Multivariate Analysis, I
1st part

Aldo Solari

INFN School of Statistics

Paestum, June 6, 2019

About me

Aldo Solari
Associate Professor at the University of Milano-Bicocca
PhD in Statistics at the University of Padua
Homepage: aldosolari.github.io

Outline

What this lecture is not

A detailed math and statistics class, a programming course

What this lecture tries to be

An overview of statistical methods for prediction

Part 1 (1.5h)

- Statistical learning: supervised and unsupervised
- Model bias and variance and the risk of overfitting
- Cross-validation

Part 2 (1.5h)

- The model versus the modeling process
- Ensemble learning: bagging, random forest and boosting
- Regularized regression: ridge and lasso

If anything is unclear, just ask

Bibliography

- **Azzalini, Scarpa** (2012) *Data analysis and data mining, an introduction*. New York: Oxford University Press
- **Bishop** (2006) *Pattern Recognition and Machine Learning*. Springer
- **Gareth, Witten, Hastie, Tibshirani** (2014) *An Introduction to Statistical Learning, with Applications in R*. Springer
- **Hastie, Tibshirani, Friedman** (2009) *The Elements of Statistical Learning. Data Mining, Inference and Prediction*. Springer
- **Hastie, Tibshirani, Wainwright** (2015) *Statistical Learning with Sparsity. The Lasso and Generalizations*. Chapman and Hall/CRC
- **Kuhn, Johnson** (2013) *Applied Predictive Modelling*. Springer
- **Kuhn, Johnson** (2019) *Feature Engineering and Selection: A Practical Approach for Predictive Models*. Chapman and Hall/CRC

Evolution of multivariate statistics

Classic

- Multivariate Analysis

  Books by Anderson (1958) and Mardia, Kent & Bibby (1979)

- Statistical Modeling

  Nelder & Wedderburn (1972) paper on GLM

Computer-age

- Data Mining $\approx$ process of discovering patterns in data
- Machine Learning $\approx$ algorithms that can learn from data

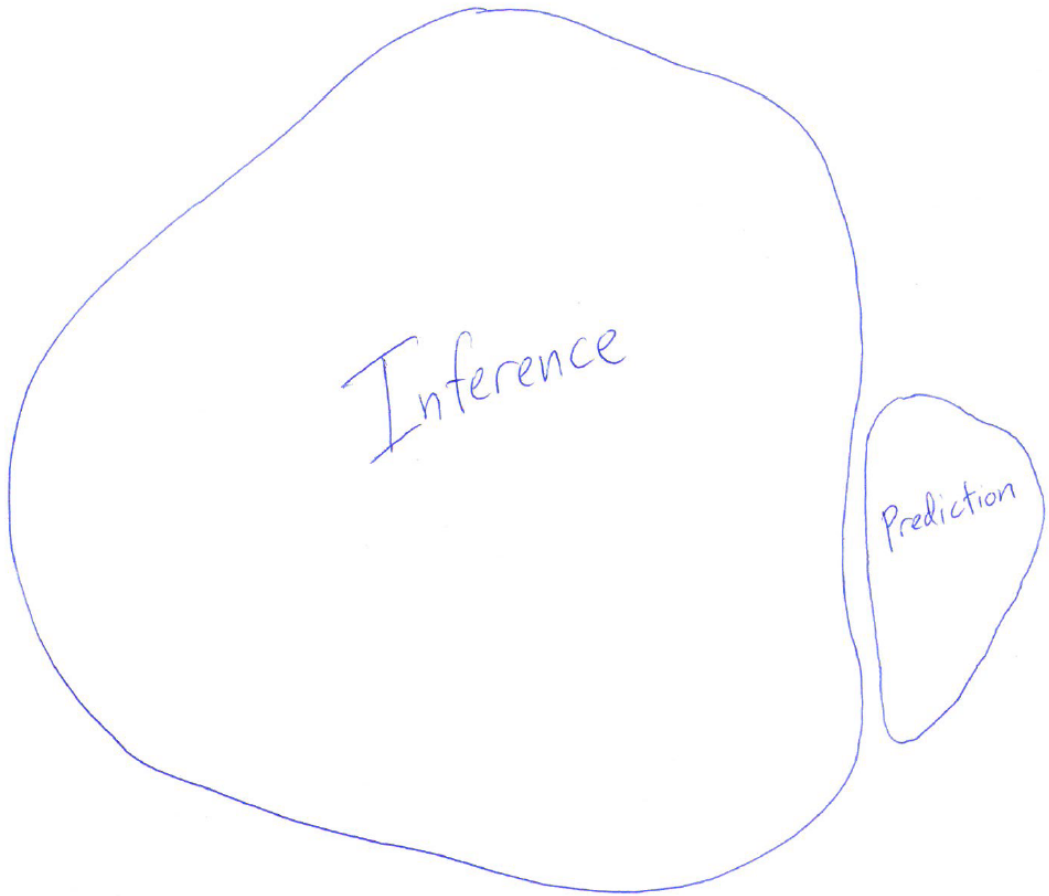
Modern

- Statistical Learning

  Hastie,Tibshirani & Friedman (2001) book

- Data Science

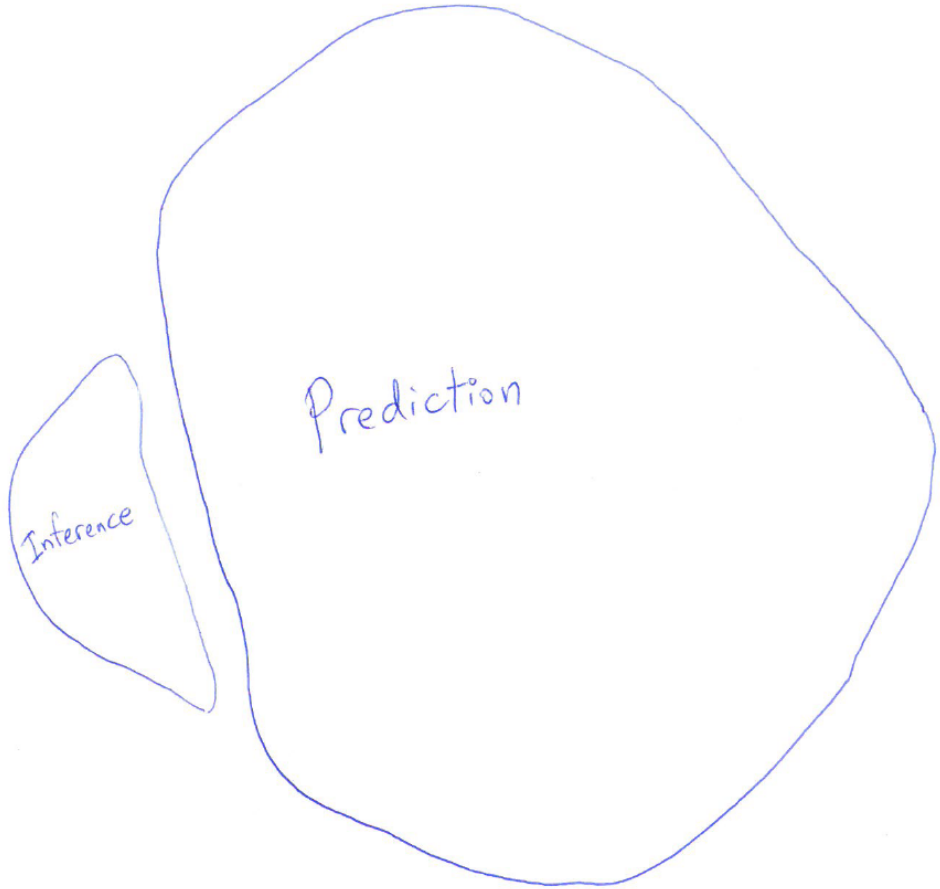
Statistics versus machine learning



How statisticians see the world?

From Robert Tibshirani

Statistics versus machine learning



How machine learners see the world?

From Robert Tibshirani

The two cultures

Breiman (2001)

Breiman distinguished between

Statistical modeling

- emphasis on probability models
- the goal is explanation

Machine learning

- emphasis on algorithms
- the goal is prediction accuracy

Today: Data Science vs. Statistics: Two Cultures?
Carmichael and Marron (2018)

To explain or to predict?

Shmueli (2010)

True model

$$Y = \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

Underspecified model

$$Y = \gamma_1 X_1 + \epsilon$$

Explanation requires estimating the coefficients of the true model, but *a wrong model can sometimes predict better* :

- when the predictors X_1 and X_2 are highly correlated
- when the data are very noisy
- when β_2 is small

Outline

Statistical learning

Unsupervised learning

- the data consists of variables $X_1, \ldots, X_p$; no variable has a special status
- the goal is
  - **dimensionality reduction**
  - **clustering**
  - etc.

Supervised learning

- the data consists of *response* Y and *predictors* $X_1, \ldots, X_p$
- the goal is **prediction** of the response:
  - Y continuous : **regression** problem
  - Y binary/multiclass: **classification** problem

Data matrix

	variable X_1	$\cdots$	variable X_j	$\cdots$	variable X_p
observation 1	x_{11}	$\cdots$	x_{1j}	$\cdots$	x_{1p}
observation 2	x_{21}	$\cdots$	x_{2j}	$\cdots$	x_{2p}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
observation i	x_{i1}	$\cdots$	x_{ij}	$\cdots$	x_{ip}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
observation n	x_{n1}	$\cdots$	x_{nj}	$\cdots$	x_{np}

- $n =$ number of observations
- $p =$ number of variables

Dimensionality reduction

$$\underset{n\times p}{X} \mapsto \underset{n\times q}{Y} \qquad q < p$$

such that the transformed data Y retains most of the information

Principal Components Analysis (PCA)

Linear transformation

$$Y_q = \tilde{X} V_q$$

where

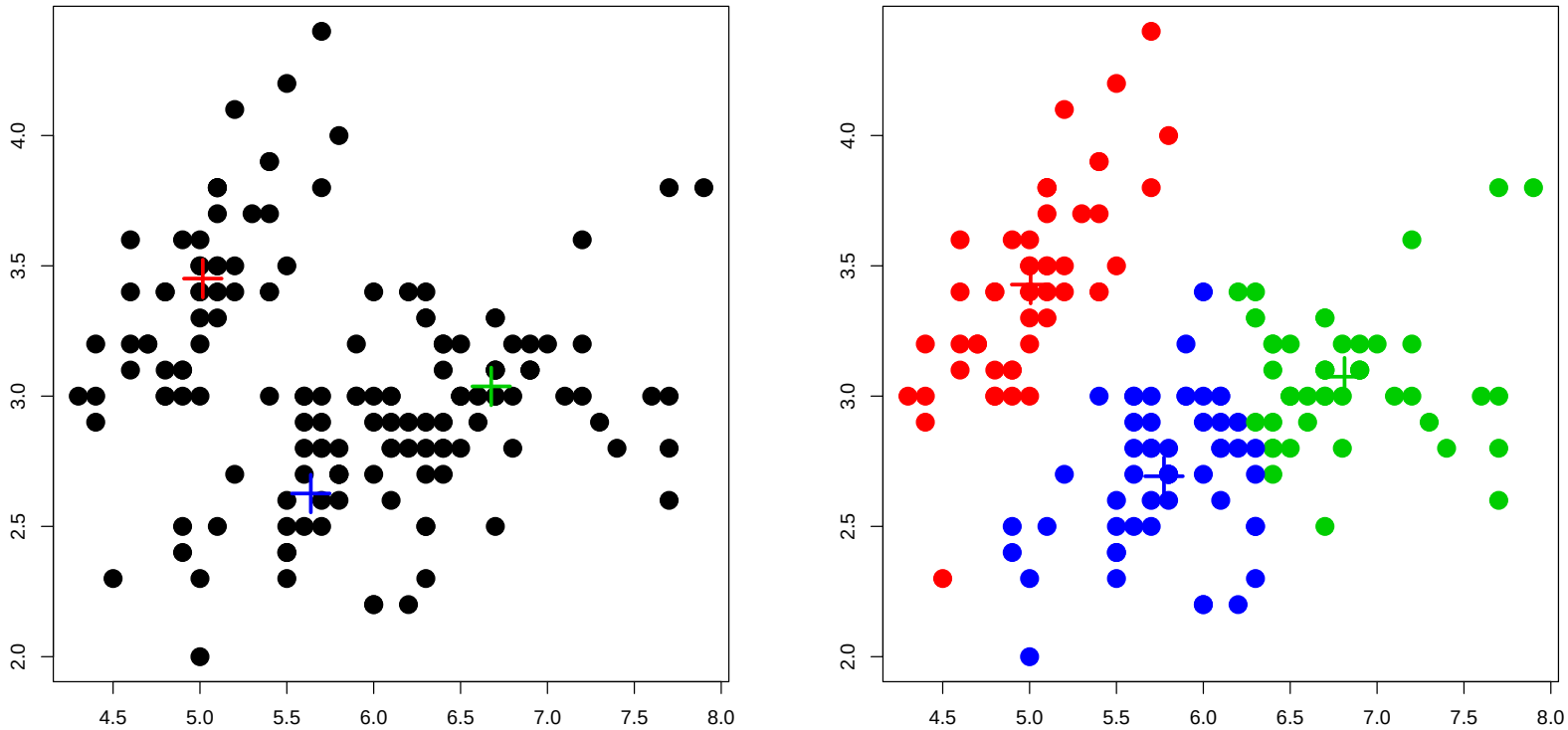
- $\tilde{X}$ is the centered matrix
- the columns of V_q are the q eigenvectors with largest eigenvalues of the variance/covariance matrix $n^{-1}\tilde{X}^\top \tilde{X}$

Image compression with PCA

$$\underset{243\times220}{X}$$

$$Y_q V_q^\top + 1\bar{x}^\top \quad q = 10$$

Cluster analysis



K-means algorithm

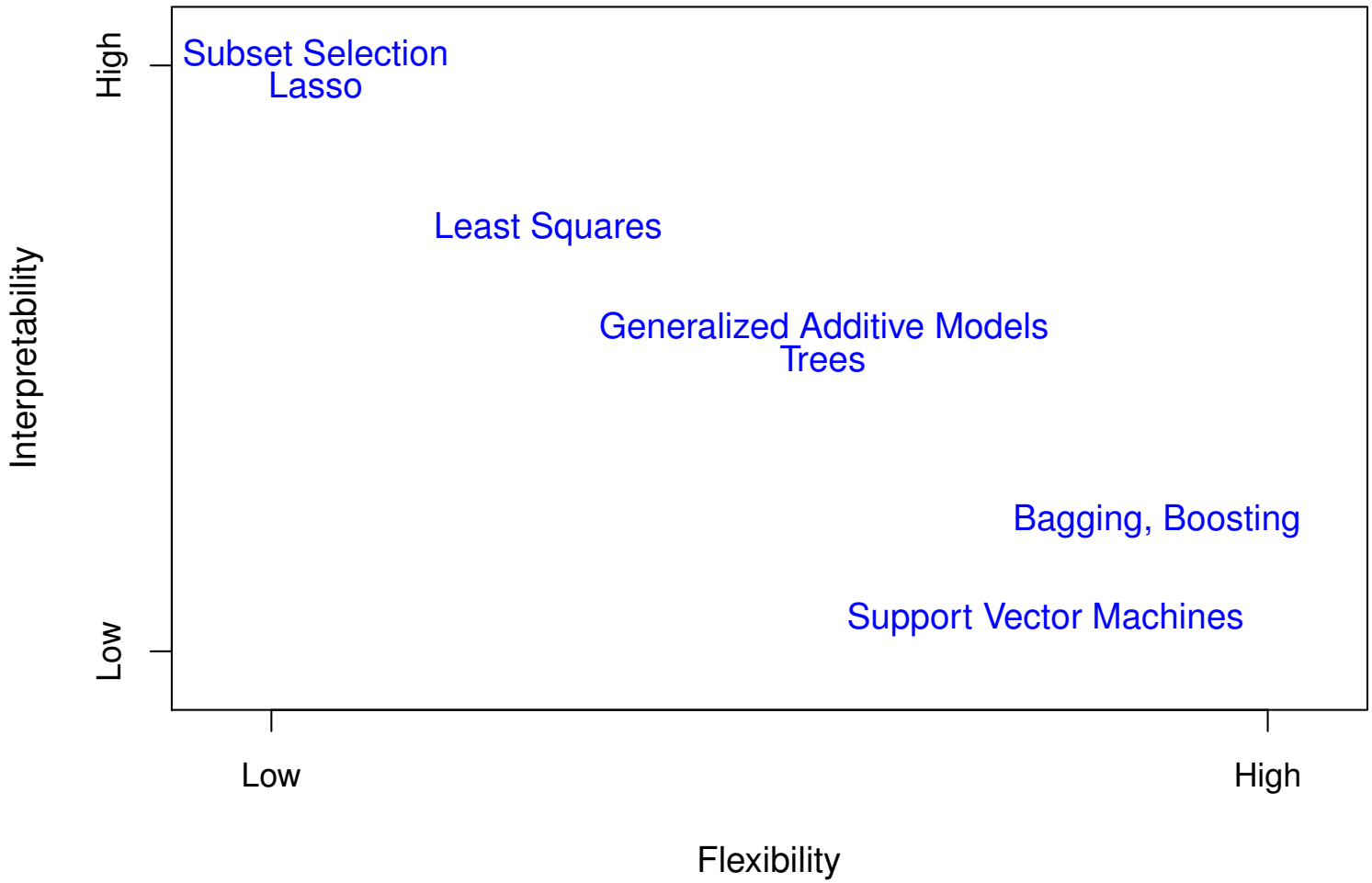
Supervised learning

Machine Learning		Statistics
target variable	Y	response variable
attribute, feature	X	predictor
hypothesis	$Y = f(X) + \varepsilon$	model
instances, examples	$(y_1, x_1), \ldots, (y_n, x_n)$	observations
learning	$\hat{f} = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \mathrm{loss}(y_i, f(x_i))$	estimation
classification	$\hat{y} = \hat{f}(x)$	prediction
generalization error	$\mathbb{E}[\,\mathrm{loss}(Y, \hat{f}(X))\,]$	prediction error

Models overview

Linear regression	✓
Logistic regression	✓
Linear discriminant analysis	✗
k-nearest neighbors	✓
Decision trees	✓
Naive Bayes	✗
Bagging	✓
Random forest	✓
Boosting	✓
Ridge regression	✓
Lasso	✓
Generalized additive models	✗
Support vector machines	✗
Neural networks	✗
Deep learning	✗

Flexibility versus interpretability



SUMMARY: statistical learning

- **Learning**: unsupervised / supervised
- **Goal**: to explain / to predict
- **Prediction problem**: regression / classification
- **Models**: interpretable / black-box

Important concepts

- Model bias and variance and the risk of overfitting
- Cross-validation
- Nonparametric methods

Outline

Training and test

Training set

$$(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)$$

Test set

$$(x_1, y_1^*), \ldots, (x_i, y_i^*), \ldots, (x_n, y_n^*)$$

where $x_1, \ldots, x_n$ are treated as fixed. In matrix notation:

$$\underset{n\times 1}{\mathbf{y}} = \begin{bmatrix} y_1 \\ \cdots \\ y_i \\ \cdots \\ y_n \end{bmatrix}, \quad \underset{n\times p}{\mathbf{X}} = \begin{bmatrix} x_1 \\ \cdots \\ x_i \\ \cdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \\ x_{n1} & \cdots & x_{nj} & \cdots & x_{np} \end{bmatrix}$$

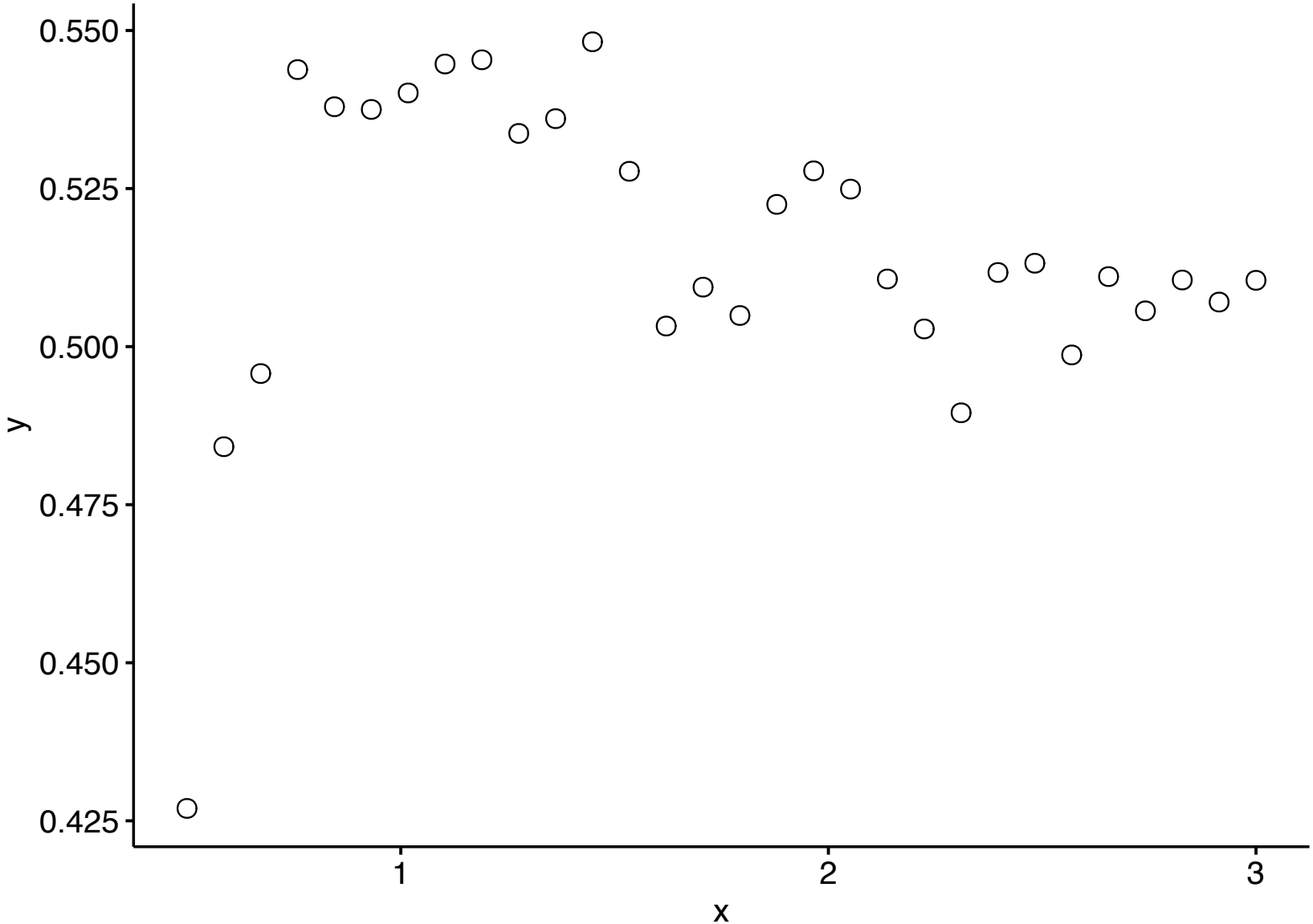
Signal and noise

Here $y_1, \ldots, y_n$ are realizations of the random variables

$$Y_i = f(x_i) + \varepsilon_i, \quad i = 1, \ldots, n$$

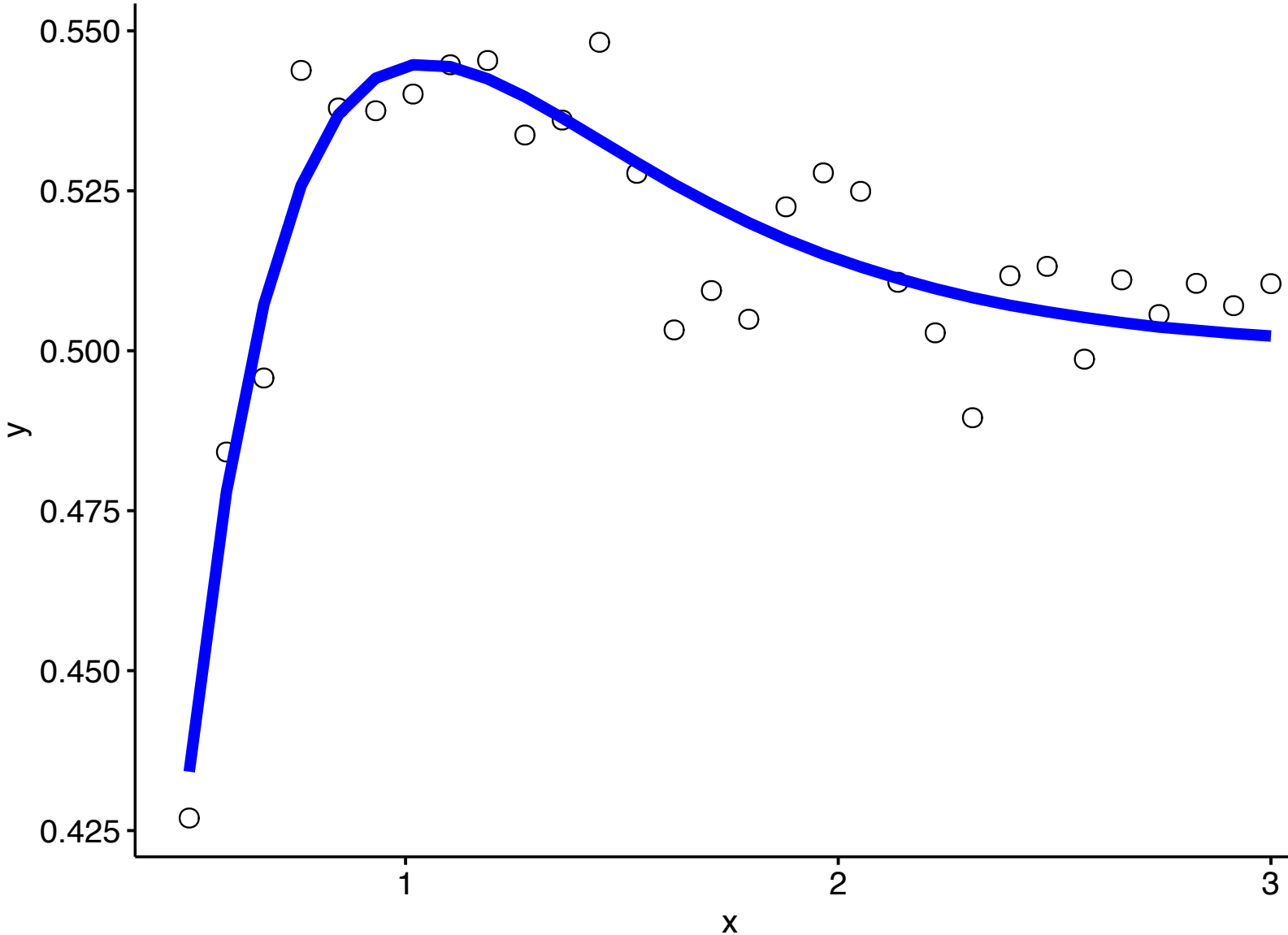
- f is the **regression function** (signal)
- ε is the **error** (noise)
  $\varepsilon_1, \ldots, \varepsilon_n$ i.i.d. with $\mathbb{E}(\varepsilon) = 0$ and $\mathbb{V}\mathrm{ar}(\varepsilon) = \sigma^2$

Training set



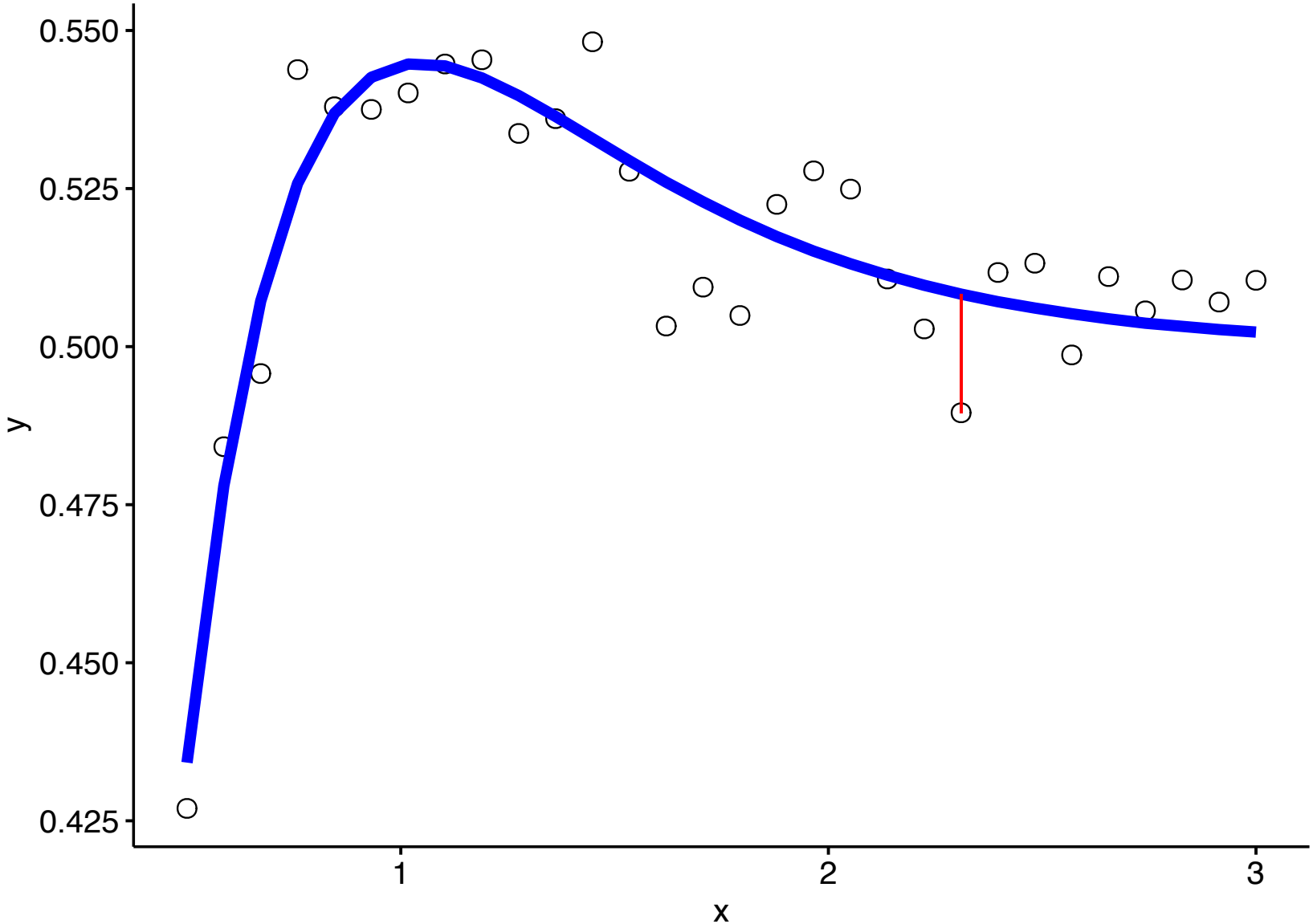
From Azzalini & Scarpa (2012)

Regression function



From Azzalini & Scarpa (2012)

Error



From Azzalini & Scarpa (2012)

Mean squared error

- Loss function = **mean squared error** (MSE)
- Suppose we have estimated f by $\hat{f}$ by using the training set
- We can compute the MSE on the training set

$$\mathrm{MSE}_{\mathrm{Tr}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

- But the goal is to have a low MSE on the test set

$$\mathrm{MSE}_{\mathrm{Te}} = \frac{1}{n}\sum_{i=1}^{n}(y_i^* - \hat{f}(x_i))^2$$

Polynomial regression

- Polynomial regression model of degree d:

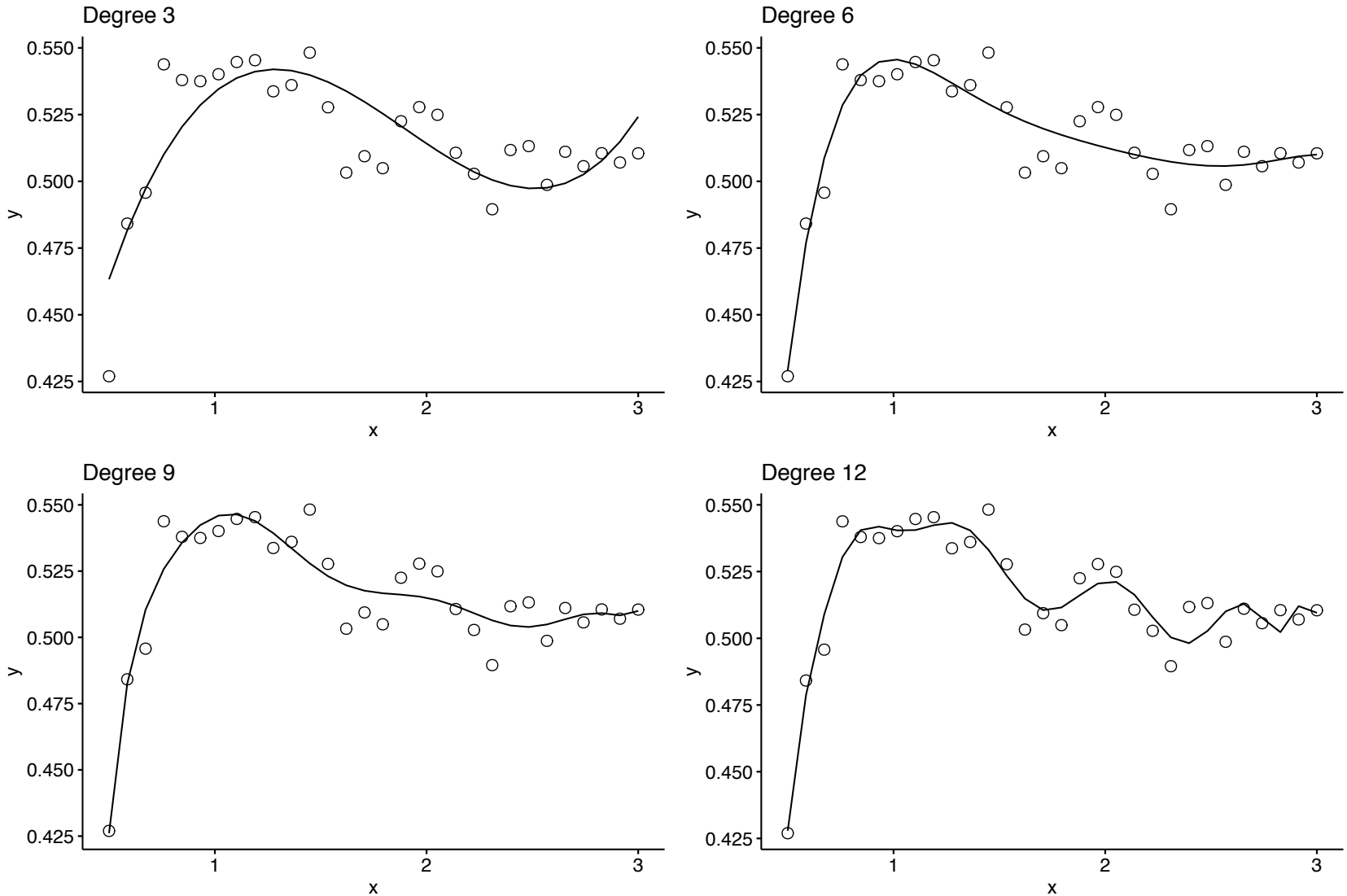
$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_d x^d$$

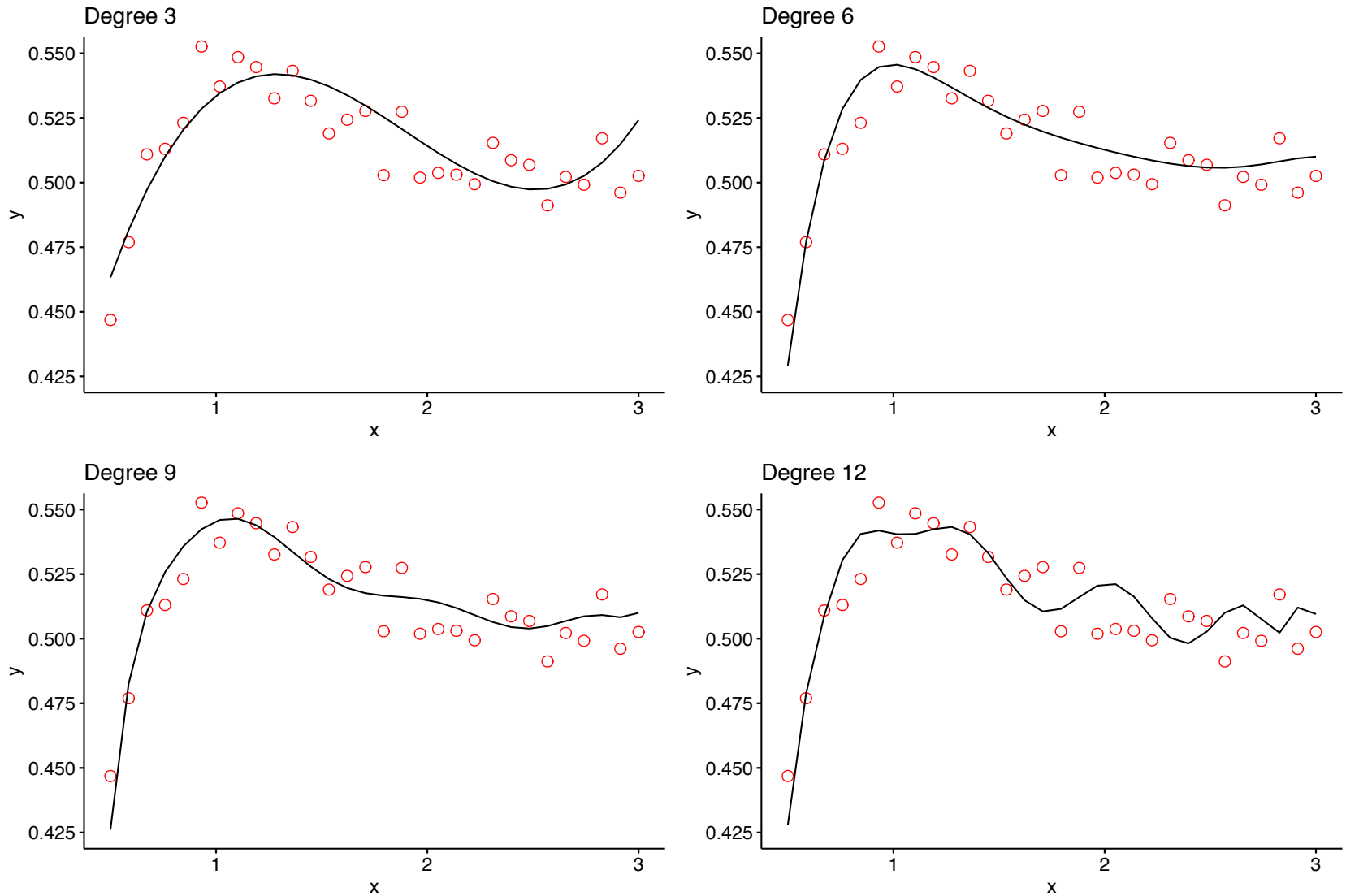
- Use the training set to estimate f:

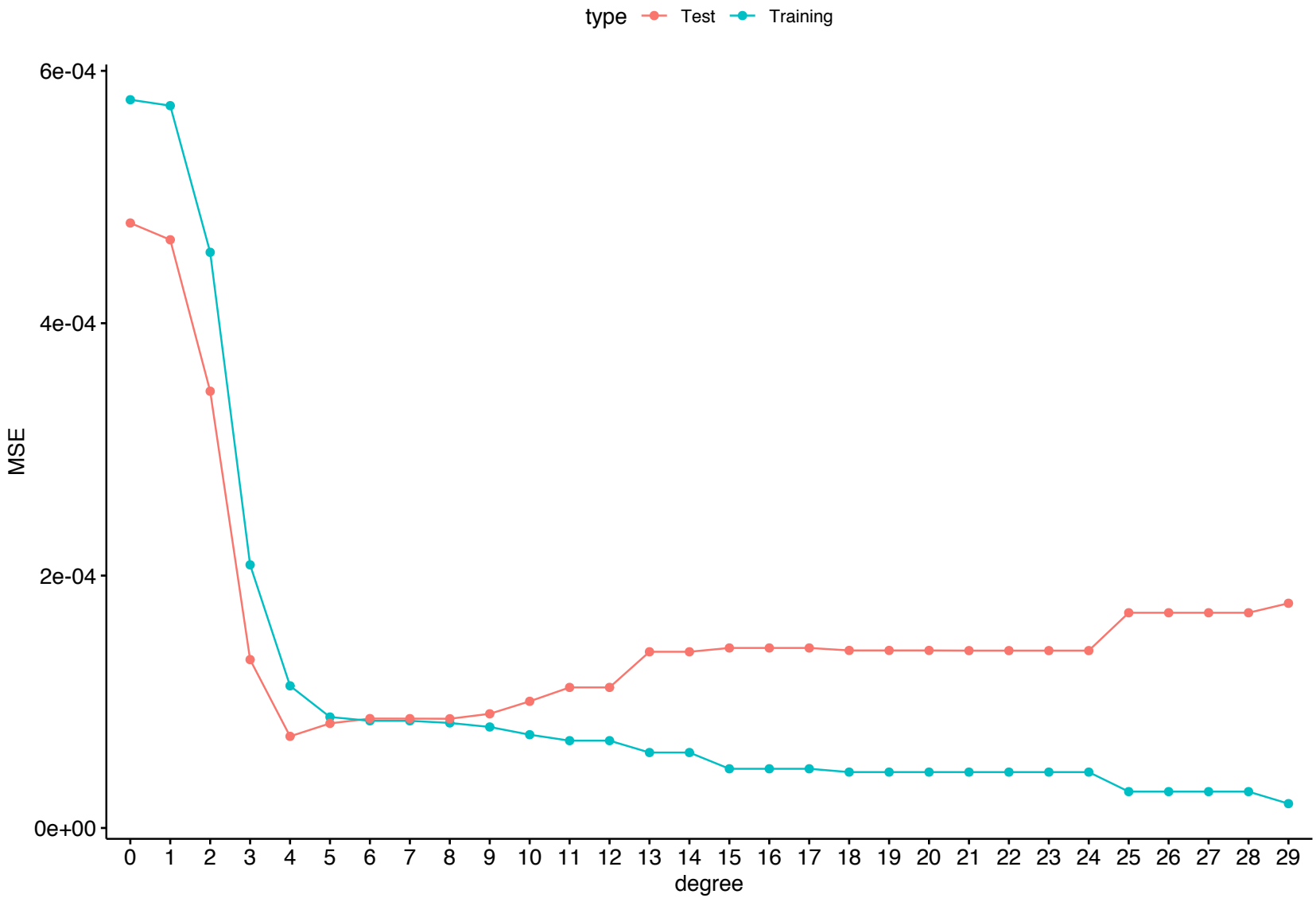
$$\hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2 + \ldots + \hat{\beta}_d x^d$$

- The degree d is a **tuning parameter**: small and large d corresponds to low and high flexibility, respectively
- Goal: find the degree d that minimizes the prediction error

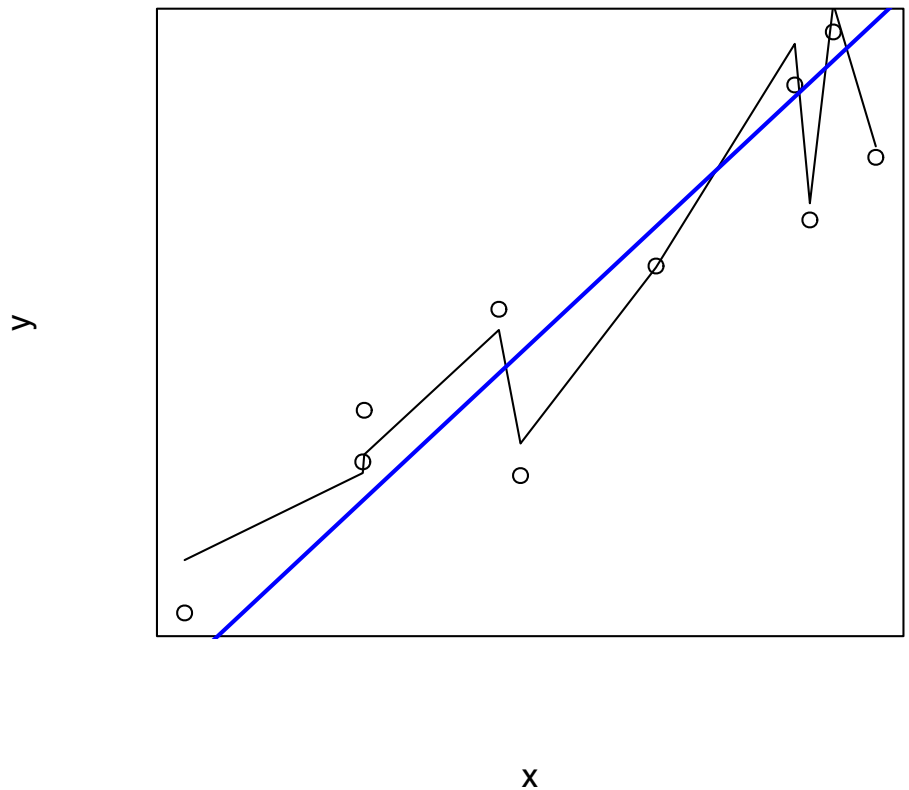
$$\mathrm{Err} = \mathbb{E}(\mathrm{MSE}_{\mathrm{Te}}) = \mathbb{E}\left\{\frac{1}{n}\sum_{i=1}^{n}[Y_i^* - \hat{f}(x_i)]^2\right\}$$


Degree 3
Degree 6
Degree 9
Degree 12
0.550
0.525
0.500
0.475
0.450
0.425
y
1
2
3
x


Degree 3
Degree 6
Degree 9
Degree 12
0.550
0.525
0.500
0.475
0.450
0.425
y
1
2
3
x


type
Test
Training
6e-04
4e-04
2e-04
0e+00
MSE
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
degree

Overfitting



mistaking noise for signal

Prediction error

Sources of error:

- **Irreducible error**
  Can we ever predict Y from X with zero error? No. Even the true regression function f cannot do this
- **Estimation bias**
  What happens if our fitted function $\hat{f}$ belongs to a model class that is far from the true f? E.g. we choose to fit a regression line in a setting where the true relationship is far from linear?
- **Estimation variance**
  What happens if our fitted function $\hat{f}$ is itself quite variable? In other words, over different copies of the training data, we end up constructing substantially different $\hat{f}$?

Reducible and irreducible error

The prediction error

$$\mathrm{Err} = \mathbb{E}(\mathrm{MSE}_{\mathrm{Te}}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left\{[Y_i^* - \hat{f}(x_i)]^2\right\}$$

depends on two quantities, the **reducible error** and the **irreducible error**:

$$\begin{aligned}
\mathbb{E}\{[Y_i^* - \hat{f}(x_i)]^2\} &= \mathbb{E}\{[f(x_i) - \hat{f}(x_i)]^2\} + \mathbb{V}\mathrm{ar}(\varepsilon_i^*) \\
&= \underbrace{\mathbb{E}\{[f(x_i) - \hat{f}(x_i)]^2\}}_{\text{Reducible}} + \underbrace{\sigma^2}_{\text{Irreducible}}
\end{aligned}$$

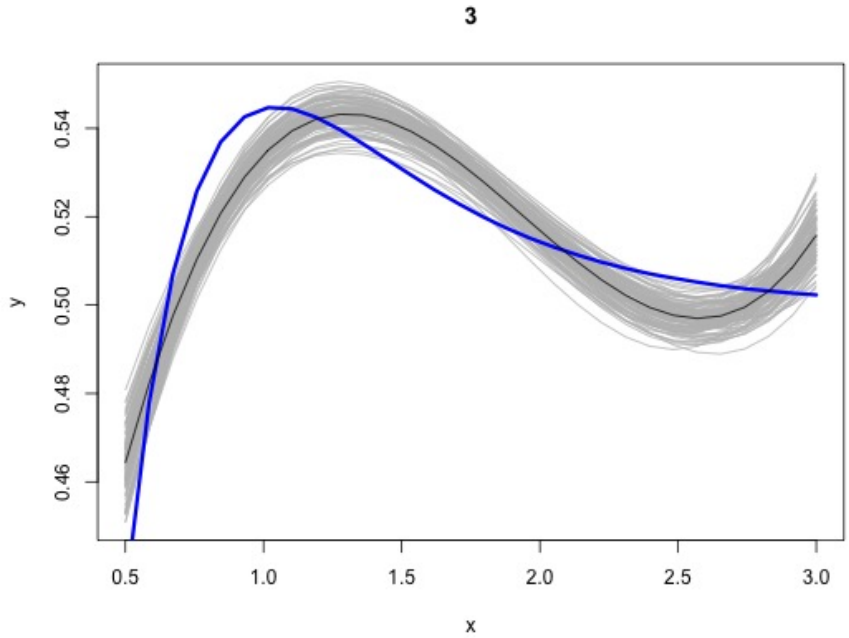
Bias-variance decomposition

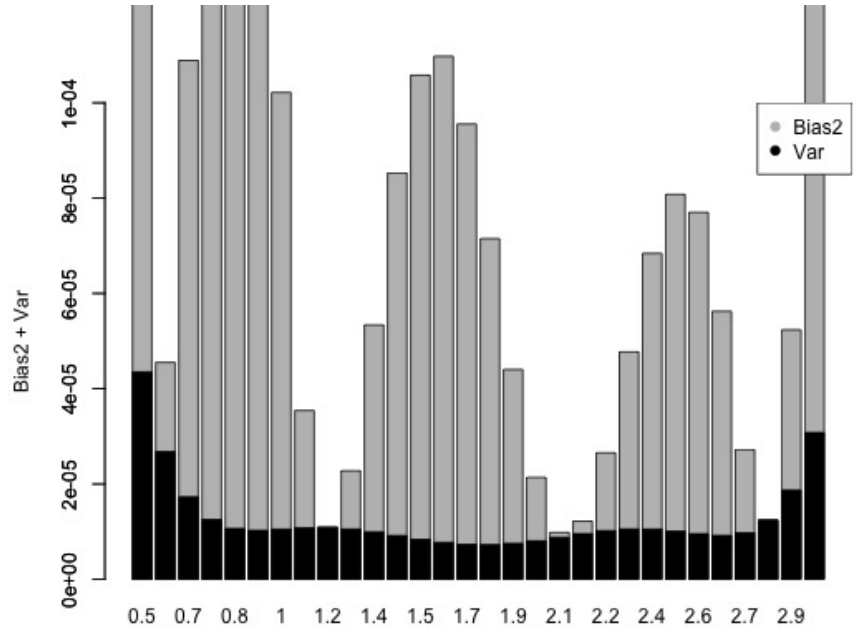
The reducible error can be decomposed into (squared) **bias** and **variance** of $\hat{f}$

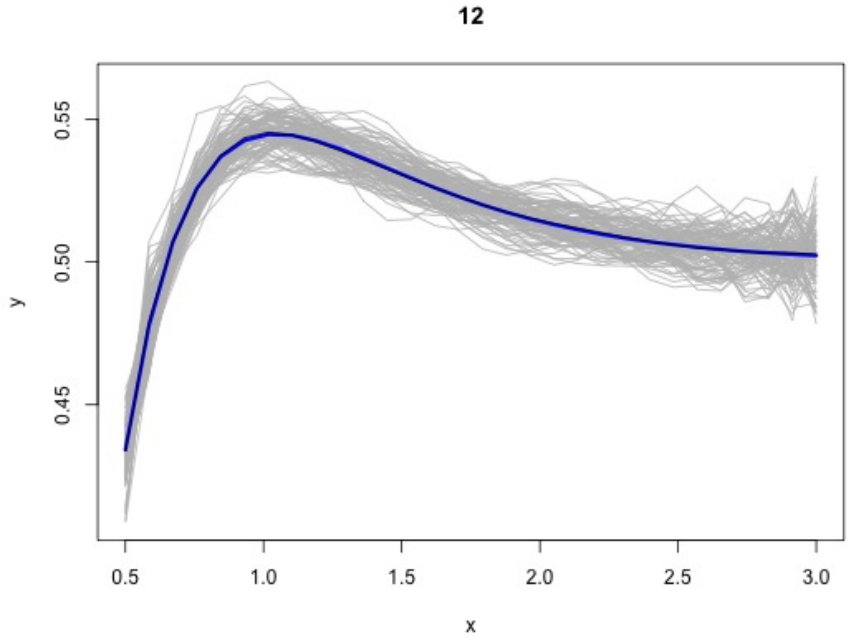
$$\mathbb{E}\{[f(x_i) - \hat{f}(x_i)]^2\} = \underbrace{[f(x_i) - \mathbb{E}\hat{f}(x_i)]^2}_{\text{Bias}^2} + \underbrace{\mathbb{V}\text{ar}[\hat{f}(x_i)]}_{\text{Variance}}$$

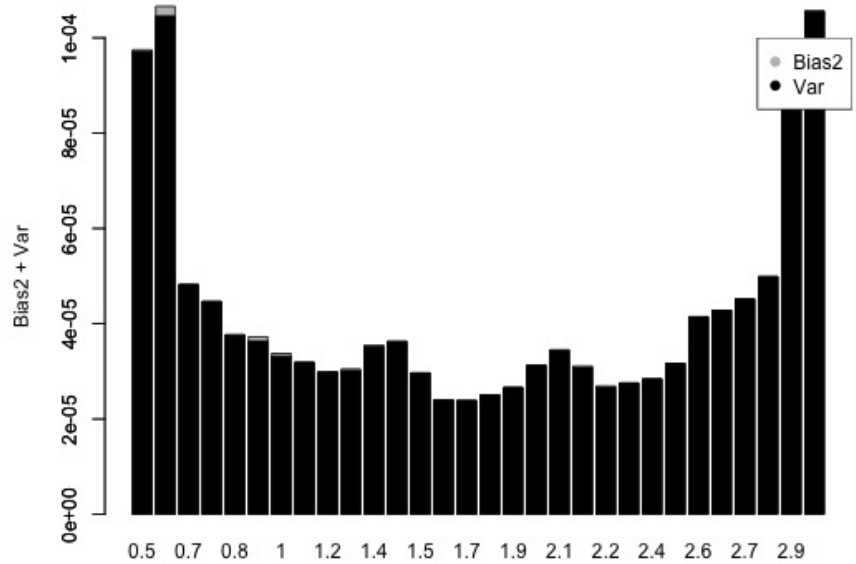
Summing up, we have

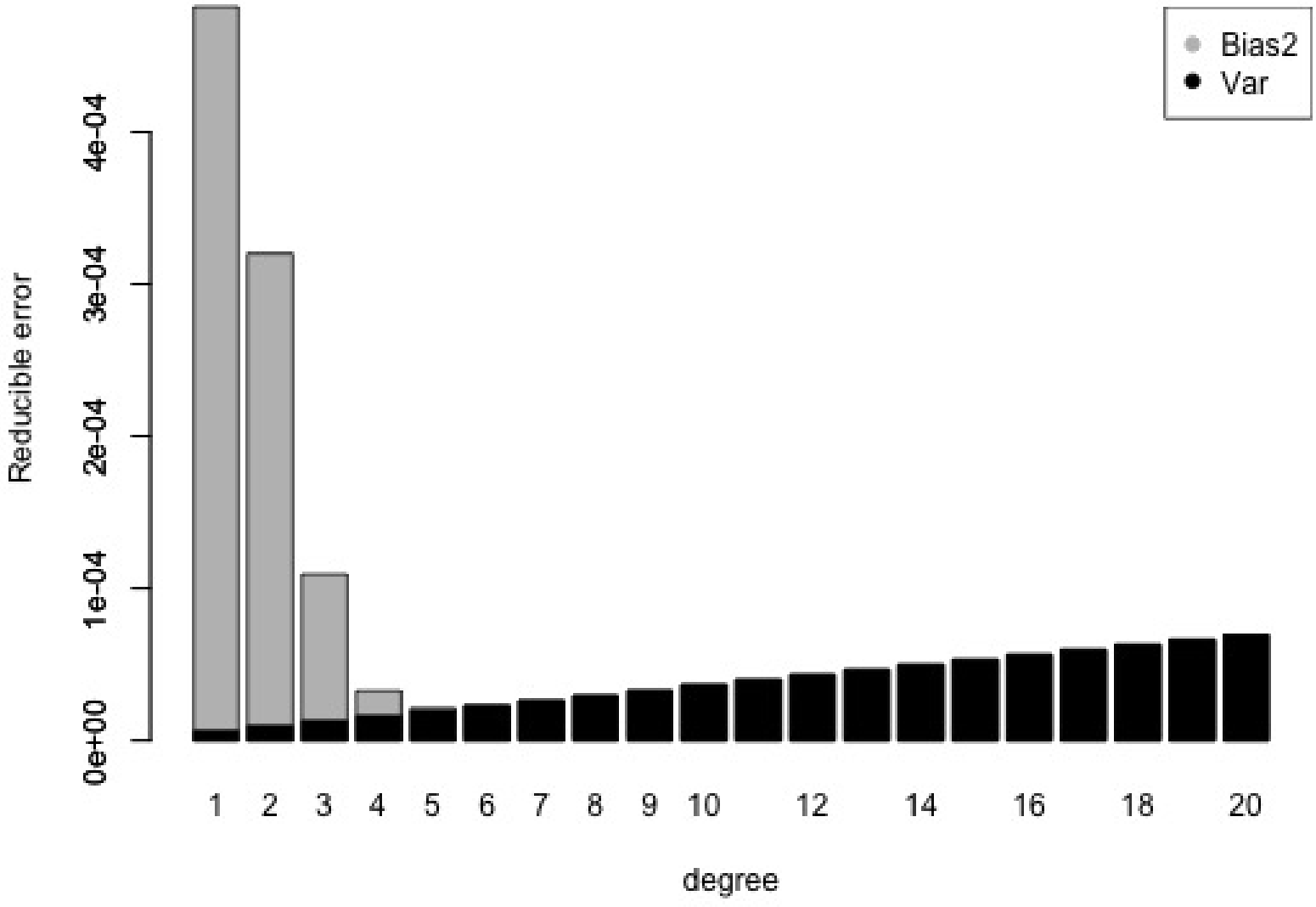
$$\text{Err} = \frac{1}{n}\sum_{i=1}^{n}[f(x_i) - \mathbb{E}\hat{f}(x_i)]^2 + \frac{p\sigma^2}{n} + \sigma^2$$


3
y
x
0.46
0.48
0.50
0.52
0.54
0.5
1.0
1.5
2.0
2.5
3.0

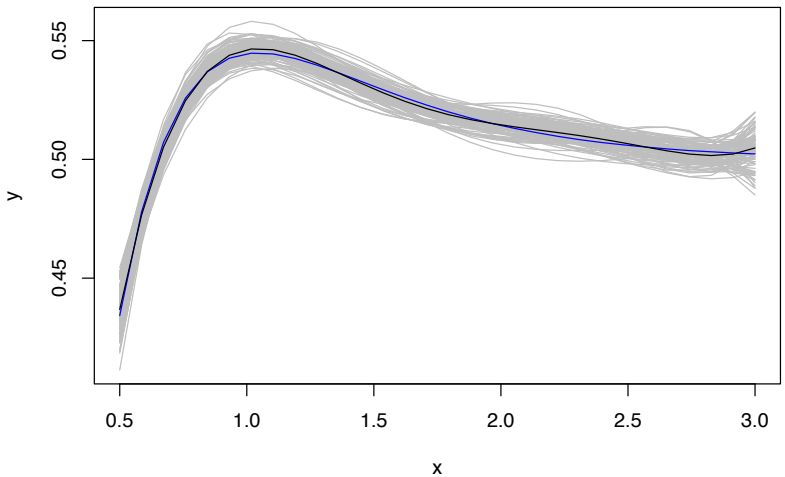

Bias2 + Var
0e+00
2e-05
4e-05
6e-05
8e-05
1e-04
0.5 0.7 0.8 1 1.2 1.4 1.5 1.7 1.9 2.1 2.2 2.4 2.6 2.7 2.9
Bias2
Var

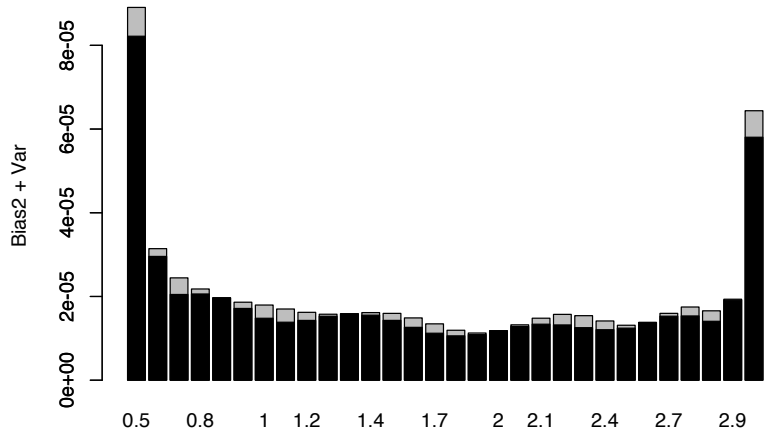

12
y
x
0.45
0.50
0.55
0.5
1.0
1.5
2.0
2.5
3.0


Bias2 + Var
0e+00
2e-05
4e-05
6e-05
8e-05
1e-04
0.5 0.7 0.8 1 1.2 1.4 1.5 1.7 1.9 2.1 2.2 2.4 2.6 2.7 2.9
Bias2
Var


Bias2
Var
Reducible error
0e+00
1e-04
2e-04
3e-04
4e-04
1
2
3
4
5
6
7
8
9
10
12
14
16
18
20
degree

Best polynomial





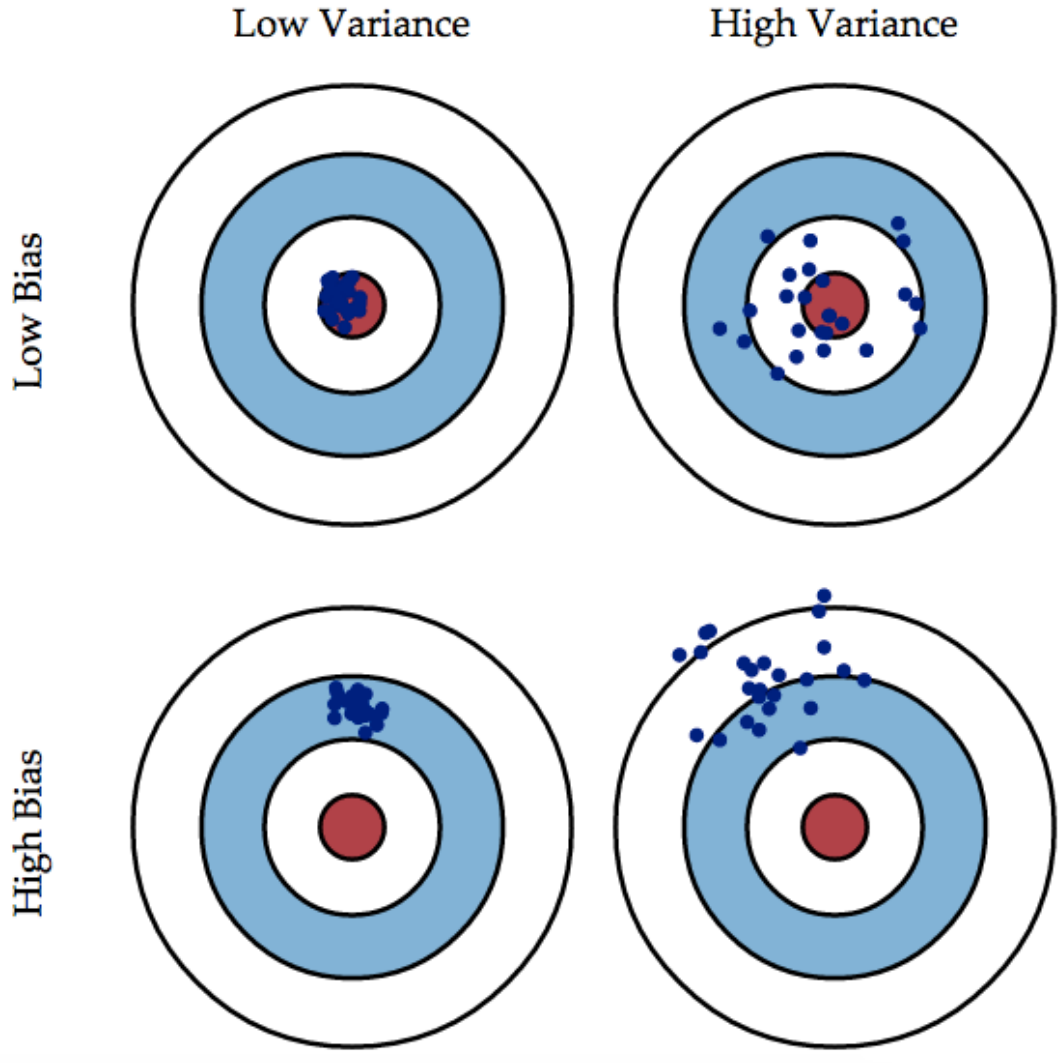
Degree $d = 5$

Bias-variance trade-off

Bias and variance are conflicting entities, and we cannot minimise both simultaneously:

- Models $\hat{f}$ with low bias tend to have high variance
- Models $\hat{f}$ with low variance tend to have high bias
- Even if our prediction is unbiased, i.e. $\mathbb{E}\hat{f}(x_i) = f(x_i)$, we can still incur a large error if it is highly variable
- On the other hand, $\hat{f}(x_i) = 0$ has 0 variance but will be terribly biased

To predict well, we must therefore choose a **trade-off** between bias and variance


Low Variance
High Variance
Low Bias
High Bias

SUMMARY: bias-variance tradeoff

- **Data**: training set / test set
- **Signal/noise**: regression function / error
- **Performance**: MSE on the same data / MSE on new data
- **Overfitting**: mistaking noise for signal
- **Prediction error**: reducible + irreducible
- **Reducible error**: bias^2 + variance
- **Tradeoff**: allow some bias if it decreases more variance

Outline

Optimism

Optimism is the expected difference of the test error and training error

$$\mathrm{Opt} = \mathbb{E}(\mathrm{MSE}_{\mathrm{Te}}) - \mathbb{E}(\mathrm{MSE}_{\mathrm{Tr}})$$

Optimism is important because an estimate of optimism leads to an estimate of prediction error

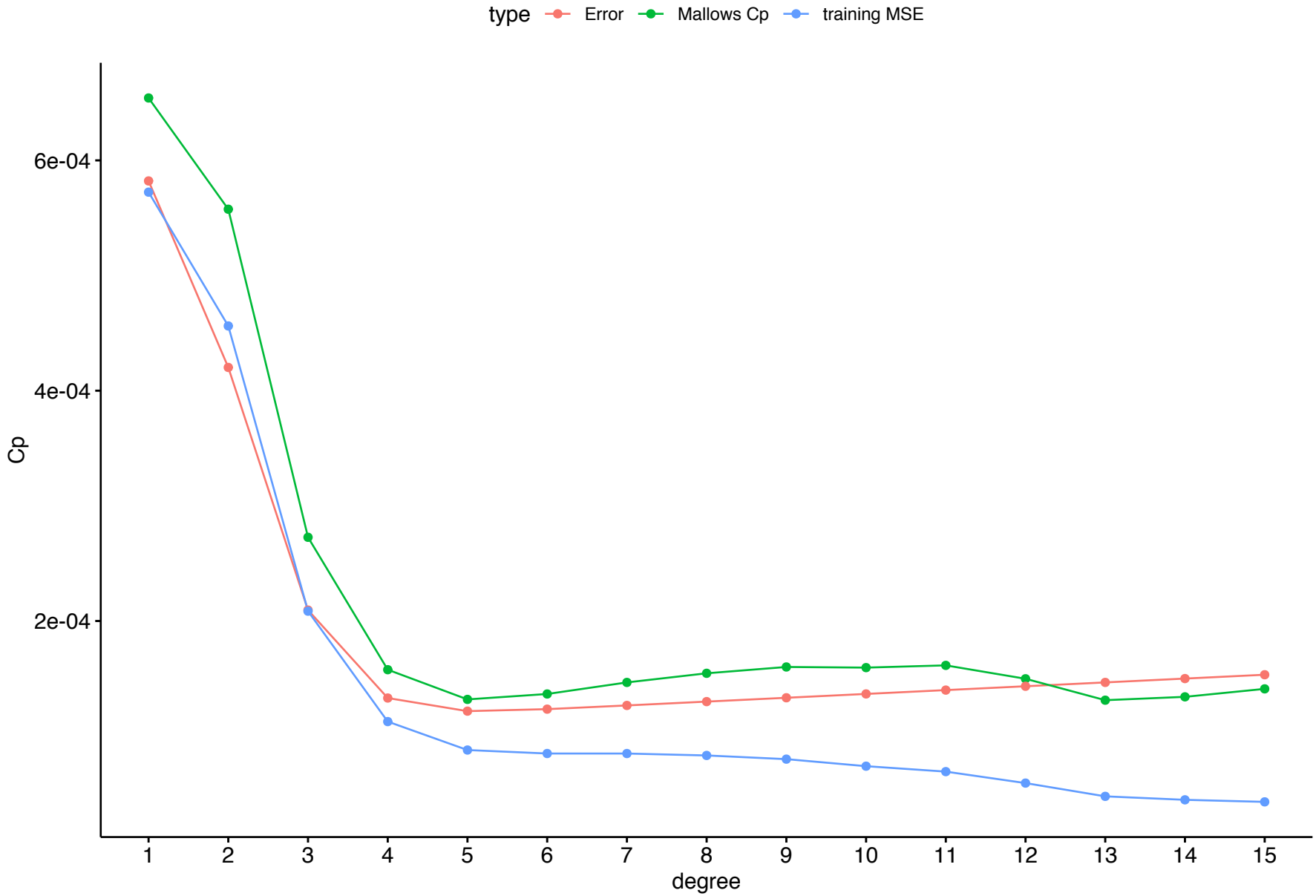
$$\mathbb{E}(\widehat{\mathrm{MSE}_{\mathrm{Te}}}) = \mathrm{MSE}_{\mathrm{Tr}} + \widehat{\mathrm{OptF}}$$

For the linear model

$$\mathrm{Opt} = \frac{2\sigma^2 p}{n}$$

it gives **Mallows' Cp**

$$\mathrm{Cp} = \mathrm{MSE}_{\mathrm{Tr}} + \frac{2\hat{\sigma}^2 p}{n}$$


type
Error
Mallows Cp
training MSE
6e-04
4e-04
2e-04
Cp
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
degree

AIC and BIC

- **AIC** is given by

$$\mathrm{AIC} = -2\ell(\hat{\beta}, \hat{\sigma}^2) + 2p$$

  where ℓ is the loglikelihood. For the linear model

$$-2\ell(\hat{\beta}, \hat{\sigma}^2) = n \log(\mathrm{MSE}_{\mathrm{Tr}})$$

- For linear models, Cp and AIC are proportional to each other, and the lowest Cp corresponds to the lowest AIC
- **BIC** is given by

$$\mathrm{BIC} = -2\ell(\hat{\beta}, \hat{\sigma}^2) + \log(n)p$$

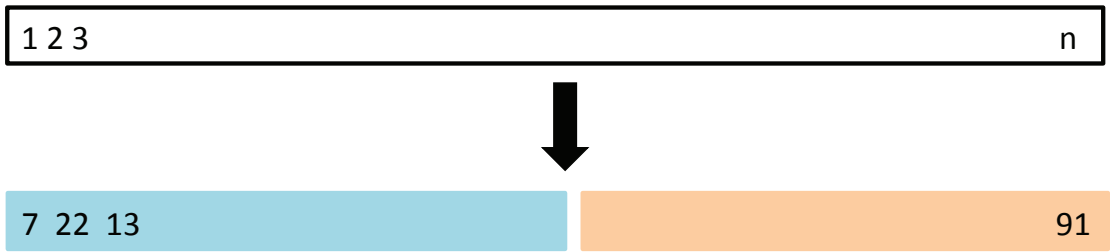
- Since $\log(n) > 2$ for any $n > 7$, the BIC statistic generally results in the selection of smaller models than AIC

Cross-validation

- Training an algorithm and evaluating its statistical performance on the same data yields an overoptimistic result
- **Cross-validation** (CV) is based on the idea of **splitting the data**: part of data (the training set) is used for training the algorithm, and the remaining data (the validation set) are used for evaluating the performance of the algorithm
- The major interest of CV lies in the universality of the data splitting heuristics. Therefore, CV is a **non-parametric method** which can be applied to any algorithm in any framework. This universality is not shared by e.g. Cp, which is specific to linear regression

Validation set approach

- A simple approach is to randomly divide the n observations into two parts: a training set and a *validation* or *hold-out* set



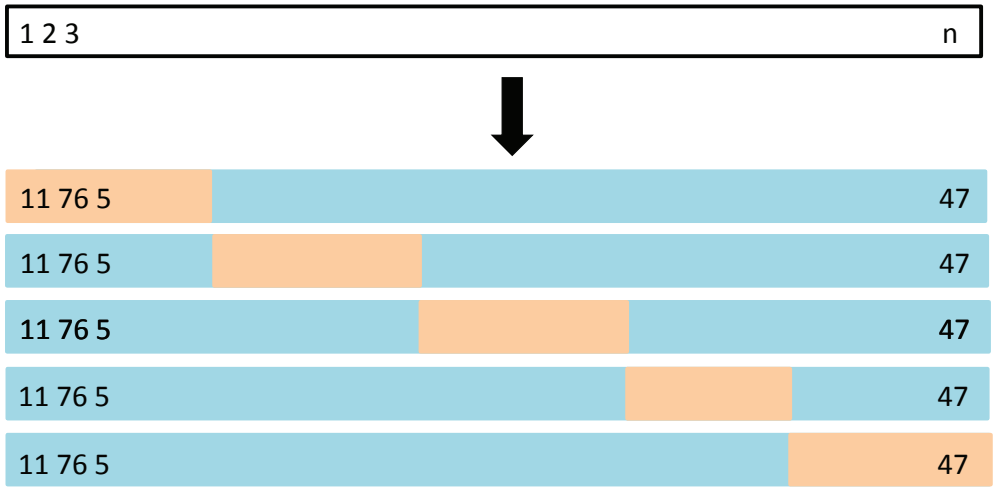
- The model is fitted on the training set $T \subset \{1, \ldots, n\}$, and the fitted model $\hat{f}^{-V}$ is used to predict the responses for the observations in the validation set $V = \{1, \ldots, n\} \setminus T$
- This results in the estimate of the expected test error

$$\widehat{\mathrm{Err}} = \frac{1}{\#V} \sum_{i \in V} [y_i - \hat{f}^{-V}(x_i)]^2$$

- This scheme reduces the sample size used for fitting the model, but this is not a problem when n is very large. If n is not very large, however, the validation estimate of the test error can be highly variable

K-fold CV

- Split the data into equal parts $V_1, \ldots, V_K$:



- Use observations $i \notin V_k$ for training the model and $i \in V_k$ for evaluating it:

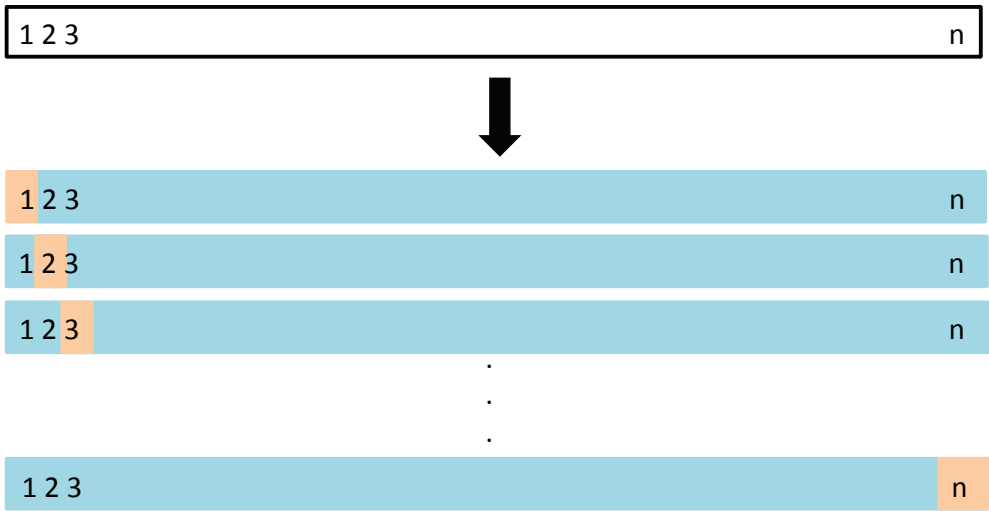
$$\frac{1}{\#V_k} \sum_{i \in V_k} [y_i - \hat{f}^{-V_k}(x_i)]^2$$

- Take the average to estimate the expected test error:

$$\widehat{\mathrm{Err}} = \frac{1}{K} \sum_{k=1}^{K} \left[\frac{1}{\#V_k} \sum_{i \in V_k} (y_i - \hat{f}^{-V_k}(x_i))^2 \right]$$

Leave-one-out CV

Each data point is left out and used for validation:



For $i = 1, \ldots, n$:

- Hold out the ith training observation (x_i, y_i)
- Use $n - 1$ observations for training the model $\hat{f}^{-i}$ and the hold-out observation (x_i, y_i) to evaluate it by $(y_i - \hat{f}^{-i}(x_i))^2$

Take the average to estimate the expected test error:

$$\widehat{\mathrm{Err}} = \frac{1}{n} \sum_{i=1}^{n} [y_i - \hat{f}^{-i}(x_i)]^2$$

Generalized CV

For the linear model, there is a shortcut for computing LOOCV:

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}^{-i}(x_i)\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}(x_i)}{1-h_i}\right)^2$$

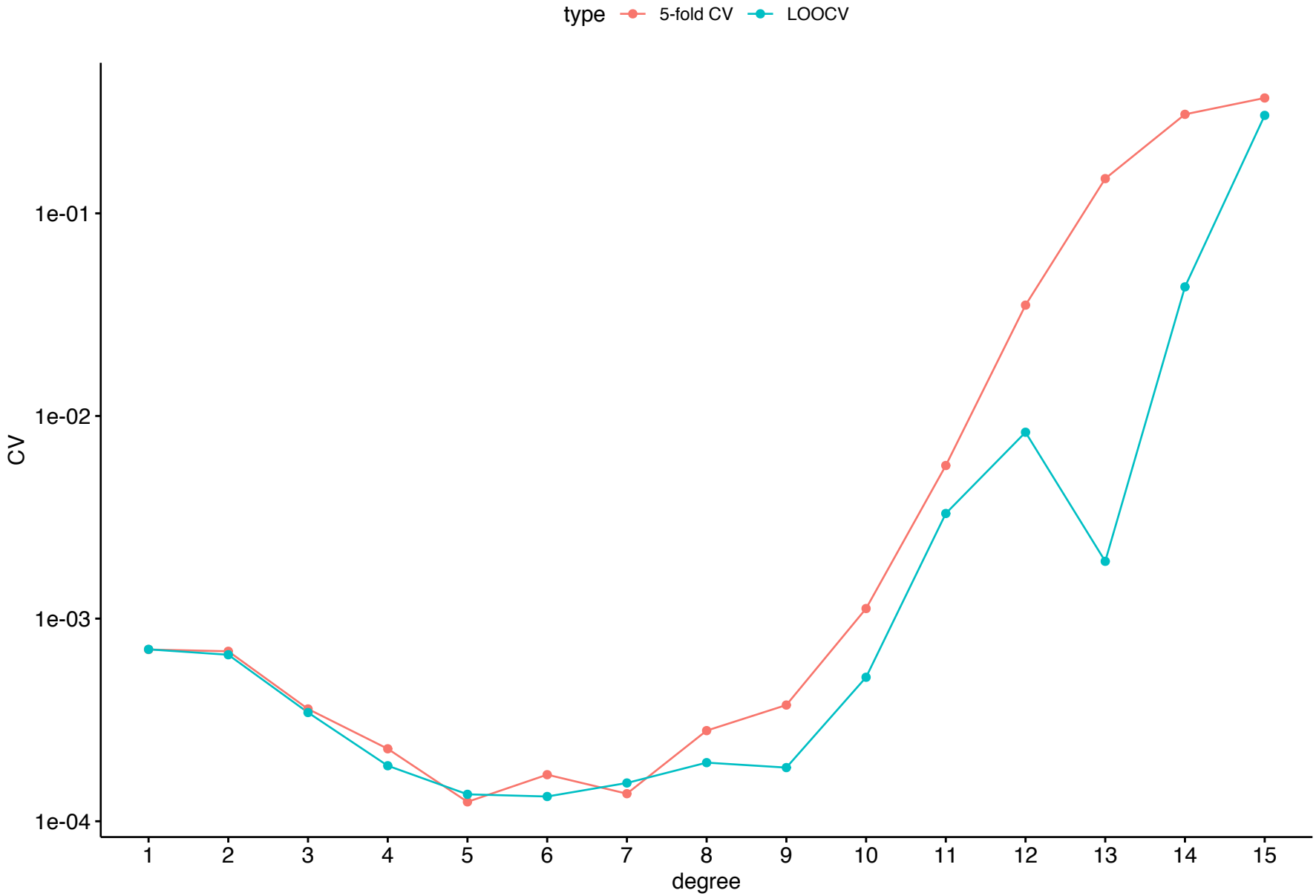
where h_i is the ith diagonal element of the projection matrix

$$\underset{n\times n}{\mathbf{H}} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$$

In generalized cross-validation we compute

$$\widehat{\mathrm{Err}} = \frac{\mathrm{MSE}_{\mathrm{Tr}}}{\left(1-\frac{p}{n}\right)^2}$$

where we approximate each h_i by their average $\frac{1}{n}\sum_{i=1}^{n} h_i = \frac{p}{n}$


type
5-fold CV
LOOCV
CV
1e-01
1e-02
1e-03
1e-04
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
degree

CV bias-variance tradeoff

- A common choice for K other than $K = n$ is to choose $K = 5$ or $K = 10$
- **Bias**: K-fold CV with $K = 5$ or 10 gives a biased (upward) estimate of $\mathbb{E}(\mathrm{MSE}_{\mathrm{Te}})$ because it uses less information (4/5 or 9/10 of the observations). LOOCV has very low bias (it uses $n - 1$ observations)
- **Variance**: Usually, LOOCV has high variance because it is an average of n extremely correlated quantities (because the fits $\hat{f}^{-i}$ and $\hat{f}^{-j}$ are based on $n - 2$ common observations), and K-fold CV with $K = 5$ or 10 has less variance because it is an average of quantities that are less correlated. Remember that the variance of the sum of highly correlated quantities is larger than that with midly correlated quantities:

$$\mathbb{V}\mathrm{ar}(A + B) = \mathbb{V}\mathrm{ar}(A) + \mathbb{V}\mathrm{ar}(B) + 2\mathbb{C}\mathrm{ov}(A, B)$$

- However, drawing a general conclusion on CV is nearly an impossible task because of the variety of frameworks

SUMMARY: optimism and cross-validation

- **Optimism**: test MSE - training MSE
- **Cp, AIC and BIC**: training MSE + penality for complexity
- **Cross-validation**: model-free approach based on data split
- **Types**: validation set, K-fold, leave-one-out, generalized

Outline

Nonparametric methods

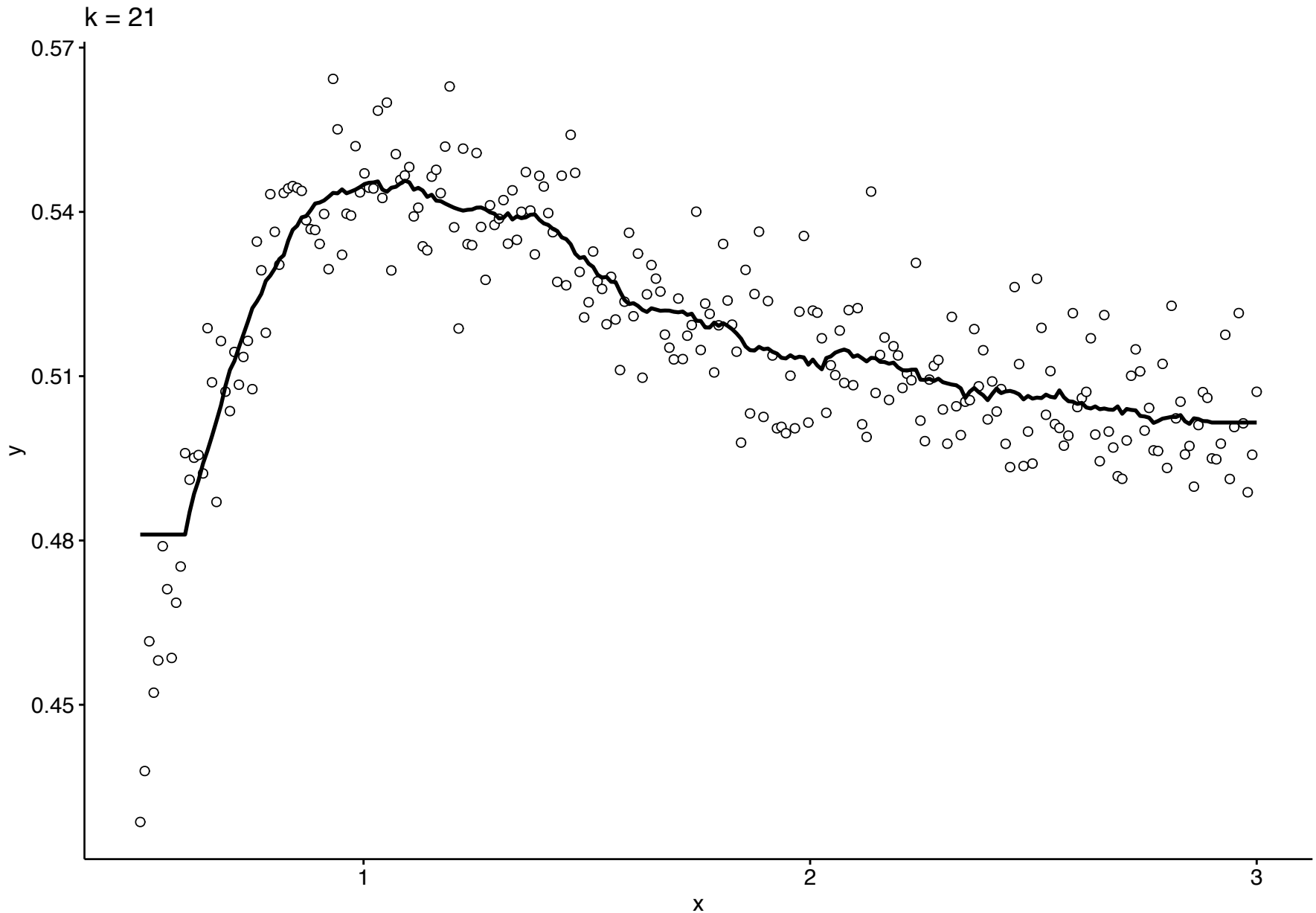
- Nonparametric methods do not make explicit assumptions about the functional form of f (e.g. polynomials)
- Leave data to **speak for themselves** in a free way
- **Advantage**: by avoiding the assumption of a particular functional form for f, they have a wider range of possible shapes for f
- **Disadvantage**: since they do not reduce the problem of estimating f to a small number of parameters, a very large number of observations is required

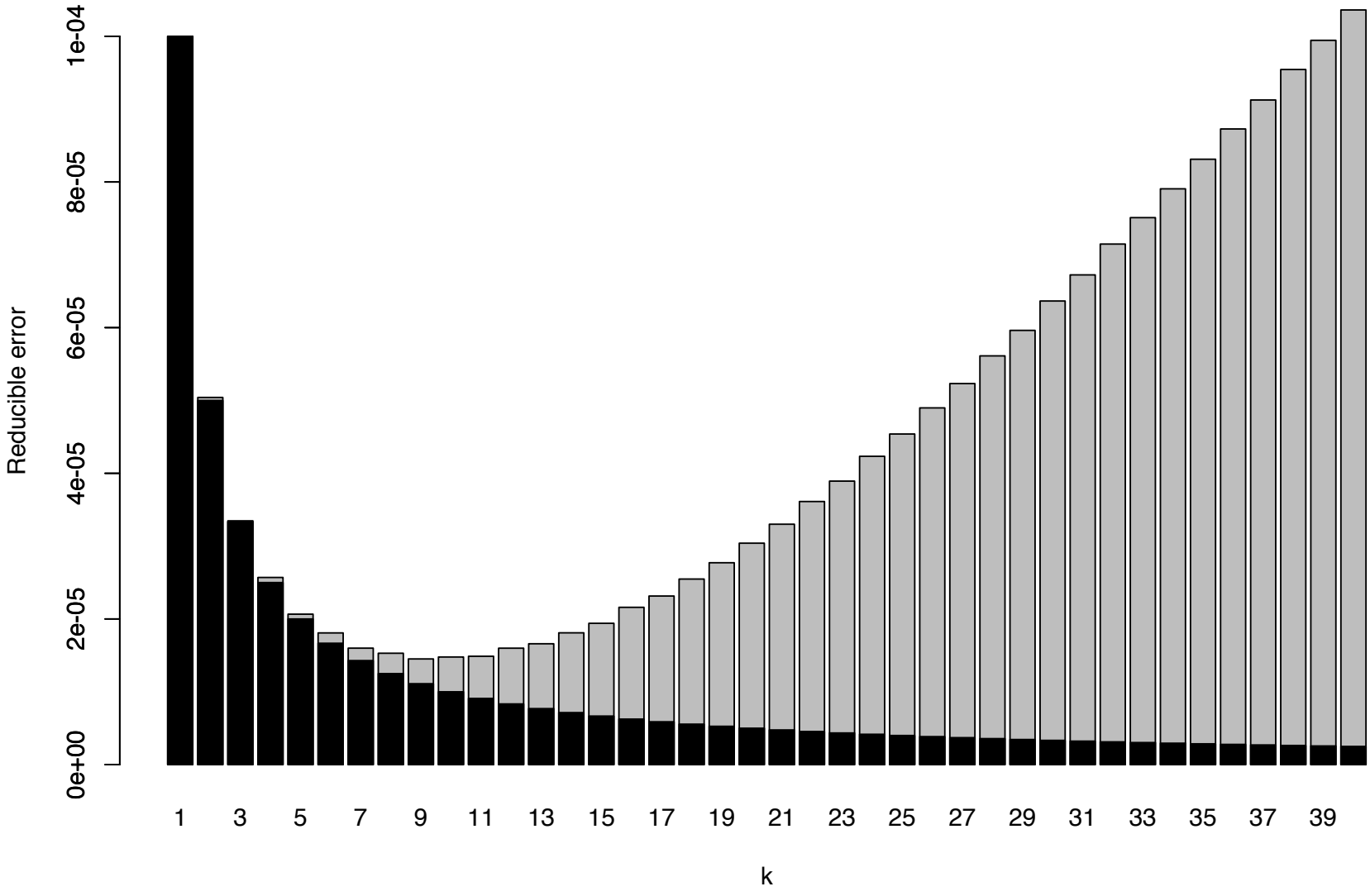
k-nearest neighbors

- A very simple and quite commonly used method k-nearest neighbors (kNN)
- Suppose we want to make a prediction at some x_1^*. Define the neighbourhood $N_k(x_1^*)$ to be the set of k training observations having values x_i closest to x_1^* in Euclidean norm $\|x_i - x_1^*\|$

$$\hat{f}(x_1^*) = \frac{1}{k} \sum_{i \in N_k(x_1^*)} y_i$$

- The number k is a **tuning parameter**: small and large k corresponds to a more and less flexible fit, respectively
- Since we are computing a distance, usually we center and scale the predictors


k = 21
0.57
0.54
0.51
0.48
0.45
y
1
2
3
x


Reducible error
0e+00
2e-05
4e-05
6e-05
8e-05
1e-04
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
k

kNN for classification

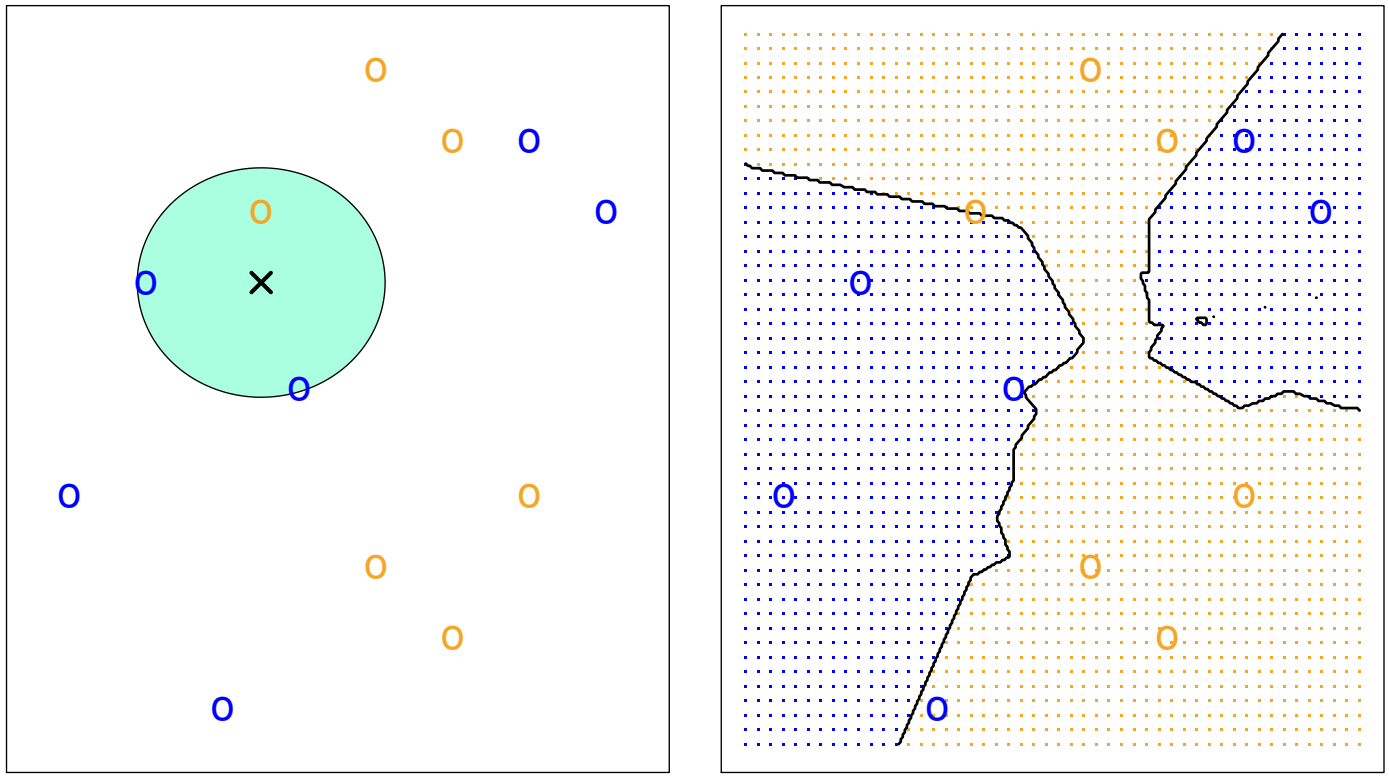
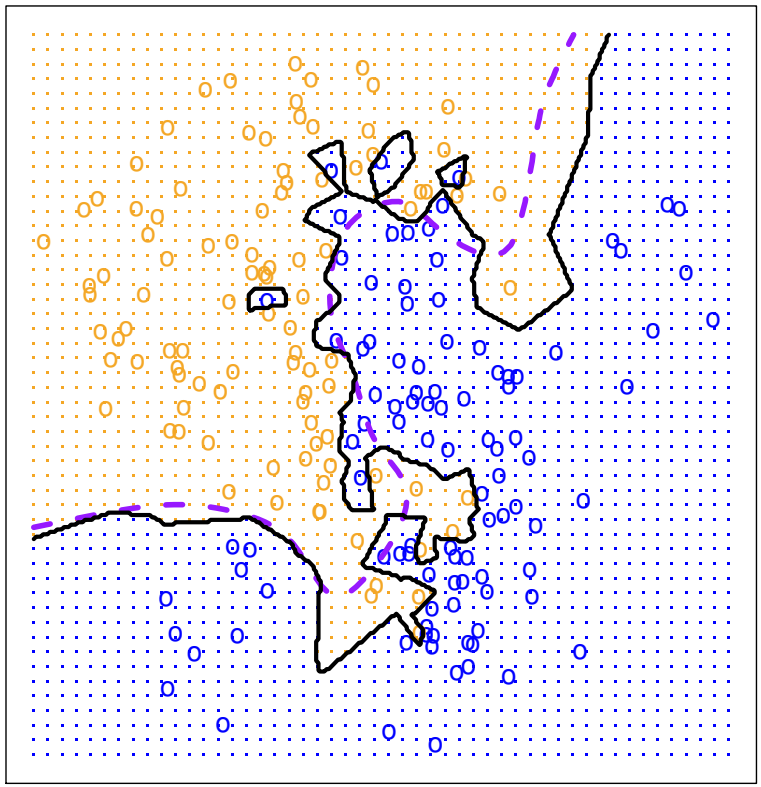


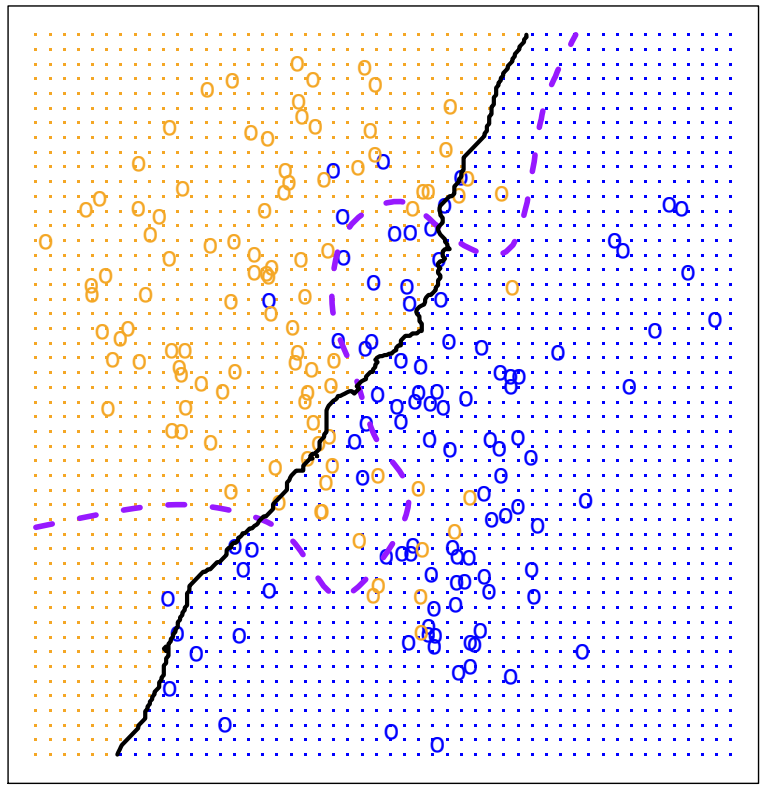
Figure from Gareth, Witten, Hastie & Tibshirani (2013)

KNN: K=1



Too flexible

KNN: K=100



Not flexible enough

Figure from Gareth, Witten, Hastie & Tibshirani (2013)

SUMMARY: nonparametric methods

- **Nonparametric**: very flexible but more observations needed
- **kNN**: look at the neighbors
- **Tuning parameter**: k defines flexibility
- **Neighbors distance**: center and scale the predictors
- **Curse of dimensionality**: closeness difficult in high dimensions