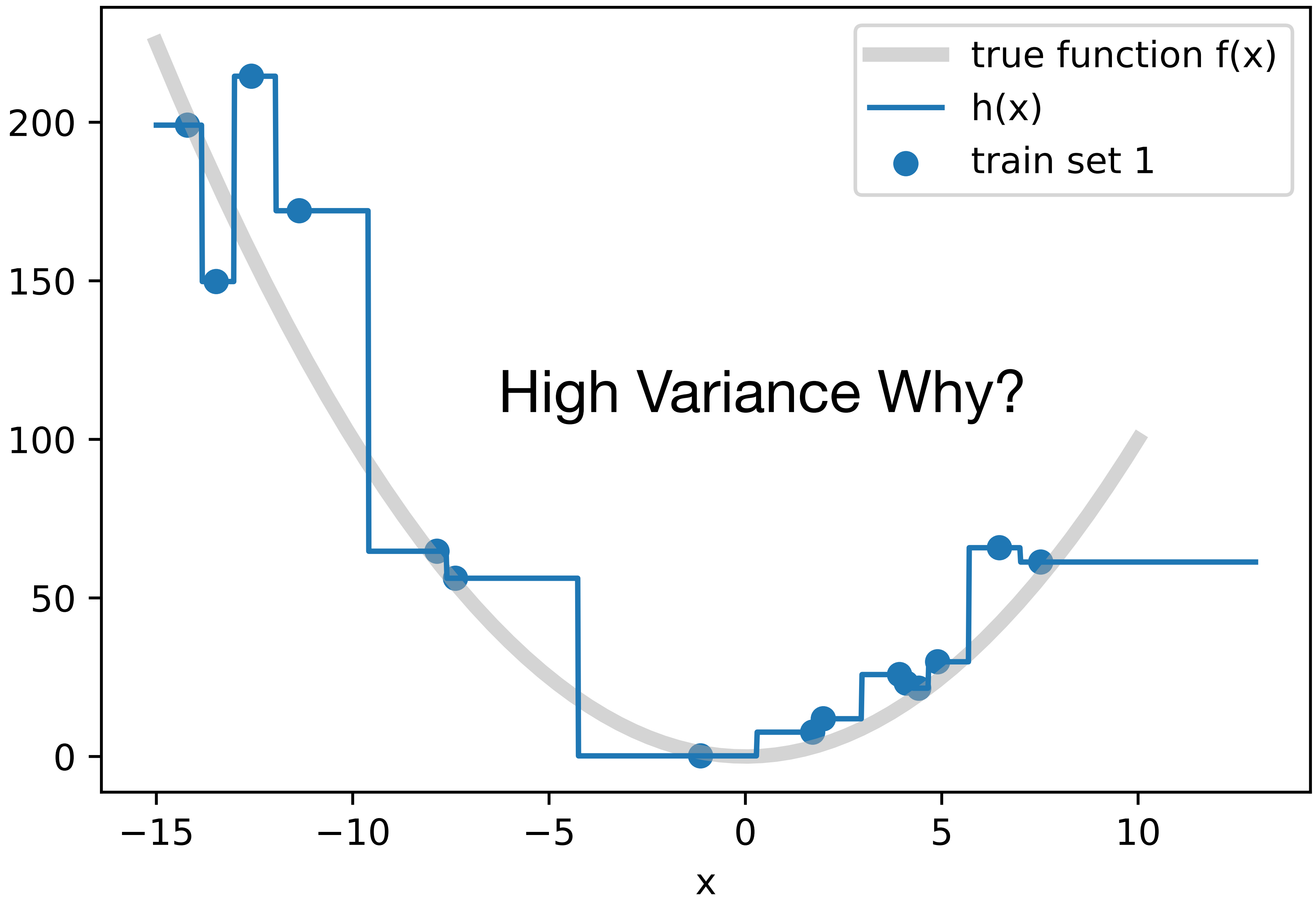


linear regression models

(There are points where the bias is zero ...)

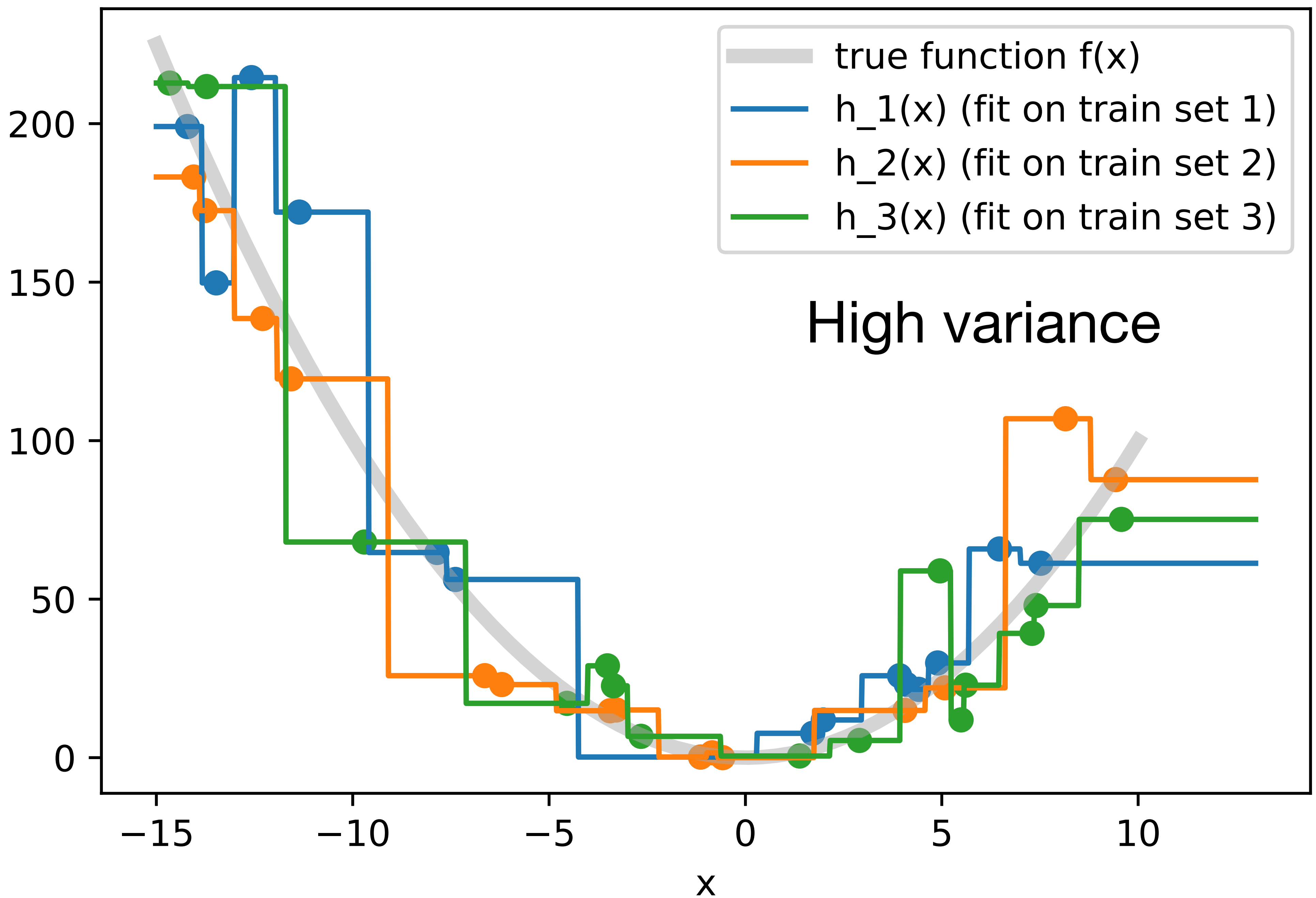
Bias-Variance Intuition

(here, I fit an unpruned decision tree)



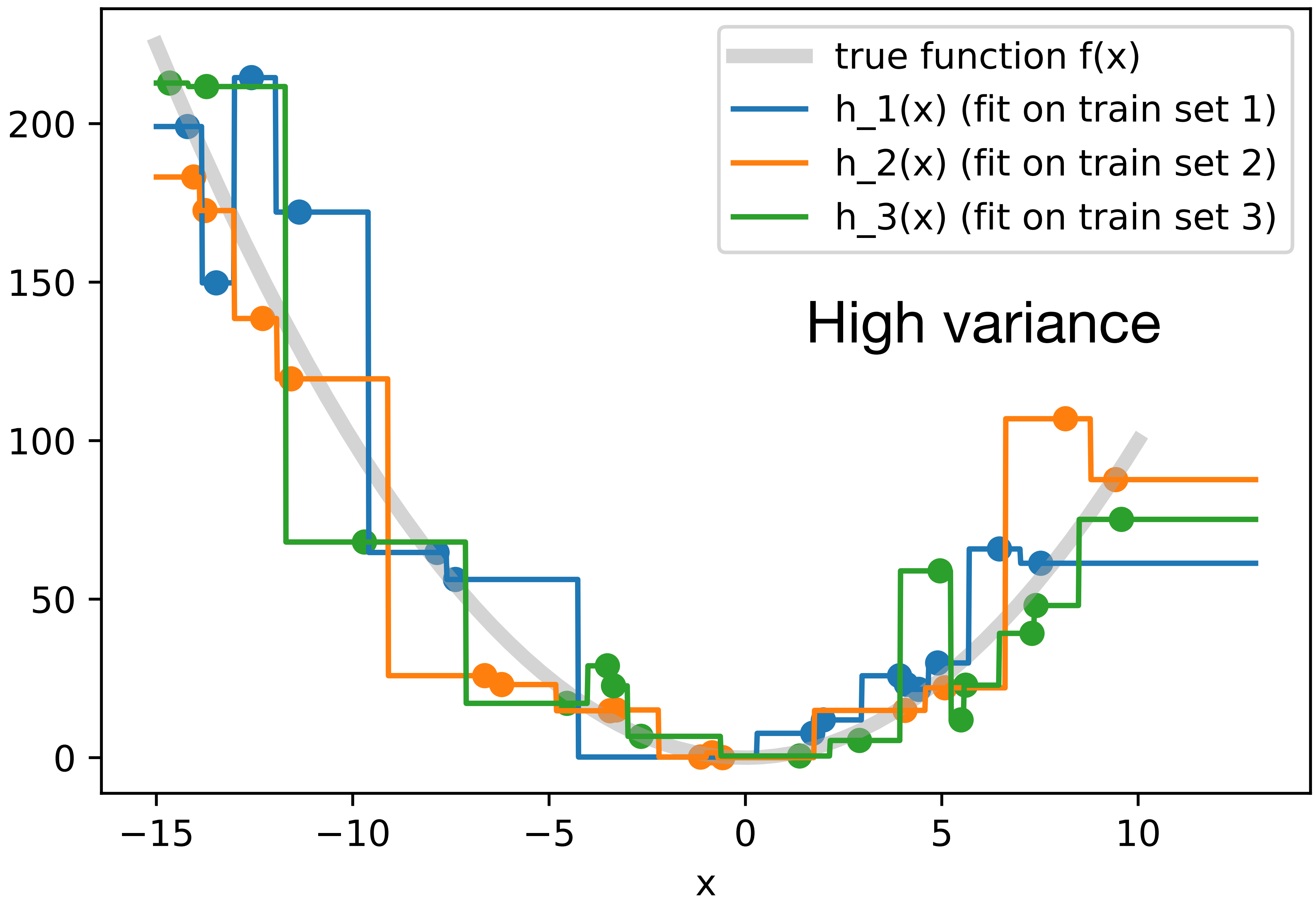
Bias-Variance Intuition

suppose we have multiple training sets

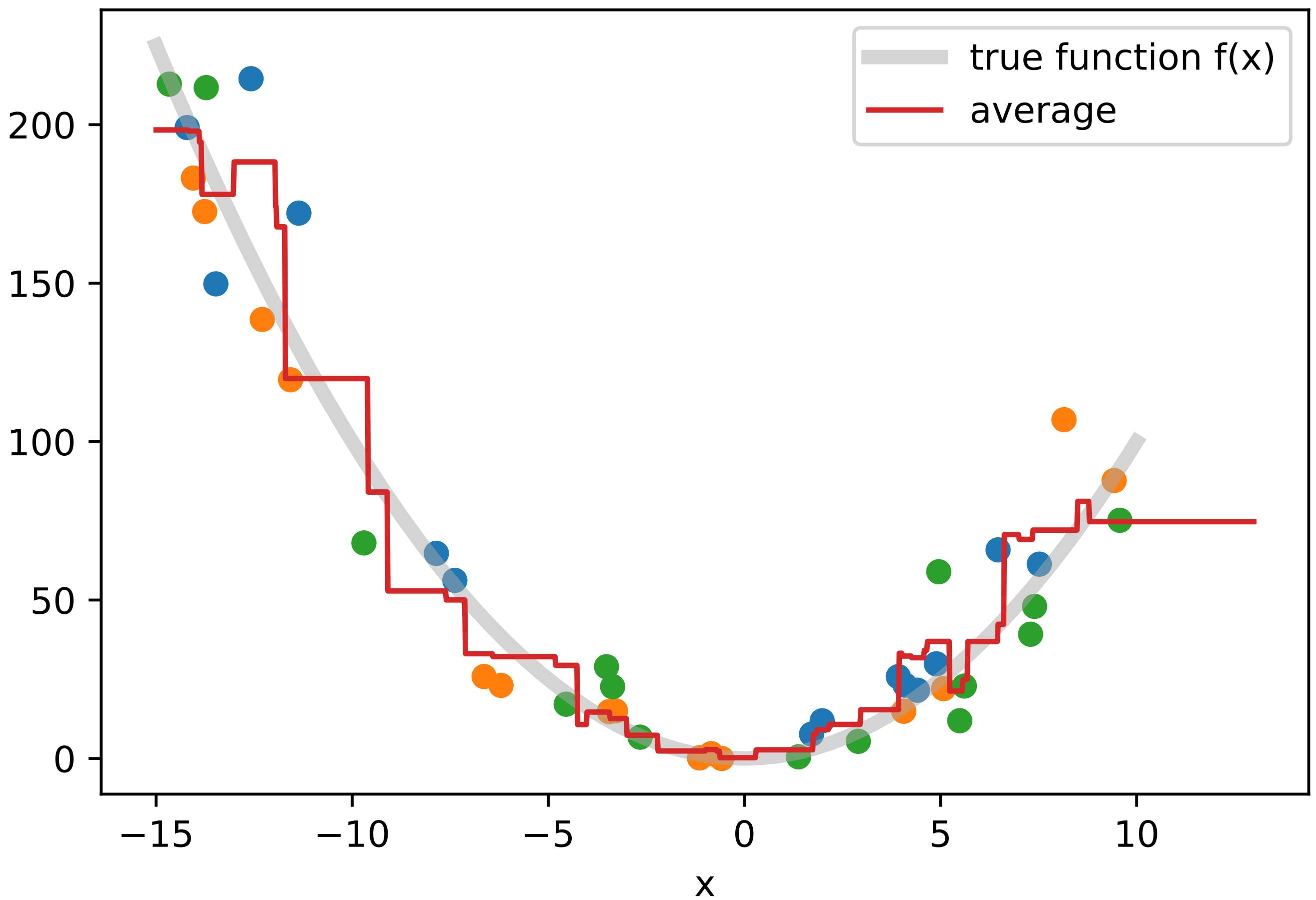


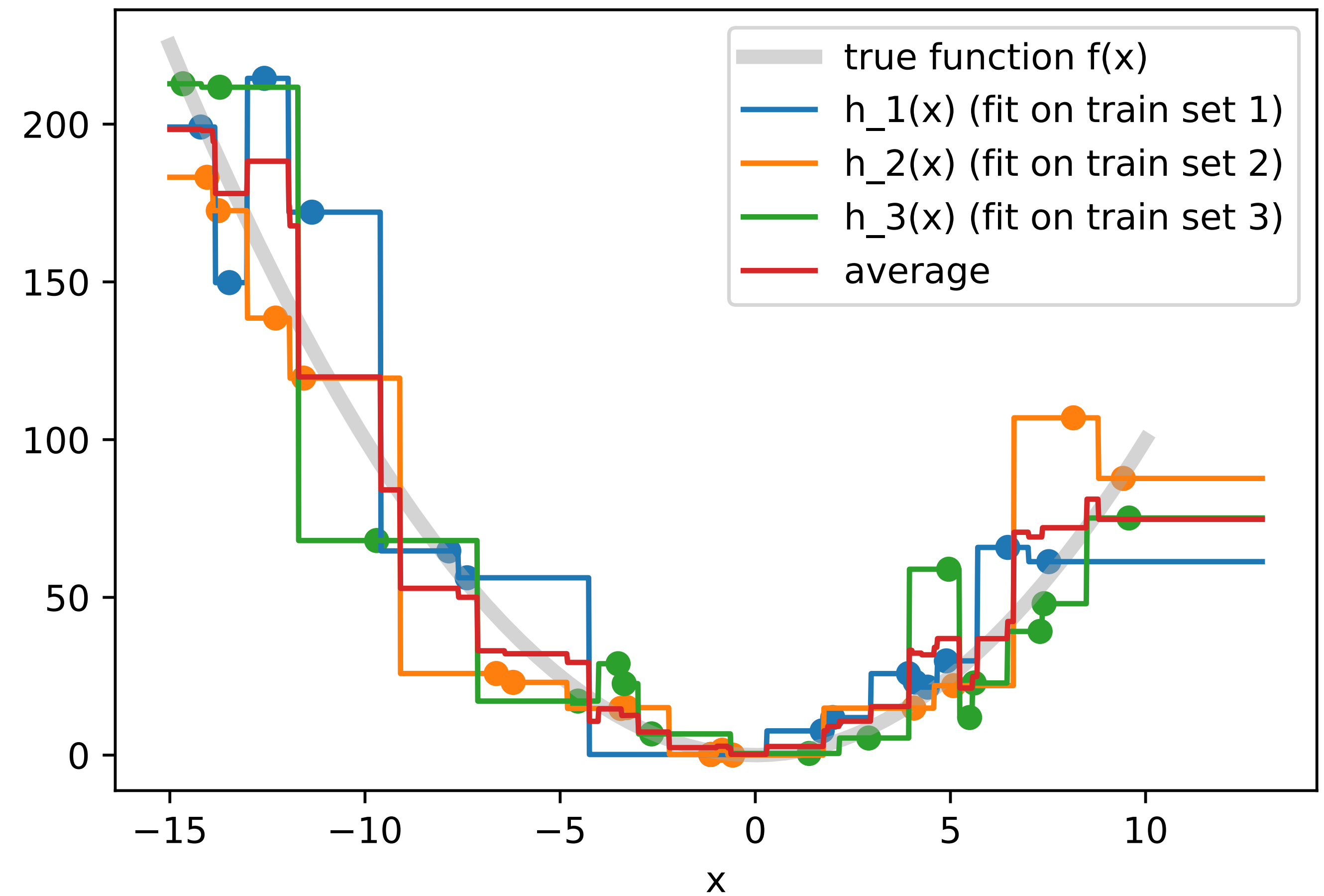
Bias-Variance Intuition

What happens if we take the average?
Does this remind you of something?



Bias-Variance Intuition





Terminology

Point estimator $\hat{\theta}$ of some parameter θ

(could also be a function, e.g., the hypothesis is an estimator of some target function)

Terminology

Point estimator $\hat{\theta}$ of some parameter θ

(could also be a function, e.g., the hypothesis is an estimator of some target function)

$$\text{Bias} = E[\hat{\theta}] - \theta$$

Terminology

General Definition

$$\text{Bias}[\hat{\theta}] = E[\hat{\theta}] - \theta$$

$$\text{Var}[\hat{\theta}] = E[\hat{\theta}^2] - (E[\hat{\theta}])^2$$

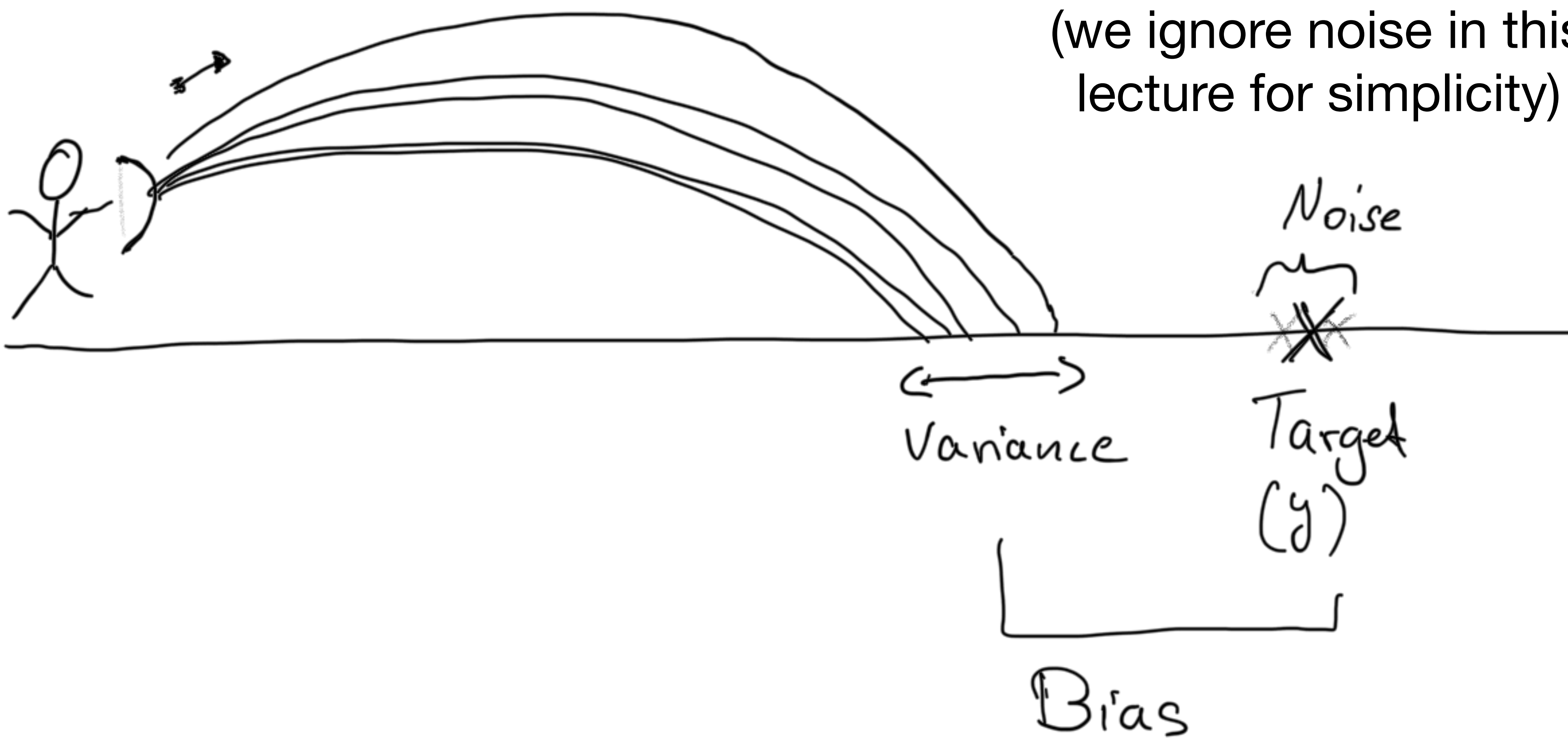
$$\text{Var}[\hat{\theta}] = E\left[(E[\hat{\theta}] - \hat{\theta})^2\right]$$

Terminology

$$\text{Bias}[\hat{\theta}] = E[\hat{\theta}] - \theta \qquad \text{Var}[\hat{\theta}] = E\left[(E[\hat{\theta}] - \hat{\theta})^2\right]$$

Intuition

(we ignore noise in this lecture for simplicity)



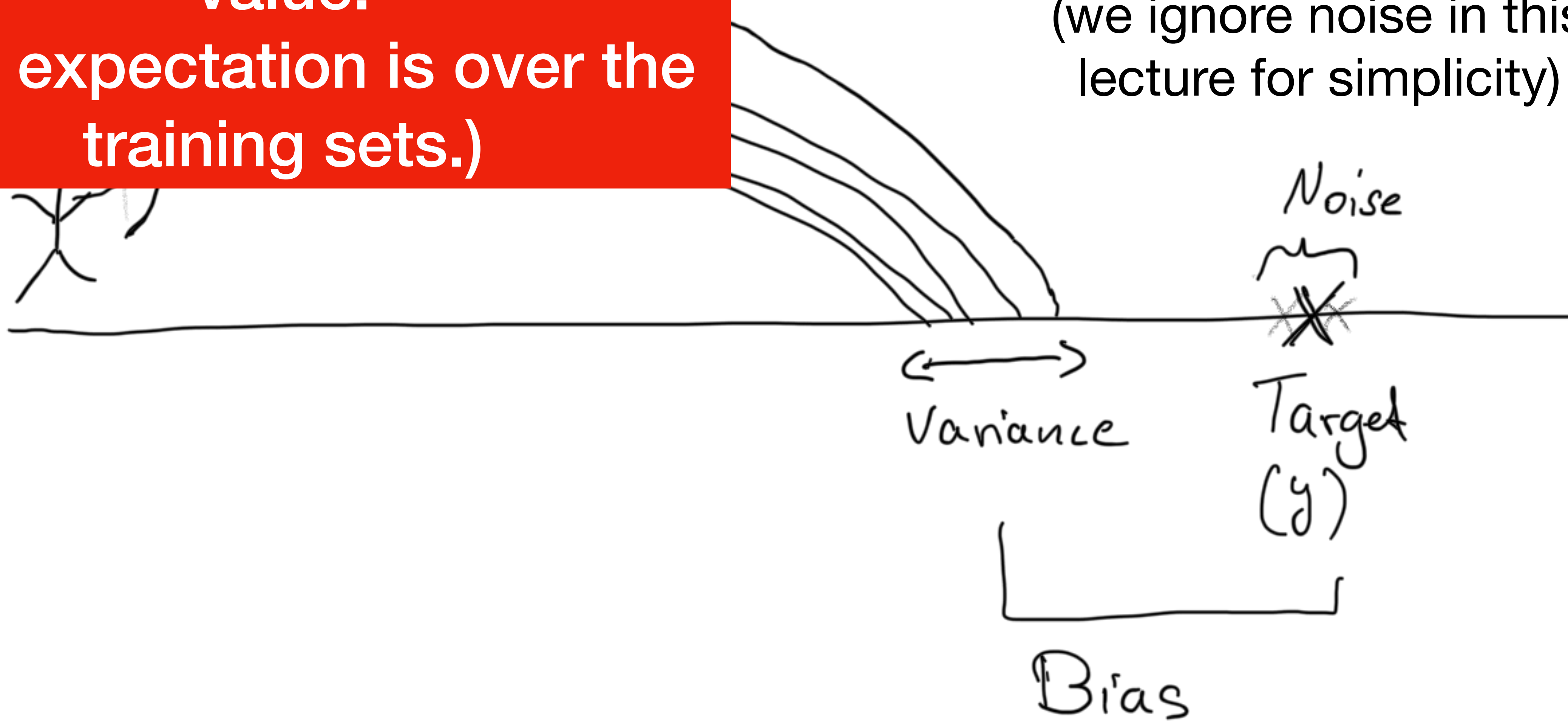
Terminology

$$\text{Bias}[\hat{\theta}] = E[\hat{\theta}] - \theta$$

$$\text{Var}[\hat{\theta}] = E\left[(E[\hat{\theta}] - \hat{\theta})^2\right]$$

Bias is the difference between the average estimator from different training samples and the true value.
(The expectation is over the training sets.)

(we ignore noise in this lecture for simplicity)



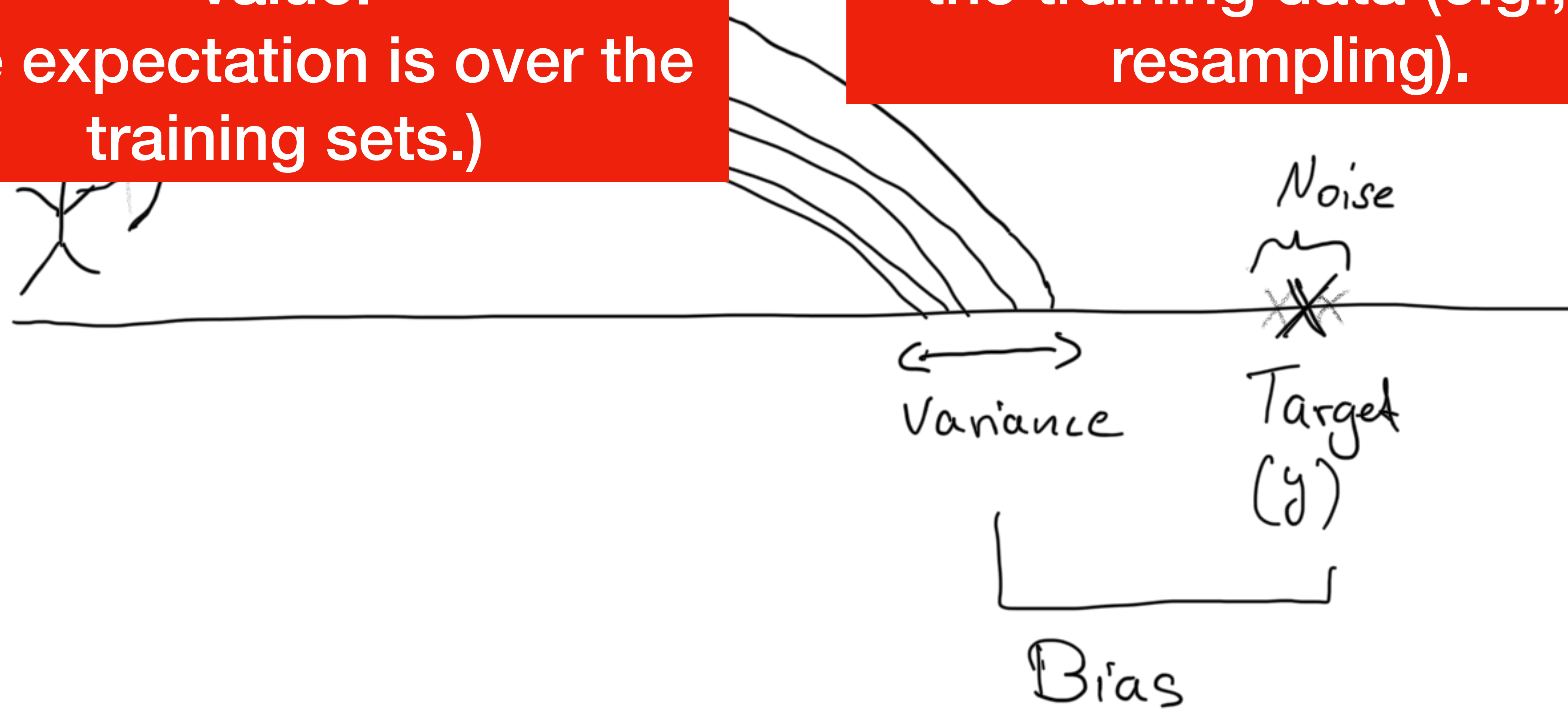
Terminology

$$\text{Bias}[\hat{\theta}] = E[\hat{\theta}] - \theta$$

$$\text{Var}[\hat{\theta}] = E\left[(E[\hat{\theta}] - \hat{\theta})^2\right]$$

Bias is the difference between the average estimator from different training samples and the true value.
(The expectation is over the training sets.)

The variance provides an estimate of how much the estimate varies as we vary the training data (e.g., by resampling).



Bias-Variance Decomposition

Loss = Bias + Variance + Noise

8.1 Overfitting and Underfitting

8.2 Intro to Bias-Variance Decomposition

8.3 Bias-Variance Decomposition of the Squared Error

8.4 Relationship between Bias-Variance Decomposition and Overfitting and Underfitting

8.5 Bias-Variance Decomposition of the 0/1 Loss

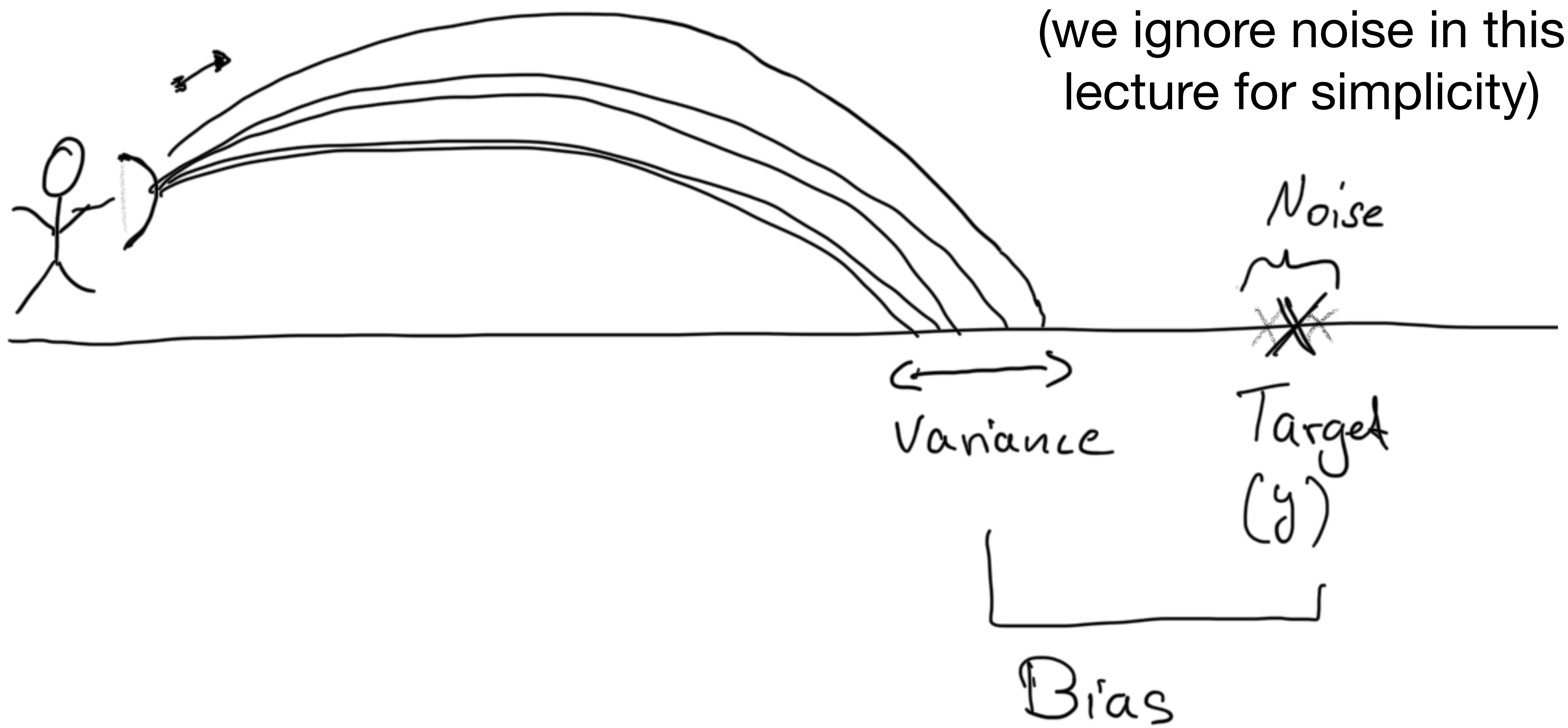
8.6 Other Forms of Bias

Terminology

$$\text{Bias}[\hat{\theta}] = E[\hat{\theta}] - \theta \qquad \text{Var}[\hat{\theta}] = E\left[(E[\hat{\theta}] - \hat{\theta})^2\right]$$

Intuition

(we ignore noise in this lecture for simplicity)



Bias-Variance Decomposition

Loss = Bias + Variance + Noise

Bias-Variance of the Squared Error

Bias$[\hat{\theta}] = E[\hat{\theta}] - \theta$

Var$[\hat{\theta}] = E[\hat{\theta}^2] - (E[\hat{\theta}])^2$

Var$[\hat{\theta}] = E\left[(E[\hat{\theta}] - \hat{\theta})^2\right]$

"ML Notation" for Squared Error Loss

$y = f(x)$ target

$\hat{y} = \hat{f}(x) = h(x)$ prediction

$S = (y - \hat{y})^2$ squared error

for simplicity, we ignore the noise term

(Next slides: the expectation is over the training data, i.e, the average estimator from different training samples)

Bias-Variance of the Squared Error

"ML Notation" for Squared Error Loss

$y = f(x)$ target

$\hat{y} = \hat{f}(x) = h(x)$ prediction

$S = (y - \hat{y})^2$ squared error

$$S = (y - \hat{y})^2$$

$$\begin{aligned}(y - \hat{y})^2 &= (y - E[\hat{y}] + E[\hat{y}] - \hat{y})^2 \\ &= (y - E[\hat{y}])^2 + (E[\hat{y}] - \hat{y})^2 - 2(y - E[\hat{y}])(E[\hat{y}] - \hat{y})\end{aligned}$$

Bias-Variance of the Squared Error

$$S = (y - \hat{y})^2$$

$$\begin{aligned}(y - \hat{y})^2 &= (y - E[\hat{y}] + E[\hat{y}] - \hat{y})^2 \\ &= (y - E[\hat{y}])^2 + (E[\hat{y}] - \hat{y})^2 + 2(y - E[\hat{y}])(E[\hat{y}] - \hat{y})\end{aligned}$$

$$E[S] = E\left[(y - \hat{y})^2\right]$$

$$E\left[(y - \hat{y})^2\right] = (y - E[\hat{y}])^2 + E\left[(E[\hat{y}] - \hat{y})^2\right]$$

$$= \text{Bias}^2 + \text{Var}$$

Bias-Variance of the Squared Error

$$S = (y - \hat{y})^2$$

$$\begin{aligned}(y - \hat{y})^2 &= (y - E[\hat{y}] + E[\hat{y}] - \hat{y})^2 \\ &= (y - E[\hat{y}])^2 + (E[\hat{y}] - \hat{y})^2 + 2(y - E[\hat{y}])(E[\hat{y}] - \hat{y})\end{aligned}$$

$$E[S] = E\left[(y - \hat{y})^2\right]$$

$$E\left[(y - \hat{y})^2\right] = (y - E[\hat{y}])^2 + E\left[(E[\hat{y}] - \hat{y})^2\right]$$

$$= \text{Bias}^2 + \text{Var}$$

$$\text{Bias}[\hat{\theta}] = E[\hat{\theta}] - \theta$$

$$\text{Var}[\hat{\theta}] = E[\hat{\theta}^2] - (E[\hat{\theta}])^2$$

$$\text{Var}[\hat{\theta}] = E\left[(E[\hat{\theta}] - \hat{\theta})^2\right]$$

Bias-Variance of the Squared Error

$$S = (y - \hat{y})^2$$

$$\begin{aligned}(y - \hat{y})^2 &= (y - E[\hat{y}] + E[\hat{y}] - \hat{y})^2 \\ &= (y - E[\hat{y}])^2 + (E[\hat{y}] - \hat{y})^2 \overset{???}{\boxed{- 2(y - E[\hat{y}])(E[\hat{y}] - \hat{y})}}\end{aligned}$$

Bias-Variance of the Squared Error

$$S = (y - \hat{y})^2$$

$$\begin{aligned}(y - \hat{y})^2 &= (y - E[\hat{y}] + E[\hat{y}] - \hat{y})^2 \\ &= (y - E[\hat{y}])^2 + (E[\hat{y}] - \hat{y})^2 \boxed{- 2(y - E[\hat{y}])(E[\hat{y}] - \hat{y})}\end{aligned}$$

???

$$\begin{aligned}E[2(y - E[\hat{y}])(E[\hat{y}] - \hat{y})] &= 2E[(y - E[\hat{y}])(E[\hat{y}] - \hat{y})] \\ &= 2(y - E[\hat{y}])E[(E[\hat{y}] - \hat{y})] \\ &= 2(y - E[\hat{y}])(E[E[\hat{y}]] - E[\hat{y}]) \\ &= 2(y - E[\hat{y}])(E[\hat{y}] - E[\hat{y}]) \\ &= 0\end{aligned}$$

```python
from mlxtend.evaluate import bias_variance_decomp
```

```python
from mlxtend.evaluate import bias_variance_decomp
from sklearn.tree import DecisionTreeRegressor
from mlxtend.data import boston_housing_data
from sklearn.model_selection import train_test_split


X, y = boston_housing_data()
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.3,
                                                    random_state=123,
                                                    shuffle=True)



tree = DecisionTreeRegressor(random_state=123)

avg_expected_loss, avg_bias, avg_var = bias_variance_decomp(
        tree, X_train, y_train, X_test, y_test,
        loss='mse',
        random_seed=123)

print('Average expected loss: %.3f' % avg_expected_loss)
print('Average bias: %.3f' % avg_bias)
print('Average variance: %.3f' % avg_var)
```

```
Average expected loss: 31.917
Average bias: 13.814
Average variance: 18.102
```

http://rasbt.github.io/mlxtend/user_guide/evaluate/bias_variance_decomp/

```python
from mlxtend.evaluate import bias_variance_decomp
```

```python
from sklearn.ensemble import BaggingRegressor

tree = DecisionTreeRegressor(random_state=123)
bag = BaggingRegressor(base_estimator=tree,
                       n_estimators=100,
                       random_state=123)

avg_expected_loss, avg_bias, avg_var = bias_variance_decomp(
        bag, X_train, y_train, X_test, y_test,
        loss='mse',
        random_seed=123)

print('Average expected loss: %.3f' % avg_expected_loss)
print('Average bias: %.3f' % avg_bias)
print('Average variance: %.3f' % avg_var)
```

```
Average expected loss: 18.593
Average bias: 15.354
Average variance: 3.239
```

http://rasbt.github.io/mlxtend/user_guide/evaluate/bias_variance_decomp/

Source code:
https://github.com/rasbt/mlxtend/blob/master/mlxtend/evaluate/bias_variance_decomp.py

```
...
    rng = np.random.RandomState(random_seed)

    all_pred = np.zeros((num_rounds, y_test.shape[0]), dtype=np.int)

    for i in range(num_rounds):

        X_boot, y_boot = _draw_bootstrap_sample(rng, X_train, y_train)

        if estimator.__class__.__name__ == 'Sequential':

            estimator.fit(X_boot, y_boot)

            pred = estimator.predict(X_test).reshape(1, -1)

        else:

            pred = estimator.fit(X_boot, y_boot).predict(X_test)

        all_pred[i] = pred
...
        avg_expected_loss = np.apply_along_axis(

            lambda x:

            ((x - y_test)**2).mean(),

            axis=1,

            arr=all_pred).mean()


        main_predictions = np.mean(all_pred, axis=0)

        avg_bias = np.sum((main_predictions - y_test)**2) / y_test.size

        avg_var = np.sum((main_predictions - all_pred)**2) / all_pred.size
...
```

$$\text{Bias}[\hat{\theta}] = E[\hat{\theta}] - \theta$$

$$\text{Var}[\hat{\theta}] = E[\hat{\theta}^2] - (E[\hat{\theta}])^2$$

$$\text{Var}[\hat{\theta}] = E\left[(E[\hat{\theta}] - \hat{\theta})^2\right]$$

$$E[S] = E\left[(y - \hat{y})^2\right]$$

$$= \text{Bias}^2 + \text{Var}$$

8.1 Overfitting and Underfitting

8.2 Intro to Bias-Variance Decomposition

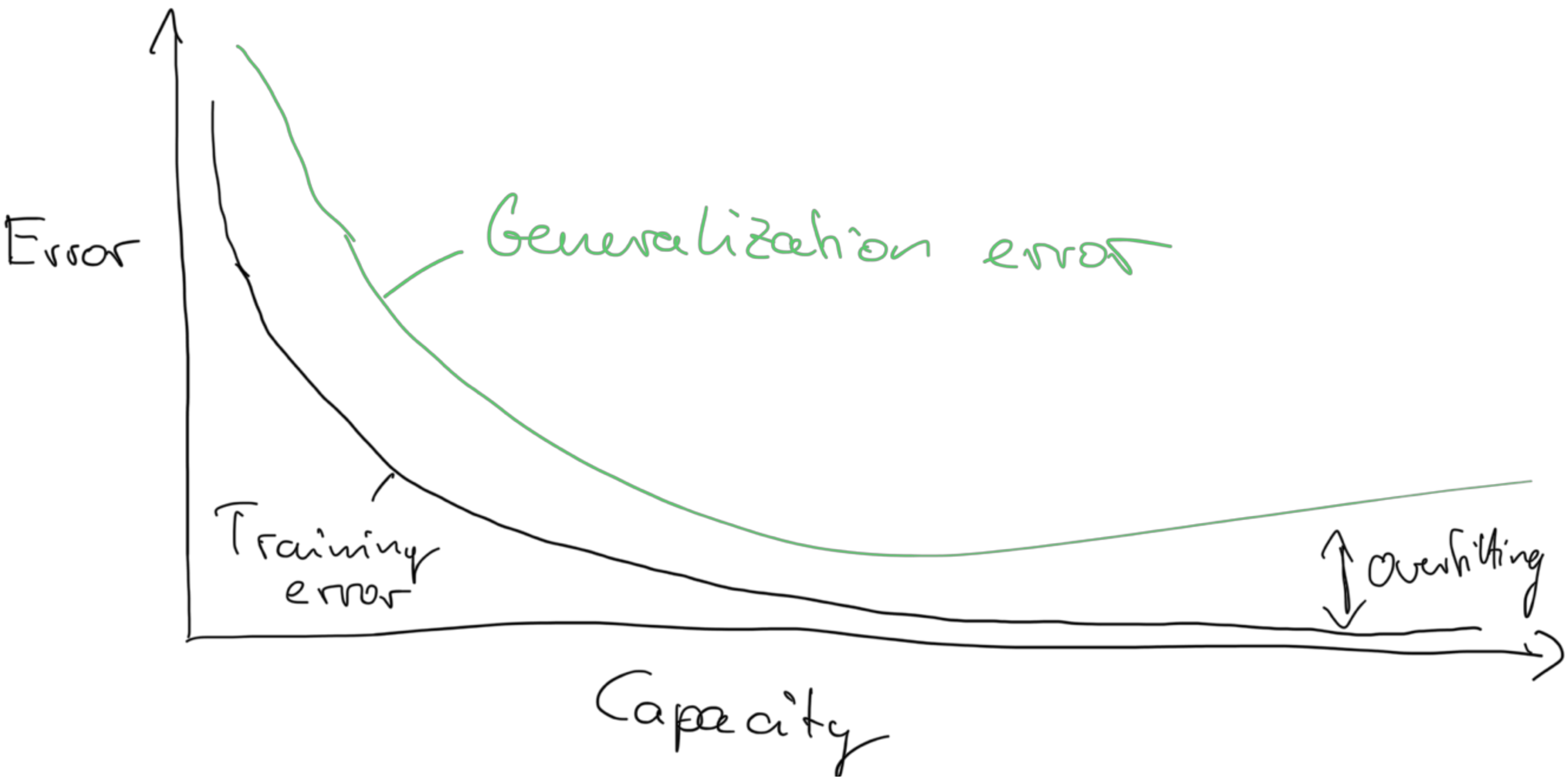
8.3 Bias-Variance Decomposition of the Squared Error

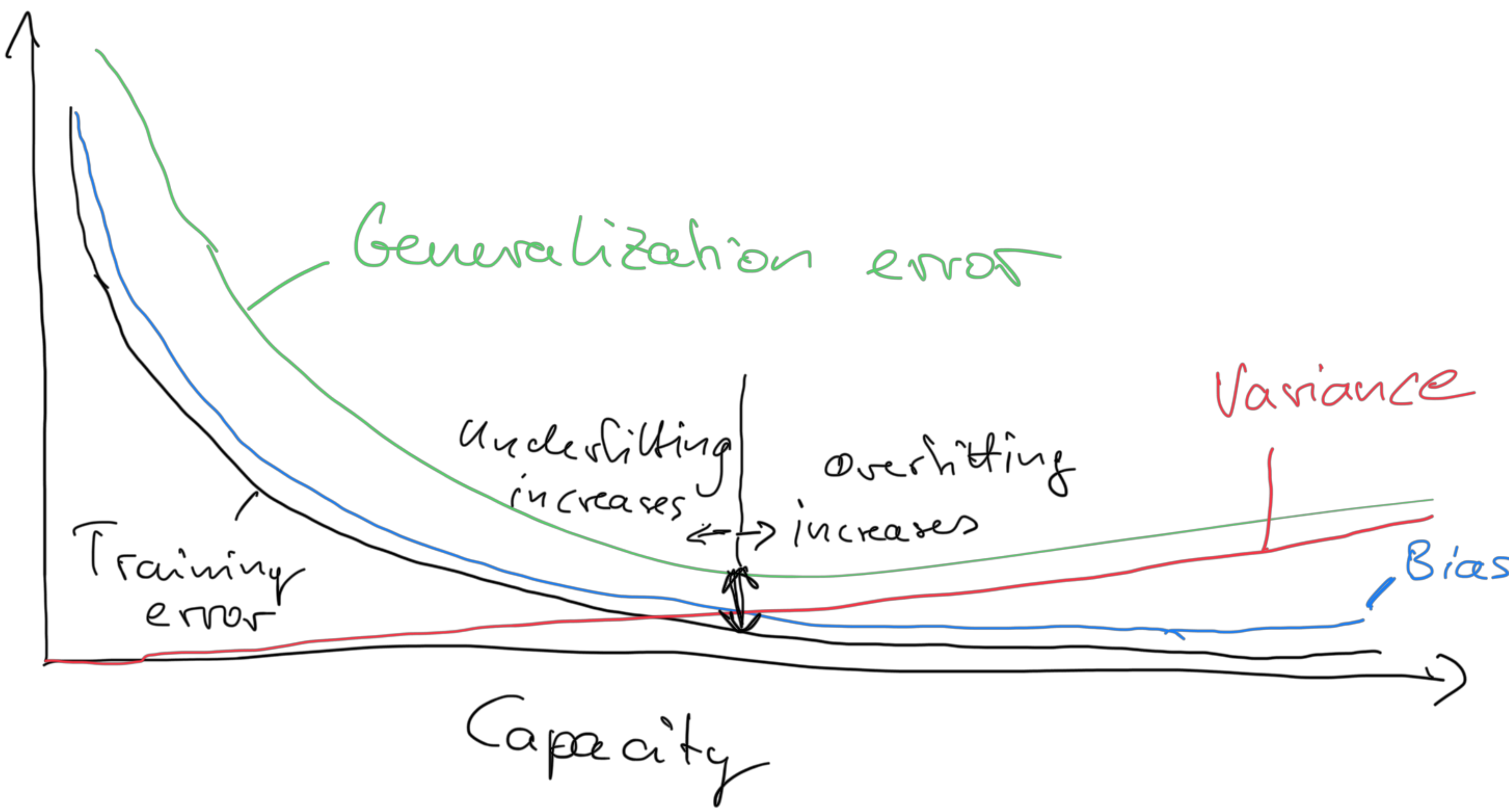
8.4 Relationship between Bias-Variance Decomposition and Overfitting and Underfitting

8.5 Bias-Variance Decomposition of the 0/1 Loss

8.6 Other Forms of Bias

Now, how is this related to overfitting and underfitting?


Error
Generalization error
Training error
overfitting
Capacity



Generalization error
Variance
underfitting increases
overfitting increases
Training error
Bias
Capacity

8.1 Overfitting and Underfitting

8.2 Intro to Bias-Variance Decomposition

8.3 Bias-Variance Decomposition of the Squared Error

8.4 Relationship between Bias-Variance Decomposition and Overfitting and Underfitting

8.5 Bias-Variance Decomposition of the 0/1 Loss

8.6 Other Forms of Bias

How can we think of the bias-variance decomposition in the context of the classification error (0/1 loss)?

Domingos, P. (2000). *A unified bias-variance decomposition.* In Proceedings of 17th International Conference on Machine Learning (pp. 231-238).

"several authors have proposed bias-variance decompositions related to zero-one loss (Kong & Dietterich, 1995; Breiman, 1996b; Kohavi & Wolpert, 1996; Tibshirani, 1996; Friedman, 1997). However, each of these decompositions has significant shortcomings."

Bias-Variance Decomposition of 0-1 Loss

Dietterich, T. G., & Kong, E. B. (1995). *Machine learning bias, statistical bias, and statistical variance of decision tree algorithms*. Technical report, Department of Computer Science, Oregon State University.

Domingos, P. (2000). A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning* (pp. 231-238).

Squared Loss

$$(y - \hat{y})^2$$

$$E[(y - \hat{y})^2]$$

Generalized Loss

$$L(y, \hat{y})$$

$$E[L(y, \hat{y})]$$

Bias-Variance Decomposition of 0-1 Loss

Dietterich, T. G., & Kong, E. B. (1995). *Machine learning bias, statistical bias, and statistical variance of decision tree algorithms*. Technical report, Department of Computer Science, Oregon State University.

Domingos, P. (2000). A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning* (pp. 231-238).

Squared Loss

$$(y - \hat{y})^2$$

$$E[(y - \hat{y})^2]$$

$$E[(y - \hat{y})^2] = (y - E[\hat{y}])^2 + E[(E[\hat{y}] - \hat{y})^2]$$

Bias^2 + Variance

Generalized Loss

$$L(y, \hat{y})$$

$$E[L(y, \hat{y})]$$

Bias-Variance Decomposition of 0-1 Loss

Dietterich, T. G., & Kong, E. B. (1995). *Machine learning bias, statistical bias, and statistical variance of decision tree algorithms*. Technical report, Department of Computer Science, Oregon State University.

Domingos, P. (2000). A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning* (pp. 231-238).

Squared Loss	Generalized Loss
$(y - \hat{y})^2$	$L(y, \hat{y})$
$E[(y - \hat{y})^2]$	$E[L(y, \hat{y})]$

$$E[(y - \hat{y})^2] = \underbrace{(y - E[\hat{y}])^2}_{\text{Bias}^2} + \underbrace{E[(E[\hat{y}] - \hat{y})^2]}_{\text{Variance}}$$

	Squared Loss	Generalized Loss
Bias^2:	$(y - E[\hat{y}])^2$	$L(y, E[\hat{y}])$
Variance:	$E[(E[\hat{y}] - \hat{y})^2]$	$E[L(\hat{y}, E[\hat{y}])]$

Define "Main Prediction"

Dietterich, T. G., & Kong, E. B. (1995). *Machine learning bias, statistical bias, and statistical variance of decision tree algorithms*. Technical report, Department of Computer Science, Oregon State University.

Domingos, P. (2000). A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning* (pp. 231-238).

The main prediction is the prediction that minimizes the average loss

$$\bar{\hat{y}} = \underset{\hat{y}'}{\operatorname{argmin}} \; E[L(\hat{y}, \hat{y}')]$$

For squared loss -> Mean

For 0-1 loss -> Mode

Bias-Variance Decomposition of 0-1 Loss

Dietterich, T. G., & Kong, E. B. (1995). *Machine learning bias, statistical bias, and statistical variance of decision tree algorithms*. Technical report, Department of Computer Science, Oregon State University.

Domingos, P. (2000). A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning* (pp. 231-238).

Squared Loss

$$(y - \hat{y})^2$$

$$E[(y - \hat{y})^2]$$

$$E[(y - \hat{y})^2] = \underbrace{(y - E[\hat{y}])^2}_{\text{Bias}^2} + \underbrace{E[(E[\hat{y}] - \hat{y})^2]}_{\text{Variance}}$$

Main prediction -> Mean

Bias^2: $(y - \boxed{E[\hat{y}]})^2$

Variance: $E[(E[\hat{y}] - \hat{y})^2]$

0-1 Loss

$$L(y, \hat{y})$$

$$E[L(y, \hat{y})]$$

Main prediction -> Mode

$$L(y, \boxed{E[\hat{y}]})$$

$$E[L(\hat{y}, E[\hat{y}])]$$

Bias-Variance Decomposition of 0-1 Loss

Dietterich, T. G., & Kong, E. B. (1995). *Machine learning bias, statistical bias, and statistical variance of decision tree algorithms*. Technical report, Department of Computer Science, Oregon State University.

Domingos, P. (2000). A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning* (pp. 231-238).

Squared Loss

$$E[(y - \hat{y})^2]$$

Main prediction -> Mean

Bias²: $(y - \boxed{E[\hat{y}])}^2$

Variance: $E[(E[\hat{y}] - \hat{y})^2]$

0-1 Loss

$$E[L(y, \hat{y})]$$

$$P(y \neq \hat{y})$$

Main prediction -> Mode

$$L(y, \boxed{E[\hat{y}])}$$

$$Bias = \begin{cases} 1 \text{ **if** } y \neq \bar{\hat{y}} \\ 0 \text{ **otherwise**} \end{cases}$$

$$E[L(\hat{y}, E[\hat{y}])]$$

$$Variance = P(\hat{y} \neq \hat{\bar{y}})$$

Bias-Variance Decomposition of 0-1 Loss

Dietterich, T. G., & Kong, E. B. (1995). *Machine learning bias, statistical bias, and statistical variance of decision tree algorithms*. Technical report, Department of Computer Science, Oregon State University.

Domingos, P. (2000). A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning* (pp. 231-238).

0-1 Loss

Loss = Bias + Variance = $P(\hat{y} \neq y)$

$$Bias = \begin{cases} 1 \textbf{ if } y \neq \bar{\hat{y}} \\ 0 \textbf{ otherwise} \end{cases}$$

Loss = Variance = $P(\hat{y} \neq y)$

$$Variance = P(\hat{y} \neq \bar{\hat{y}})$$

Bias-Variance Decomposition of 0-1 Loss

Dietterich, T. G., & Kong, E. B. (1995). *Machine learning bias, statistical bias, and statistical variance of decision tree algorithms*. Technical report, Department of Computer Science, Oregon State University.

Domingos, P. (2000). A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning* (pp. 231-238).

0-1 Loss

Loss = $P(\hat{y} \neq y)$

$$Bias = \begin{cases} 1 \textbf{ if } y \neq \bar{\hat{y}} \\ 0 \textbf{ otherwise} \end{cases}$$

Loss = $P(\hat{y} \neq y) = 1 - P(\hat{y} = y) = 1 - P(\hat{y} \neq \bar{\hat{y}})$

Bias-Variance Decomposition of 0-1 Loss

Dietterich, T. G., & Kong, E. B. (1995). *Machine learning bias, statistical bias, and statistical variance of decision tree algorithms*. Technical report, Department of Computer Science, Oregon State University.

Domingos, P. (2000). A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning* (pp. 231-238).

0-1 Loss

Loss = $P(\hat{y} \neq y)$

$$Bias = \begin{cases} 1 \textbf{ if } y \neq \bar{\hat{y}} \\ 0 \textbf{ otherwise} \end{cases}$$

Loss = $P(\hat{y} \neq y) = 1 - P(\hat{y} = y) = 1 - P(\hat{y} \neq \bar{\hat{y}})$

Bias-Variance Decomposition of 0-1 Loss

Dietterich, T. G., & Kong, E. B. (1995). *Machine learning bias, statistical bias, and statistical variance of decision tree algorithms*. Technical report, Department of Computer Science, Oregon State University.

Domingos, P. (2000). A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning* (pp. 231-238).

0-1 Loss

Loss = $P(\hat{y} \neq y)$

$$Bias = \begin{cases} 1 \textbf{ if } y \neq \bar{\hat{y}} \\ 0 \textbf{ otherwise} \end{cases}$$

Loss = $P(\hat{y} \neq y) = 1 - P(\hat{y} = y) = 1 - P(\hat{y} \neq \bar{\hat{y}})$

Loss = Bias - Variance

Bias-Variance Decomposition of 0-1 Loss

Dietterich, T. G., & Kong, E. B. (1995). *Machine learning bias, statistical bias, and statistical variance of decision tree algorithms*. Technical report, Department of Computer Science, Oregon State University.

Domingos, P. (2000). A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning* (pp. 231-238).

0-1 Loss

Loss = $P(\hat{y} \neq y)$

$$Bias = \begin{cases} 1 \textbf{ if } y \neq \bar{\hat{y}} \\ 0 \textbf{ otherwise} \end{cases}$$

Variance can improve loss!!
Why is that so?

Loss = $P(\hat{y} \neq y) = 1 - P(\hat{y} = y) = 1 - P(\hat{y} \neq \bar{\hat{y}})$

Loss = Bias - Variance

Recommended Reading Resources for Bias-Decomposition

Dietterich, T. G., & Kong, E. B. (1995). *Machine learning bias, statistical bias, and statistical variance of decision tree algorithms*. Technical report, Department of Computer Science, Oregon State University.

0-1 loss

Domingos, P. (2000). A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning* (pp. 231-238).

includes noise
and more general: Loss = Bias + *c* Variance

or more precisely $c_1 N(x) + B(x) + c_2 V(x)$

where, e.g., $c_1 = c_2 = 1$ for squared loss

```python
from mlxtend.evaluate import bias_variance_decomp
from sklearn.tree import DecisionTreeClassifier
from mlxtend.data import iris_data
from sklearn.model_selection import train_test_split


X, y = iris_data()
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.3,
                                                    random_state=123,
                                                    shuffle=True,
                                                    stratify=y)



tree = DecisionTreeClassifier(random_state=123)

avg_expected_loss, avg_bias, avg_var = bias_variance_decomp(
        tree, X_train, y_train, X_test, y_test,
        loss='0-1_loss',
        random_seed=123)

print('Average expected loss: %.3f' % avg_expected_loss)
print('Average bias: %.3f' % avg_bias)
print('Average variance: %.3f' % avg_var)
```

```
Average expected loss: 0.062
Average bias: 0.022
Average variance: 0.040
```

```python
from sklearn.ensemble import BaggingClassifier

tree = DecisionTreeClassifier(random_state=123)
bag = BaggingClassifier(base_estimator=tree,
                        n_estimators=100,
                        random_state=123)

avg_expected_loss, avg_bias, avg_var = bias_variance_decomp(
        bag, X_train, y_train, X_test, y_test,
        loss='0-1_loss',
        random_seed=123)

print('Average expected loss: %.3f' % avg_expected_loss)
print('Average bias: %.3f' % avg_bias)
print('Average variance: %.3f' % avg_var)
```

```
Average expected loss: 0.048
Average bias: 0.022
Average variance: 0.026
```

```python
    all_pred = np.zeros((num_rounds, y_test.shape[0]), dtype=np.int)

    for i in range(num_rounds):
        X_boot, y_boot = _draw_bootstrap_sample(rng, X_train, y_train)
        if estimator.__class__.__name__ == 'Sequential':
            estimator.fit(X_boot, y_boot)
            pred = estimator.predict(X_test).reshape(1, -1)
        else:
            pred = estimator.fit(X_boot, y_boot).predict(X_test)
        all_pred[i] = pred

    if loss == '0-1_loss':
        main_predictions = np.apply_along_axis(lambda x:
                                               np.argmax(np.bincount(x)),
                                               axis=0,
                                               arr=all_pred)

        avg_expected_loss = np.apply_along_axis(lambda x:
                                                (x != y_test).mean(),
                                                axis=1,
                                                arr=all_pred).mean()

        avg_bias = np.sum(main_predictions != y_test) / y_test.size

        var = np.zeros(pred.shape)

        for pred in all_pred:
            var += (pred != main_predictions).astype(np.int)
        var /= num_rounds

        avg_var = var.sum()/y_test.shape[0]

    else:
        avg_expected_loss = np.apply_along_axis(
            lambda x:
            ((x - y_test)**2).mean(),
```

https://github.com/rasbt/mlxtend/blob/master/mlxtend/evaluate/bias_variance_decomp.py

8.1 Overfitting and Underfitting

8.2 Intro to Bias-Variance Decomposition

8.3 Bias-Variance Decomposition of the Squared Error

8.4 Relationship between Bias-Variance Decomposition and Overfitting and Underfitting

8.5 Bias-Variance Decomposition of the 0/1 Loss

8.6 Other Forms of Bias

Other "Biases"

Statistical Bias vs "Machine Learning Bias"

"Machine learning bias" sometimes also called "inductive bias"

e.g., decision tree algorithms consider small trees before they consider large trees

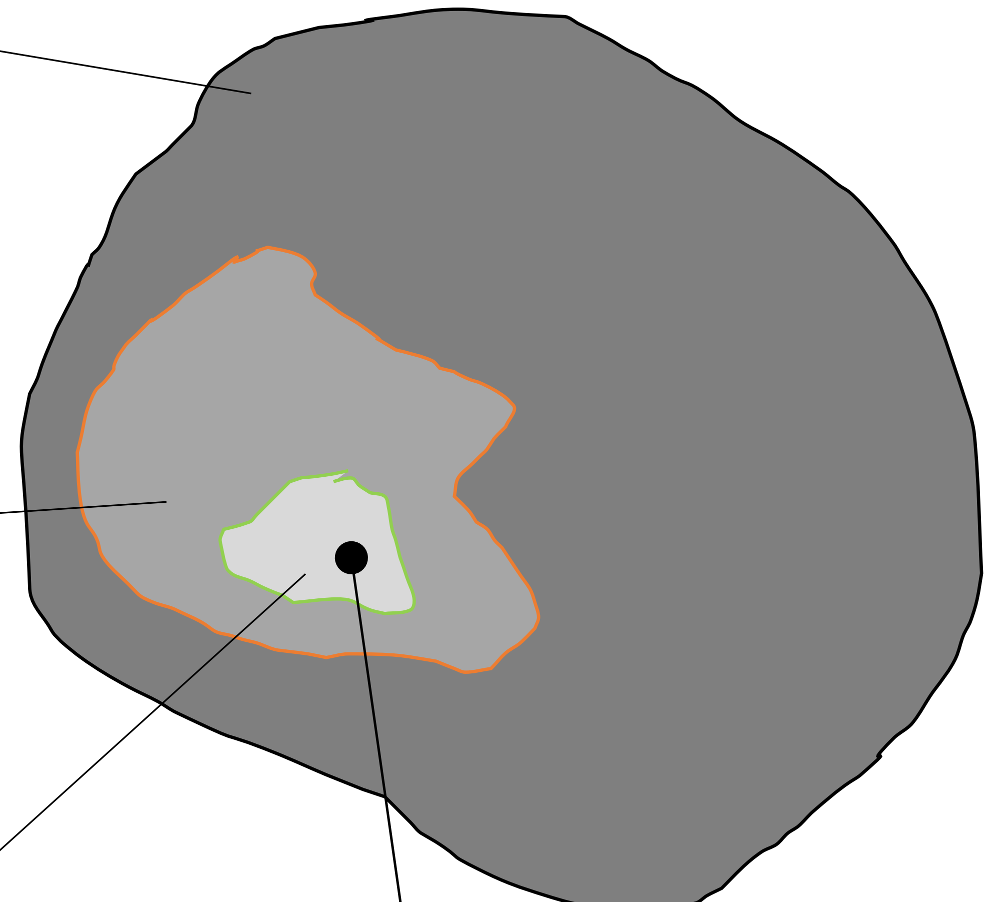
(if training data can be classified by small tree, large trees are not considered)

Hypothesis Space

(From Lecture 1)

Entire hypothesis space

e.g,. decision tree + KNN



Hypothesis space
a particular learning
algorithm category
has access to

e.g,. decision tree

Hypothesis space
a particular learning
algorithm can sample

e.g,. ID3

Particular hypothesis
(i.e., a model/classifier)

Dietterich, T. G., & Kong, E. B. (1995). *Machine learning bias, statistical bias, and statistical variance of decision tree algorithms*. Technical report, Department of Computer Science, Oregon State University.

Table 1: Relationship between ML bias and statistical bias and variance

ML Bias		Statistical	
Absolute	Relative	Bias	Variance
appropriate	too strong	high	low
appropriate	ok	low	low
appropriate	too weak	low	high
inappropriate	too strong	high	low
inappropriate	ok	high	moderate
inappropriate	too weak	high	high

bias can be characterized as appropriate or inappropriate. The hypothesis space of an inappropriate absolute bias does not contain any good approximations to the target function. An appropriate bias does contain good approximations.

A relative bias can be described as being too strong or too weak. A bias that is too strong is one that, though it may not rule out good approximations to the target function, prefers other, poorer hypotheses instead. A bias that is too weak does not focus the learning algorithm on the appropriate hypotheses but instead allows it to consider too many hypotheses.

Bias-Variance Simulation of C 4.5

Dietterich, T. G., & Kong, E. B. (1995). *Machine learning bias, statistical bias, and statistical variance of decision tree algorithms*. Technical report, Department of Computer Science, Oregon State University.

- simulation on 200 training sets with 200 examples each (0-1 labels)
  - 200 hypotheses

- test set: 22,801 examples (1 data point for each grid point)

- mean error rate is 536 errors (out of the 22,801 test examples)
  - 297 as a result of bias
  - 239 as a result of variance

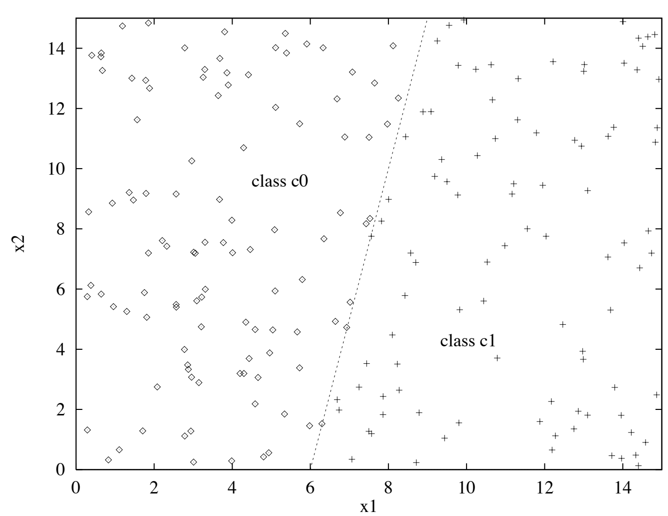


Figure 1: A two-class problem with 200 training examples.

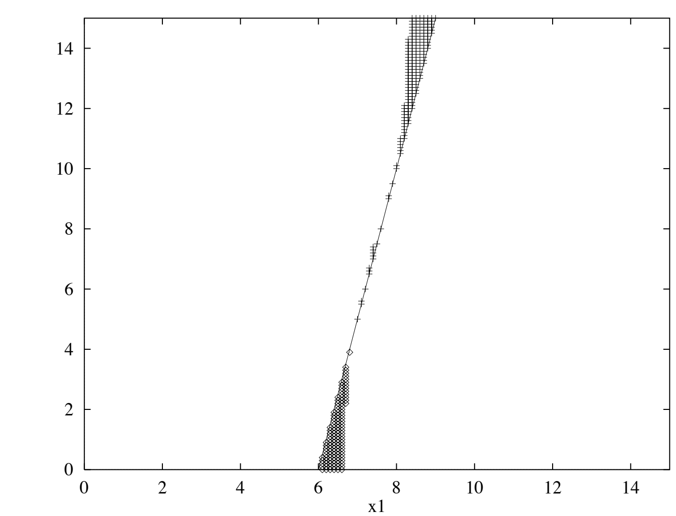
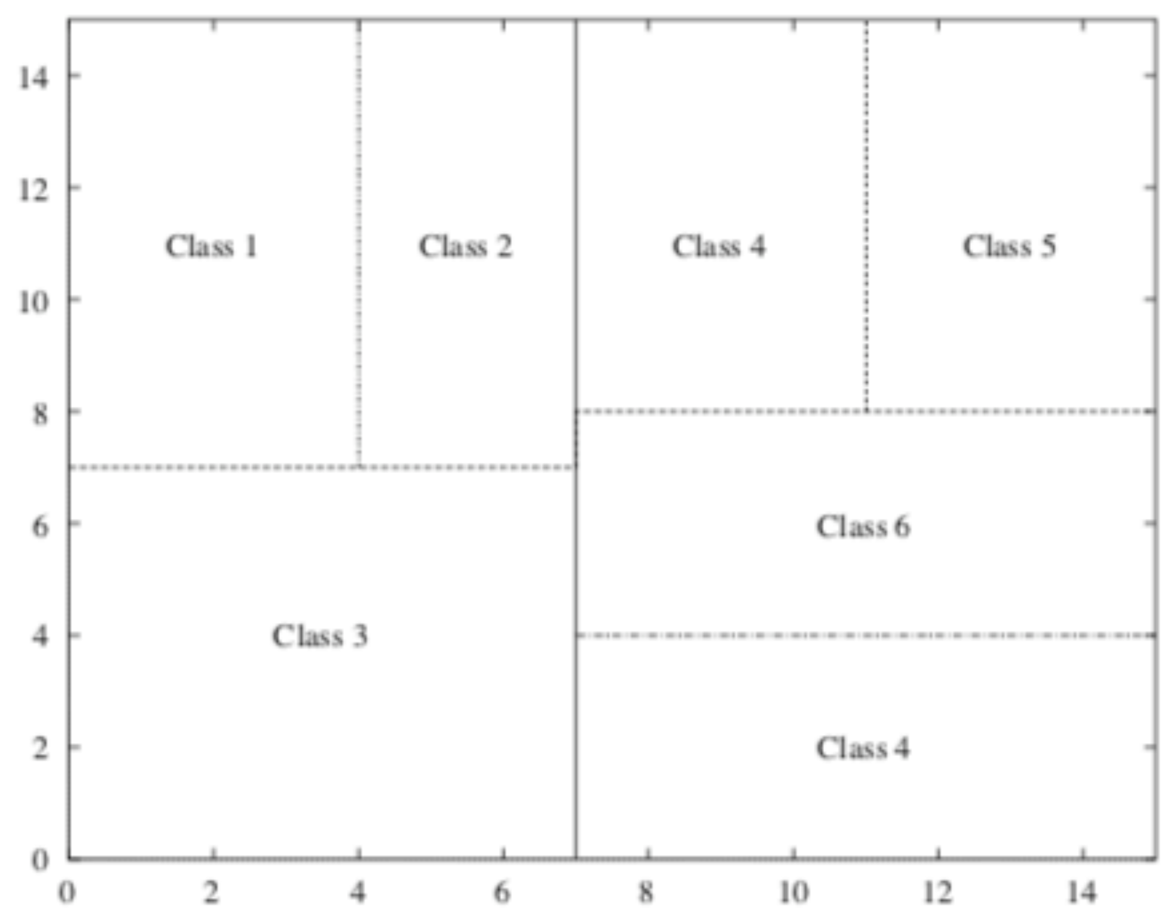


Figure 2: Bias errors of C4.5 on the problem from Figure 1.

(remember that trees use a "staircase" to approximate diagonal boundaries)

Bias-Variance Simulation of C 4.5

Dietterich, T. G., & Kong, E. B. (1995). *Machine learning bias, statistical bias, and statistical variance of decision tree algorithms*. Technical report, Department of Computer Science, Oregon State University.



errors due to bias: 0
errors due to variance: 17

ML Bias		Statistical	
Absolute	Relative	Bias	Variance
appropriate	too strong	high	low
appropriate	ok	low	low
appropriate	too weak	low	high
inappropriate	too strong	high	low
inappropriate	ok	high	moderate
inappropriate	too weak	high	high

"Fairness" Bias

"The term bias is often used to refer to demographic disparities in algorithmic systems that are objectionable for societal reasons. "

Barocas, S., Hardt, M., & Narayanan, A. Fairness and Machine Learning.
https://fairmlbook.org/introduction.html

G-COTS predictions on CelebA (<u>before</u> applying SAN)

Dark skin color		Predicted Male	Predicted Female
Actual	Male	0.548	0.016
Actual	Female	0.013	0.424

- P(wrong | male) = 0.028
- P(wrong | female) = 0.029

Light skin color		Predicted Male	Predicted Female
Actual	Male	0.399	0.013
Actual	Female	0.008	0.579

- P(wrong | male) = 0.032
- P(wrong | female) = 0.014

If dark skin is associated with a male gender attribute, we expect a <u>high prediction error</u> if someone is

Vahid Mirjalili, Sebastian Raschka, Anoop Namboodiri, and Arun Ross (2018) *Semi-adversarial Networks: Convolutional Autoencoders for Imparting Privacy to Face Images.* Proc. of 11th IAPR International Conference on Biometrics

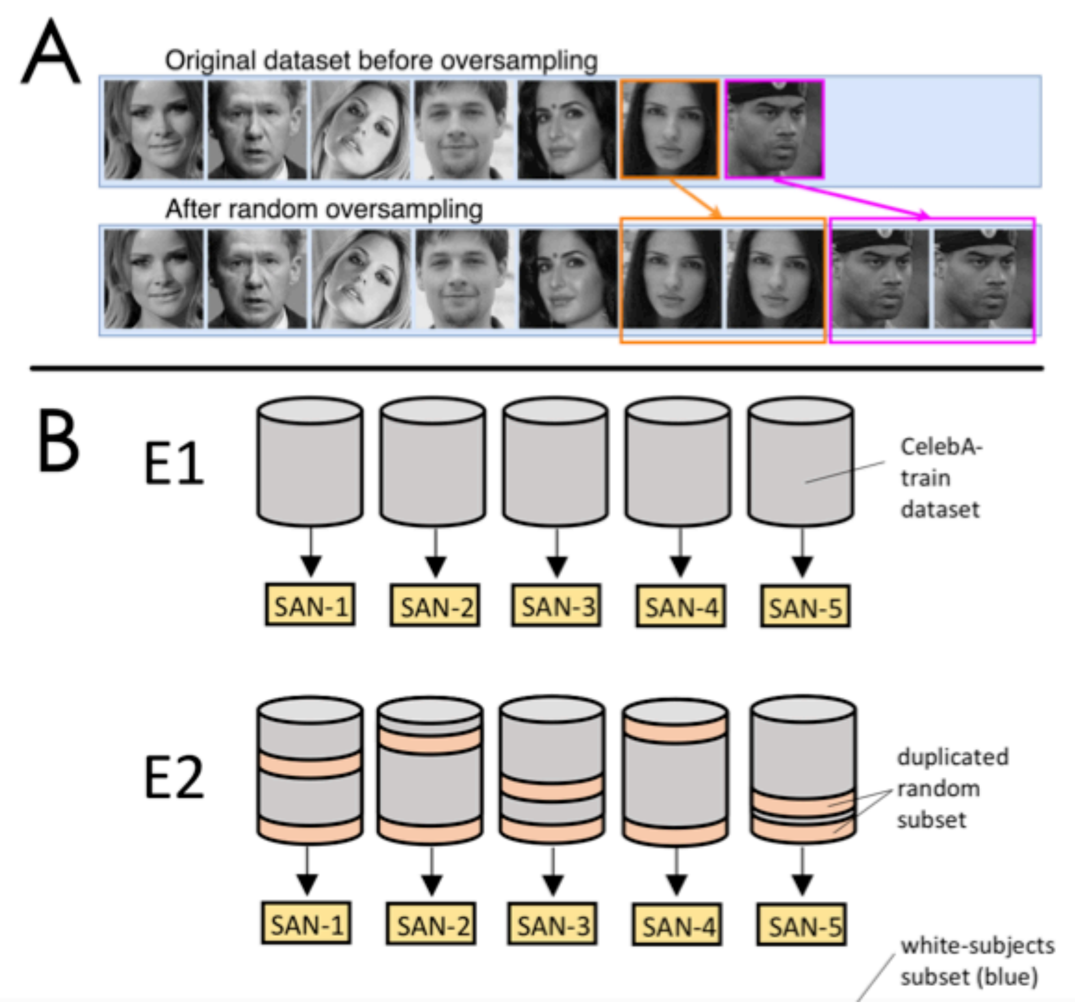
Vahid Mirjalili, Sebastian Raschka, and Arun Ross (2018) *Gender Privacy: An Ensemble of Semi Adversarial Networks for Confounding Arbitrary Gender Classifiers.* 9th IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS 2018)

Vahid Mirjalili, Sebastian Raschka, and Arun Ross (2019)
FlowSAN: Privacy-enhancing Semi-Adversarial Networks to Confound Arbitrary Face-based Gender Classifiers
IEEE Access 2019, 10.1109/ACCESS.2019.2924619

Vahid Mirjalili, Sebastian Raschka, and Arun Ross (2020)
PrivacyNet: Semi-Adversarial Networks for Multi-attribute Face Privacy
IEEE Transactions in Image Processing. Vol. 29, pp. 9400-9412, 2020

Figure 4: Face prototypes computed for each group of attribute labels. The abbreviations at the bottom of each image refer to the prototype attribute-classes, where Y=young, O=old, M=male, F=female, W=white, B=black.

groups. For each group, we generate a prototype image, which is the average of all face images from the training dataset that belong to that group. Hence, given eight distinct categories or groups, eight different prototypes are computed. Next, an opposite-attribute prototype is defined by flipping one of the binary attribute labels of an input im-



Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka (2020) *Rank-consistent Ordinal Regression for Neural Networks* https://arxiv.org/abs/1901.07884 (to appear in *Pattern Recognition Letters*)

Table 1. Age prediction errors on the test sets. All models are based on the ResNet-34 architecture.

Method	Random Seed	MORPH-2 MAE	MORPH-2 RMSE	AFAD MAE	AFAD RMSE	CACD MAE	CACD RMSE
CE-CNN	0	3.26	4.62	3.58	5.01	5.74	8.20
	1	3.36	4.77	3.58	5.01	5.68	8.09
	2	3.39	4.84	3.62	5.06	5.53	7.92
	AVG ± SD	3.34 ± 0.07	4.74 ± 0.11	3.60 ± 0.02	5.03 ± 0.03	5.65 ± 0.11	8.07 ± 0.14
OR-CNN (Niu et al., 2016)	0	2.87	4.08	3.56	4.80	5.36	7.61
	1	2.81	3.97	3.48	4.68	5.40	7.78
	2	2.82	3.87	3.50	4.78	5.37	7.70
	AVG ± SD	2.83 ± 0.03	3.97 ± 0.11	3.51 ± 0.04	4.75 ± 0.06	5.38 ± 0.02	7.70 ± 0.09
CORAL-CNN (ours)	0	2.66	3.69	3.42	4.65	5.25	7.41
	1	2.64	3.64	3.51	4.76	5.25	7.50
	2	2.62	3.62	3.48	4.73	5.24	7.52
	AVG ± SD	**2.64 ± 0.02**	**3.65 ± 0.04**	**3.47 ± 0.05**	**4.71 ± 0.06**	**5.25 ± 0.01**	**7.48 ± 0.06**