Two Machine Learning Challenges

Ljupčo Todorovski

University of Ljubljana, Faculty of Public Administration
Jožef Stefan Institute, Department of Knowledge Technologies

HSE, Jan-Feb 2021

The Task of Machine Learning

Recall the definition from Nada's lecture

Machine Learning: computer algorithms/machines that learn predictive models from labeled data

Formal definition of the task

- Given a data set S_{train} of examples of the form $(\boldsymbol{x}, y)$
- Find a model m that can estimate the value of y for a given $\boldsymbol{x}$

What Model to Select?

Accurate

Model has to provide estimates that accurately reconstruct the values of y

$$TrainError(m) = \frac{1}{|S_{train}|} \sum_{(\boldsymbol{x},y)\in S_{train}} (y - m(\boldsymbol{x}))^2$$

General

Model has to generalize well to examples outside the training set S_{train}

$$TestError(m) = \frac{1}{|S_{test}|} \sum_{(\boldsymbol{x},y)\in S_{test}} (y - m(\boldsymbol{x}))^2,\ S_{train} \cap S_{test} = \emptyset$$

Crucial Problem of Model Selection

Accurate model is not necessarily general, even more

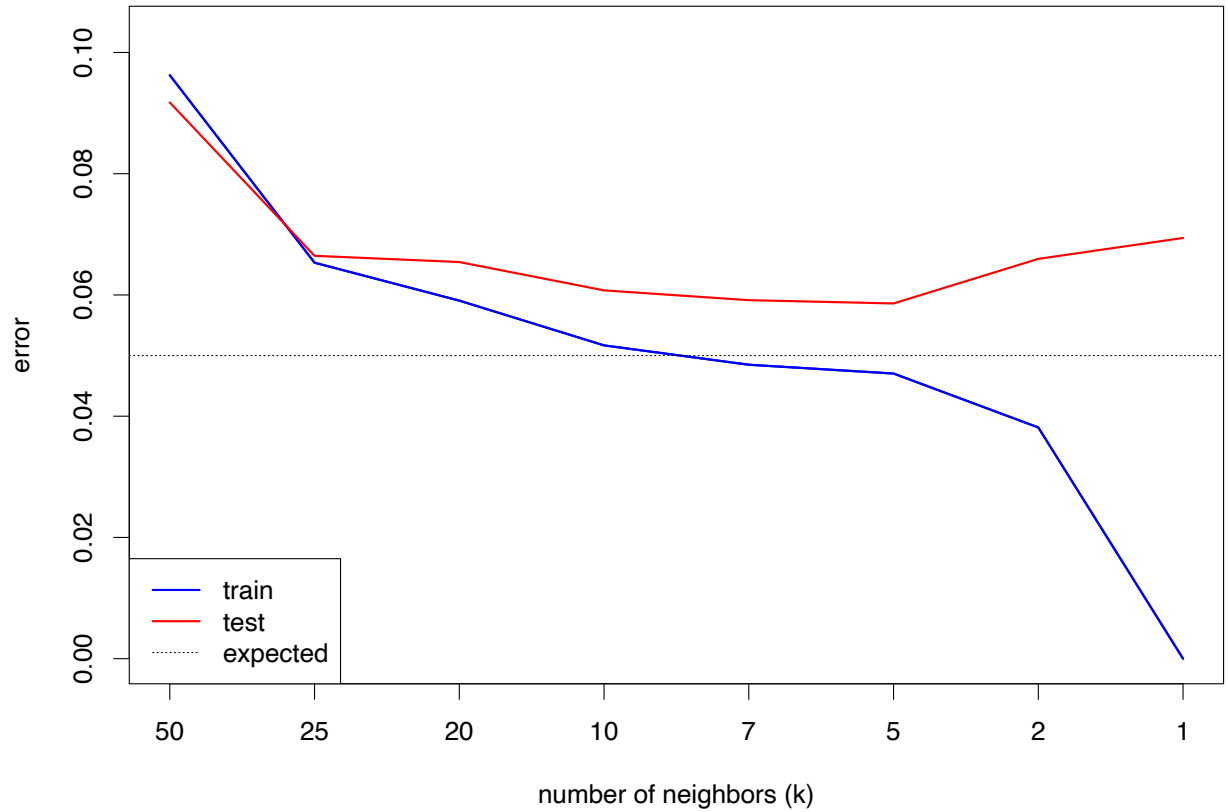
- If we choose the most accurate model m on training data
- Model m typically does not generalize well

Various reasons, including

- Noise in the data
- Limited training data sample

Accuracy vs Generality

With increasing the model complexity, we can accurately fit the training data (blue line), but fail to generalize to test data (red line).



Lecture Overview

Overfitting (Underfitting)

- Bias-variance decomposition
- Trade-off between bias and variance

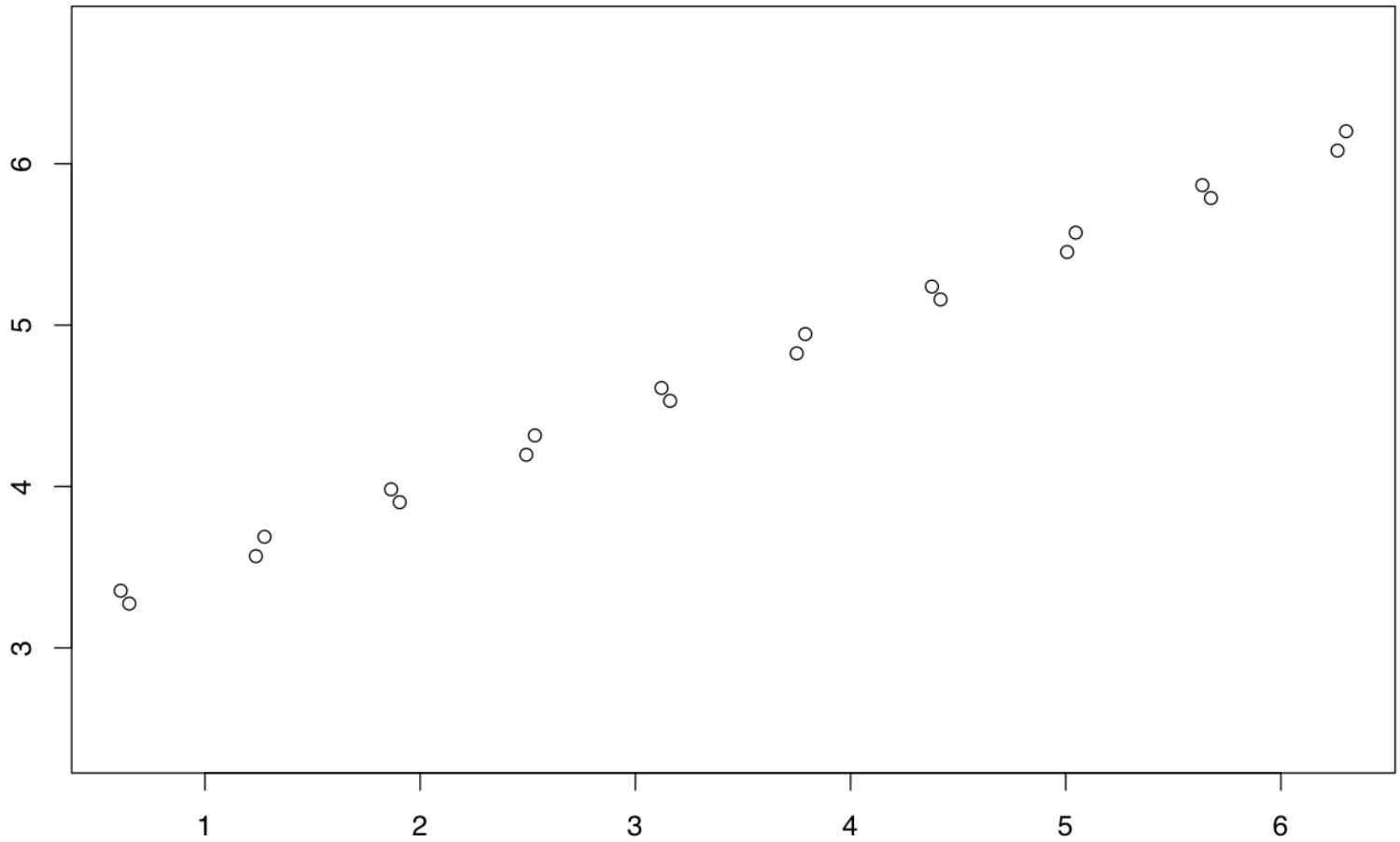
Curse of Dimensionality

- Optimal model and nearest neighbors
- Empirical illustration and mathematical results

Software platforms for practical exercises

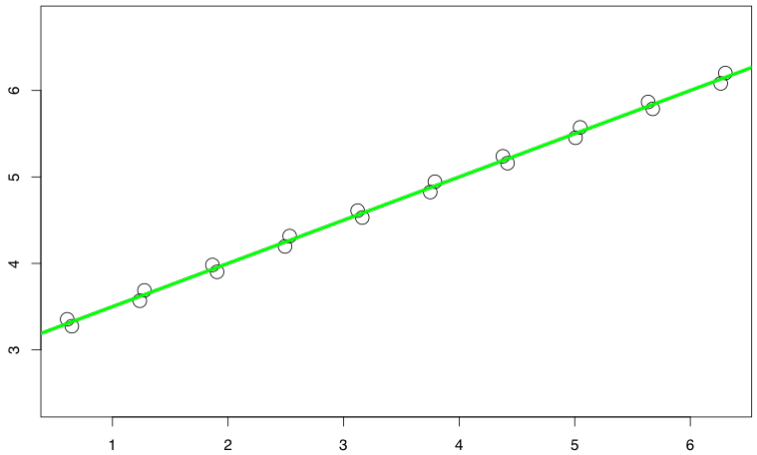
- R environment for statistical computing
- Python and Jypyter notebooks

Data Set: $p = 1, D_1, D_Y \subseteq \mathbb{R}$

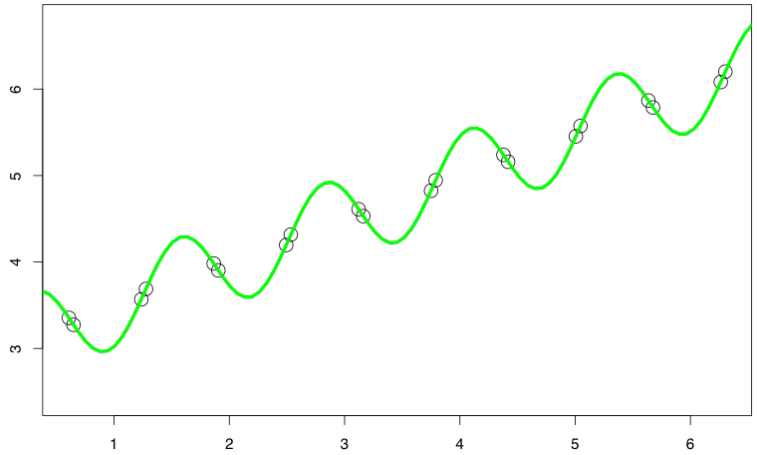


Model Selection: Which Model is Better?

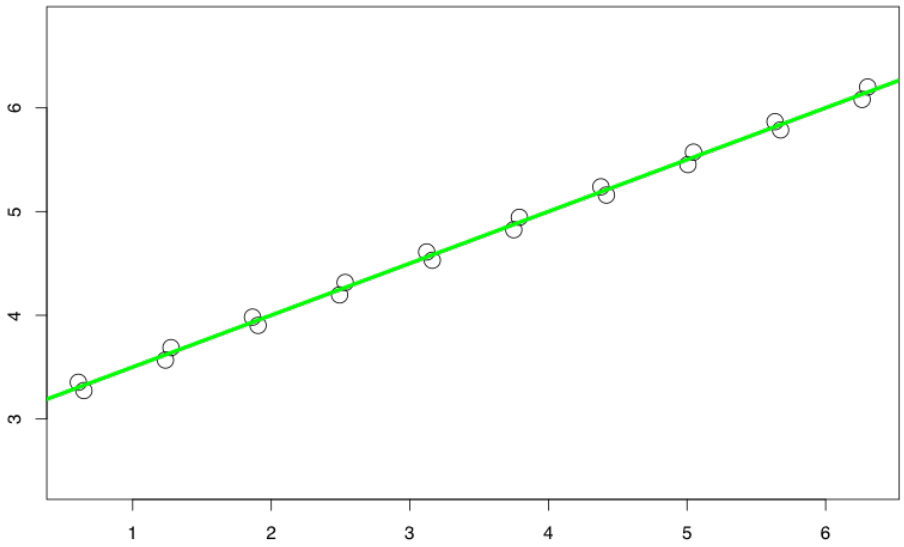
Linear

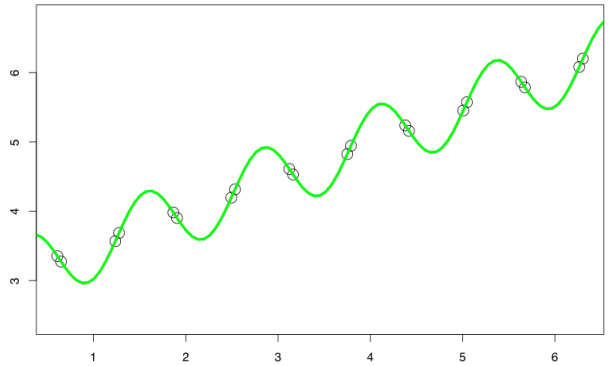


Oscillatory

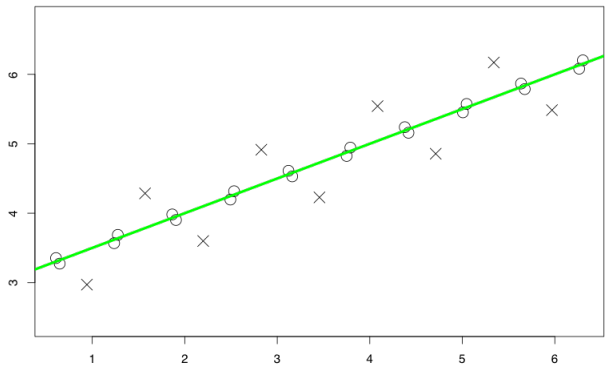


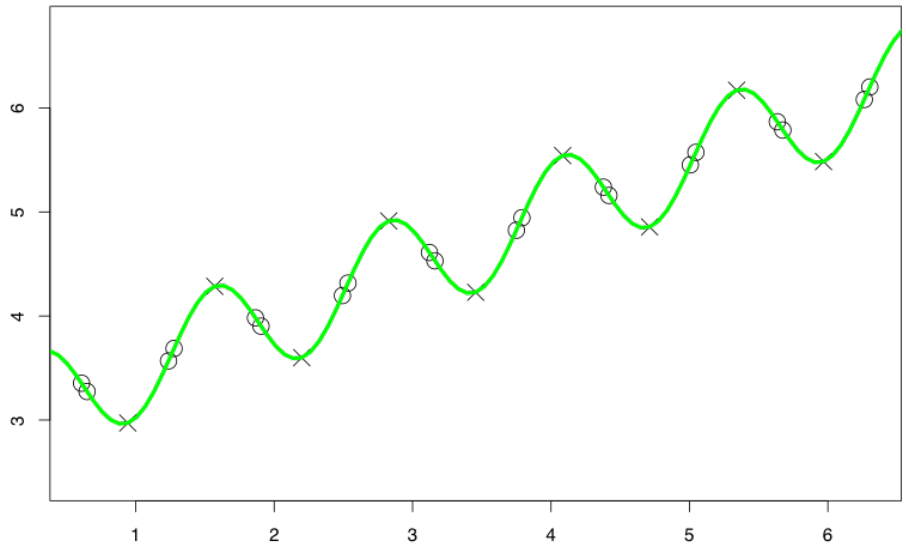
Model Selection: Linear (William of Ockham)





Model Selection: Oscillatory (Test Data)



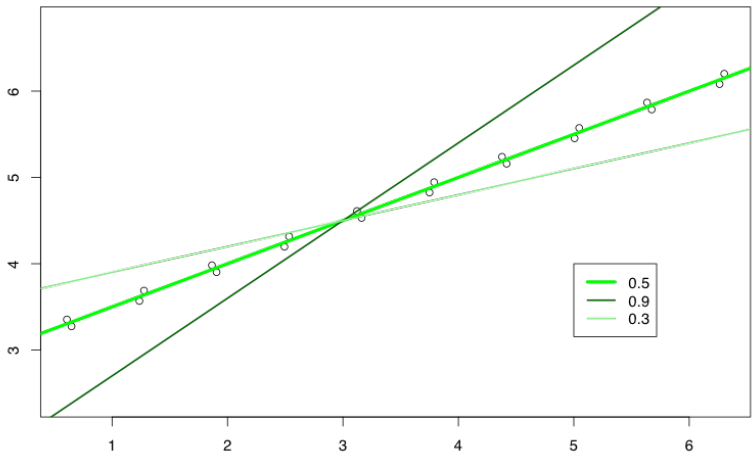


Model Spaces

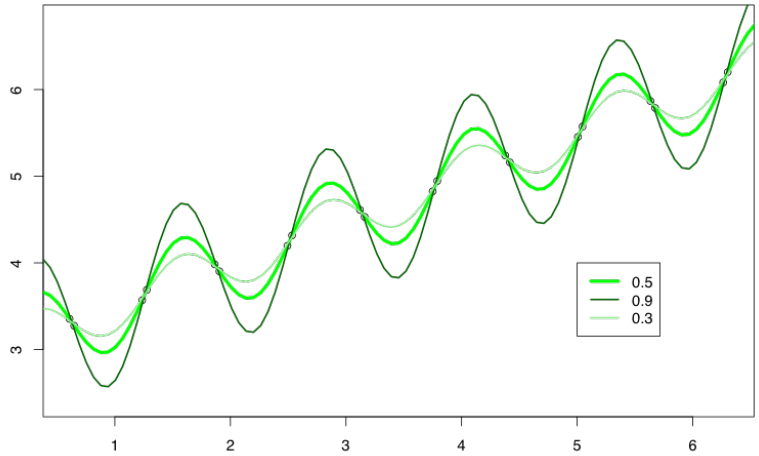
Model Variance

- Low-variance linear model: model error highly sensitive to model parameters (slope of the linear model)
- High-variance oscillatory model: model error robust to model parameters (oscillation amplitude)

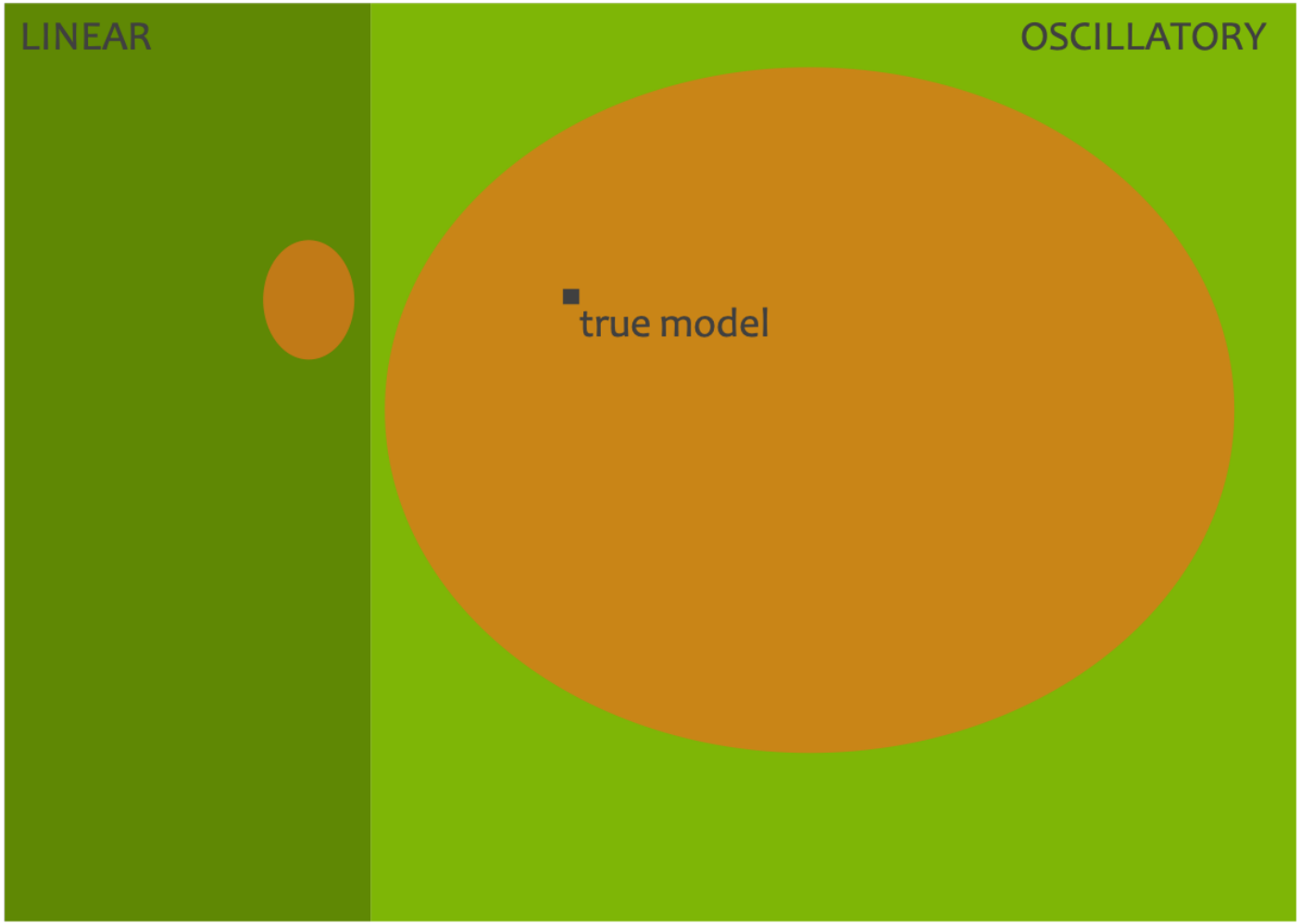
Linear



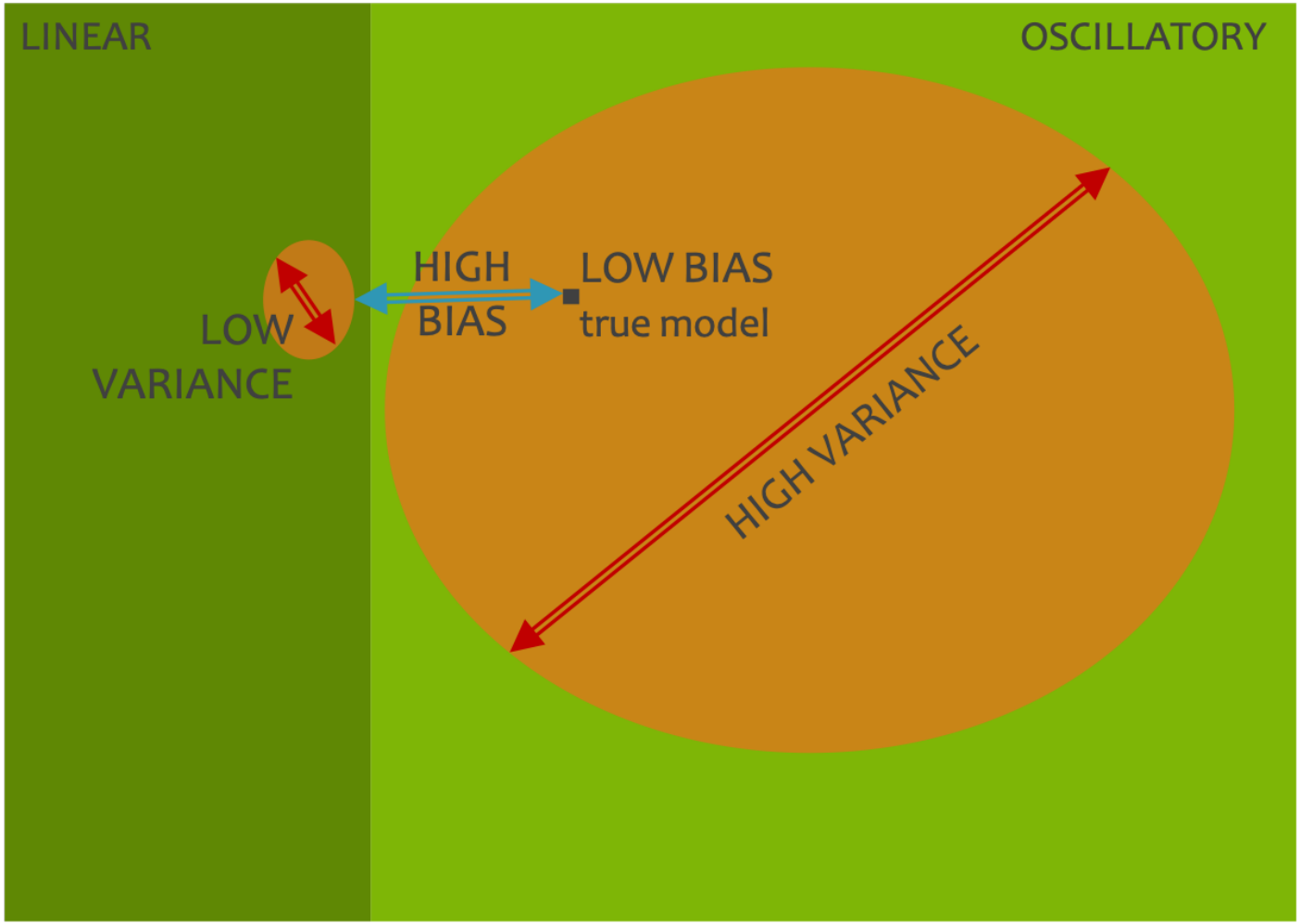
Oscillatory



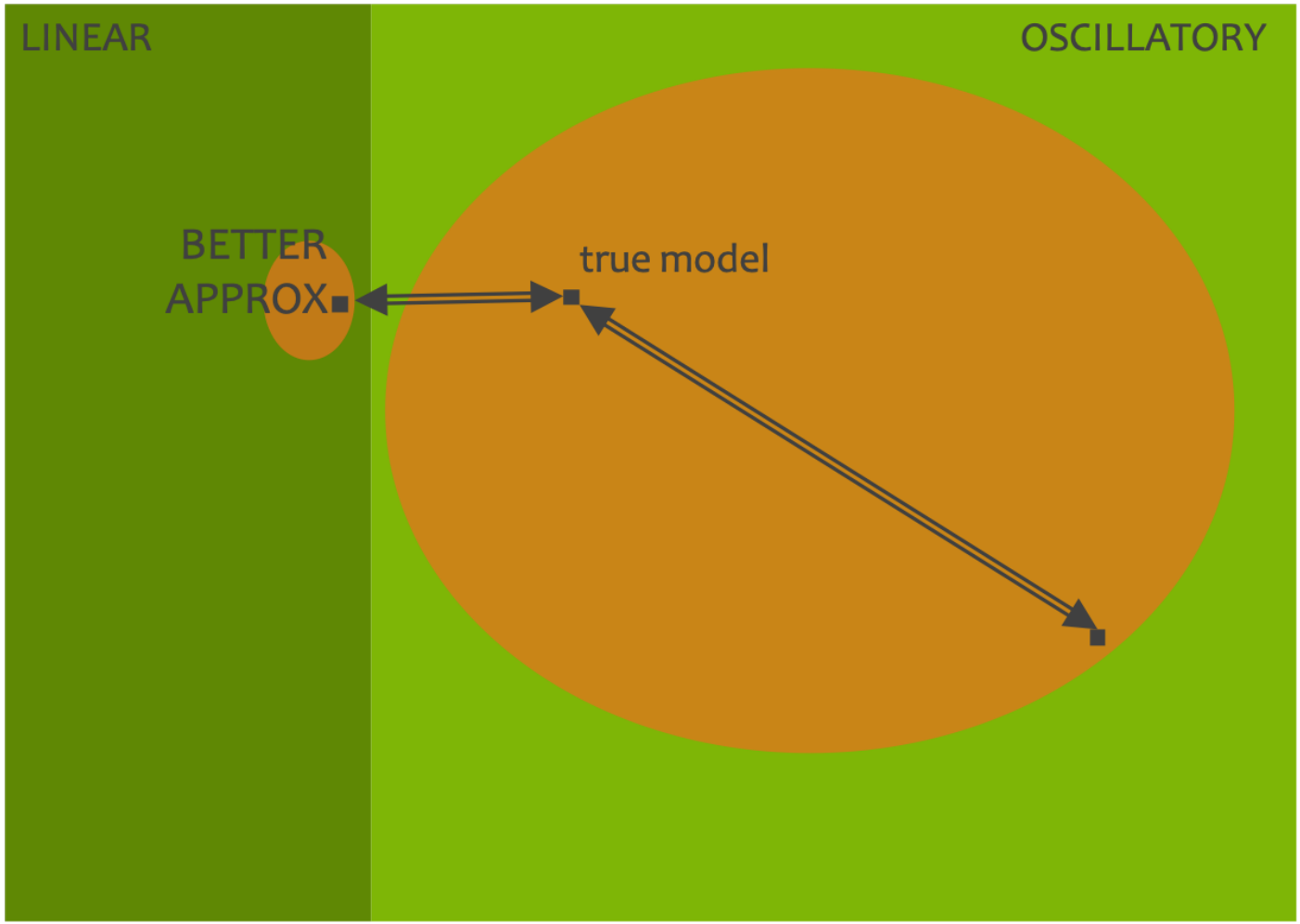
Model Spaces and Data



Model Spaces and the Bias-Variance Trade-Off



Consequence: Wrong Model Selected (Underfitting)



True (Target) Model $m : Y = m(X) + \epsilon, E[\epsilon] = 0$

Note some important properties of the true model

1. $E[m(\boldsymbol{x}_0)] = m(\boldsymbol{x}_0)$, since m is deterministic
2. $E[Y|X = \boldsymbol{x}_0] = E[m(\boldsymbol{x}_0) + \epsilon] =_{[1]} m(\boldsymbol{x}_0)$
3. $Var[Y|X = \boldsymbol{x}_0] = E[(Y - E(Y))^2|X = \boldsymbol{x}_0] =$
   $= E[(Y - m(\boldsymbol{x}_0))^2|X = \boldsymbol{x}_0] = E[\epsilon^2] = \sigma_\epsilon^2$
4. $E[Y^2|X = \boldsymbol{x}_0] = E[(m(\boldsymbol{x}_0)+\epsilon)^2] = E[m(\boldsymbol{x}_0)]^2+2E[m(\boldsymbol{x}_0)\epsilon]+E[\epsilon^2] =$
   $=_{[2,3]} m(\boldsymbol{x}_0)^2 + \sigma_\epsilon^2$

Learned Model $\hat{Y} = \hat{m}$

1. $E[\hat{m}(\boldsymbol{x}_0)^2] = E[\hat{m}(\boldsymbol{x}_0)]^2 + Var[\hat{m}(\boldsymbol{x}_0)]$,
   using the well-known variance formula $Var[U] = E[U^2] - E[U]^2$

Important note

- The learned model $\hat{m}$ is also deterministic
- However, it varies with the variation of the training set S

Decomposition of the Model Error

$$
\begin{aligned}
Err(\boldsymbol{x}_0) &= E_S[(Y - \hat{m}(\boldsymbol{x}_0))^2 | X = \boldsymbol{x}_0] \\
&= E_S[Y^2 + \hat{m}(\boldsymbol{x}_0)^2 - 2Y\hat{m}(\boldsymbol{x}_0) | X = \boldsymbol{x}_0] \\
&= E_S[Y^2 | X = \boldsymbol{x}_0] + E_S[\hat{m}(\boldsymbol{x}_0)^2] - 2E_S[Y\hat{m}(\boldsymbol{x}_0) | X = \boldsymbol{x}_0] \\
&=_{[m2,m4]} m(\boldsymbol{x}_0)^2 + \sigma_\epsilon^2 + E_S[\hat{m}(\boldsymbol{x}_0)^2] - 2m(\boldsymbol{x}_0)E_S[\hat{m}(\boldsymbol{x}_0)] \\
&=_{[\hat{m}1]} \sigma_\epsilon^2 + m(\boldsymbol{x}_0)^2 + E_S[\hat{m}(\boldsymbol{x}_0)]^2 + Var_S[\hat{m}(\boldsymbol{x}_0)] \\
&\quad - 2m(\boldsymbol{x}_0)E_S[\hat{m}(\boldsymbol{x}_0)] \\
&= \sigma_\epsilon^2 + (E_S[\hat{m}(\boldsymbol{x}_0)] - m(\boldsymbol{x}_0))^2 + Var_S[\hat{m}(\boldsymbol{x}_0)] \\
&= \sigma_\epsilon^2 + Bias^2 + Variance
\end{aligned}
$$

Components of the Model Error

Data noise term σ_ϵ^2

The learning does not influence this component.

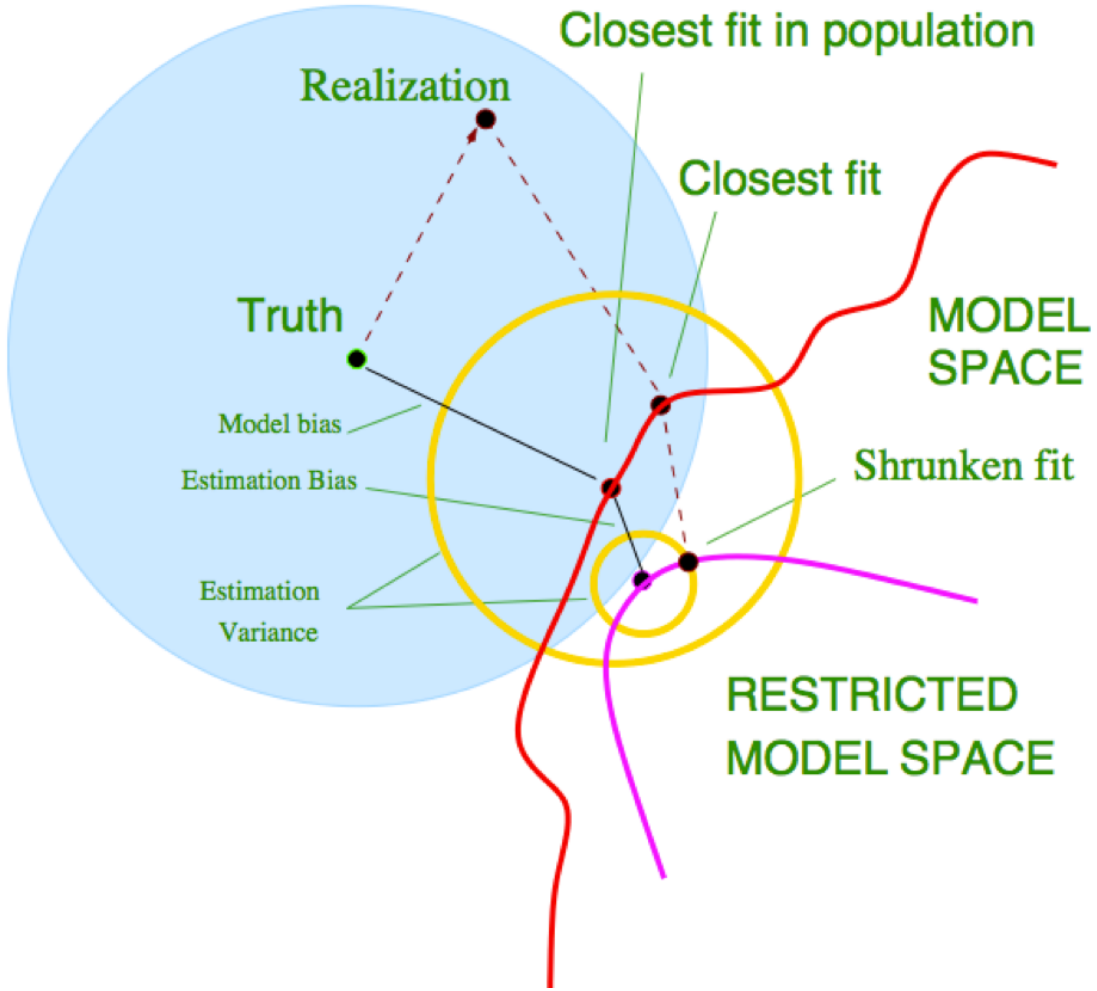
Bias term $(E_S[\hat{m}(\boldsymbol{x}_0)] - m(\boldsymbol{x}_0))^2$

- Average difference between the expected value of the model prediction $\hat{Y} = \hat{m}(\boldsymbol{x}_0)$ and the true value $Y = m(\boldsymbol{x})$
- For a model space tells us how close can $\hat{m}$ get to m

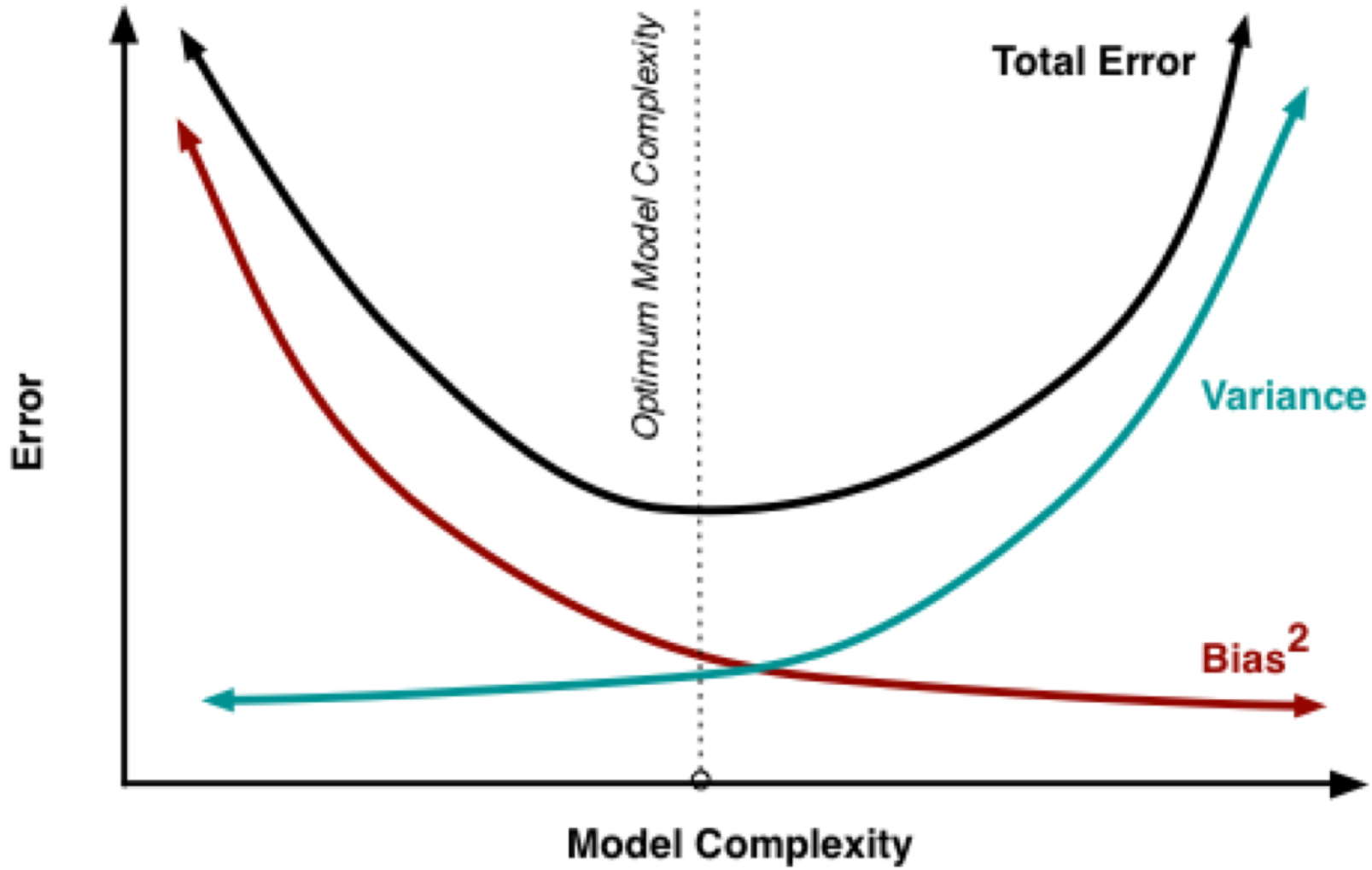
Variance term $Var_S[\hat{m}(\boldsymbol{x}_0)]$

- The variance of the prediction $\hat{Y}$ around its expected value $E_S[\hat{Y}]$
- For a model space tells us how stable are the predictions of $\hat{m}$

Bias and Variance: Graphical Interpretation



Bias-Variance Trade-Off



Overfitting vs Bias-Variance

Overfitting

- Situation when we face low bias and high variance
- Too complex model space, e.g., nearest neighbors (NN) with low k
- The right-hand side of the graph on the previous slide

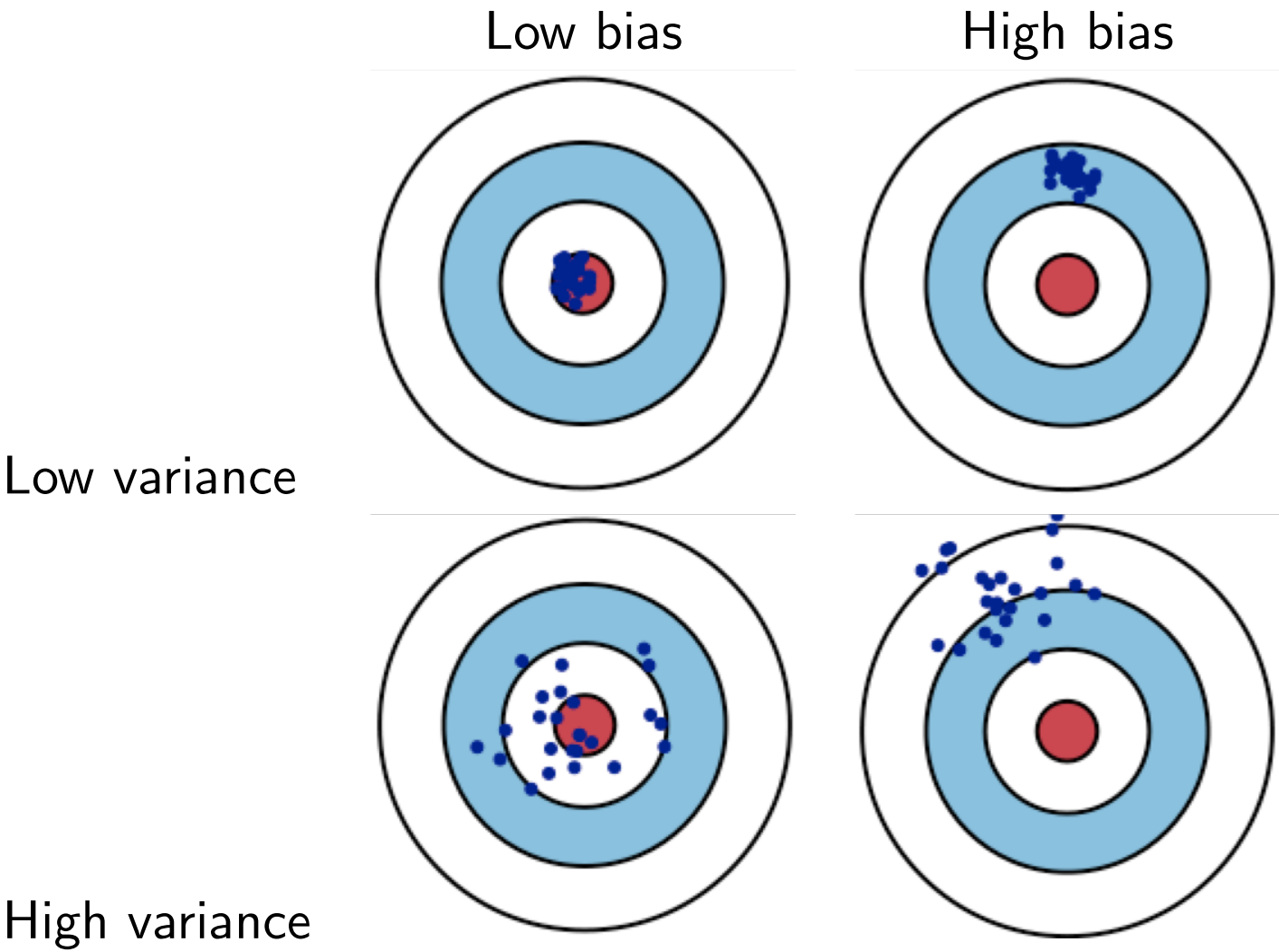
Underfitting

- Situation when we face high bias and low variance
- Too simple model space, e.g., linear models or NN with high k
- The left-hand side of the graph on the previous slide

Optimal model

- Ideal compromise between the model bias and variance
- Perfect level of the model space complexity

Overfitting vs Bias-Variance



Control over Bias and Variance

Reducing bias

- Linear models: add non-linear (higher order/degree) terms
- Nearest neighbors: lowering the number of neighbors

Reducing variance

- Linear models: regularization and variable selection
- Nearest neighbors: increasing the number of neighbors
- General method: ensembles of models

Problem: Often reducing variance increases bias and vice-versa.

Lecture Overview

Overfitting (Underfitting)

- Bias-variance decomposition
- Trade-off between bias and variance

Curse of Dimensionality

- Optimal model and nearest neighbors
- Empirical illustration and mathematical results

Software platforms for practical exercises

- R environment for statistical computing
- Python and Jypyter notebooks

Optimal Predictive Model m^*

$m^*(\boldsymbol{x}) = E[Y|\boldsymbol{X} = \boldsymbol{x}]$ **minimizes** the mean squared error $E[(Y - m(\boldsymbol{X}))^2]$.

Model estimate on a training data set S

$$m^*(\boldsymbol{x}_0) = \frac{1}{|S_0|} \sum_{(\boldsymbol{x},y)\in S_0} y, \quad S_0 = \{(\boldsymbol{x}, y) \in S : \boldsymbol{x} = \boldsymbol{x}_0\}$$

Obvious problem: training data set S contains too few (if any) examples $(\boldsymbol{x}, y)$, such that $\boldsymbol{x} = \boldsymbol{x}_0$, i.e., S_0 is a useless empty set for most $\boldsymbol{x}_0$.

Obvious Solution: Nearest Neighbors Method

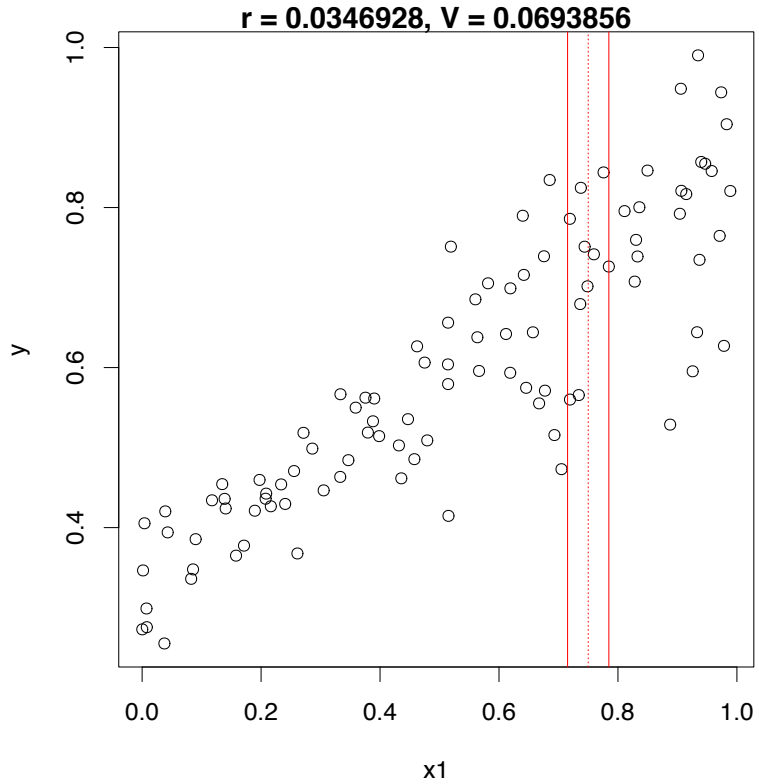
We slightly change the definition of S_0

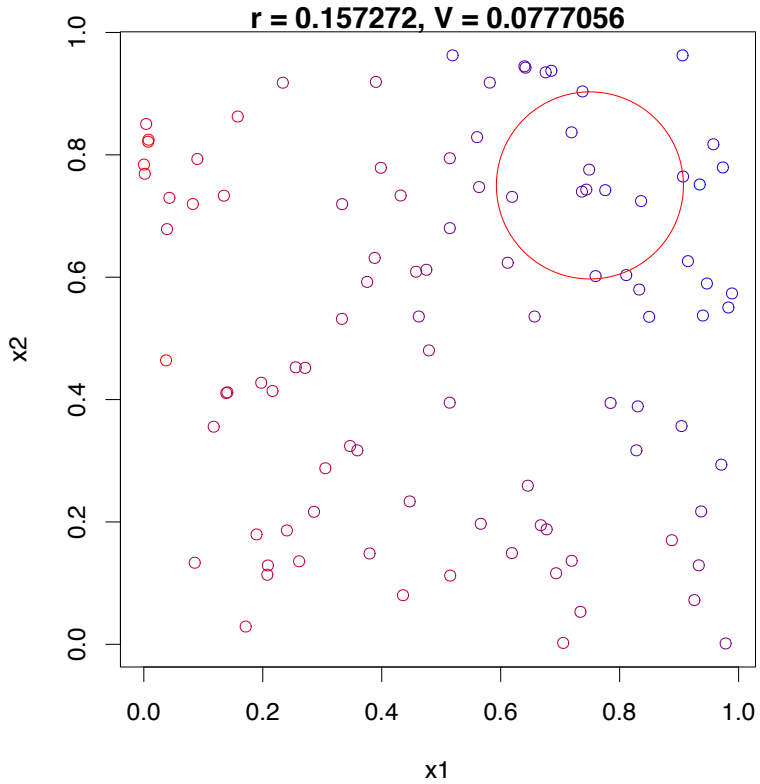
- It includes k examples that are the closest to the example $\boldsymbol{x}_0$
- Assumes a measure of distance between the examples, e.g., Euclidean

How close are the nearest neighbors?

In mathematical terms: what is the radius (volume) of S_0?

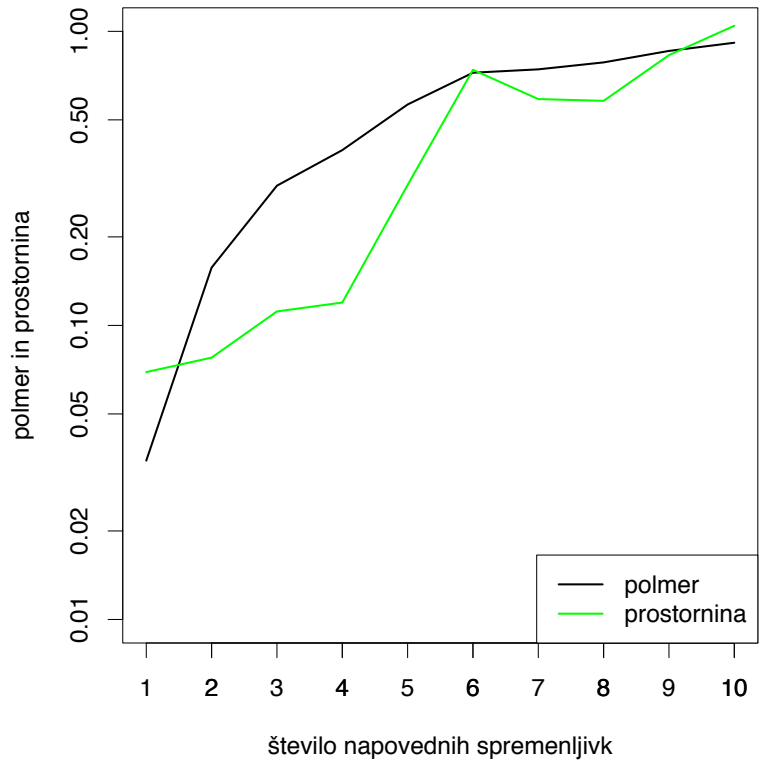
Empirically Measured Radius for $k = 10$ in 1-D and 2-D

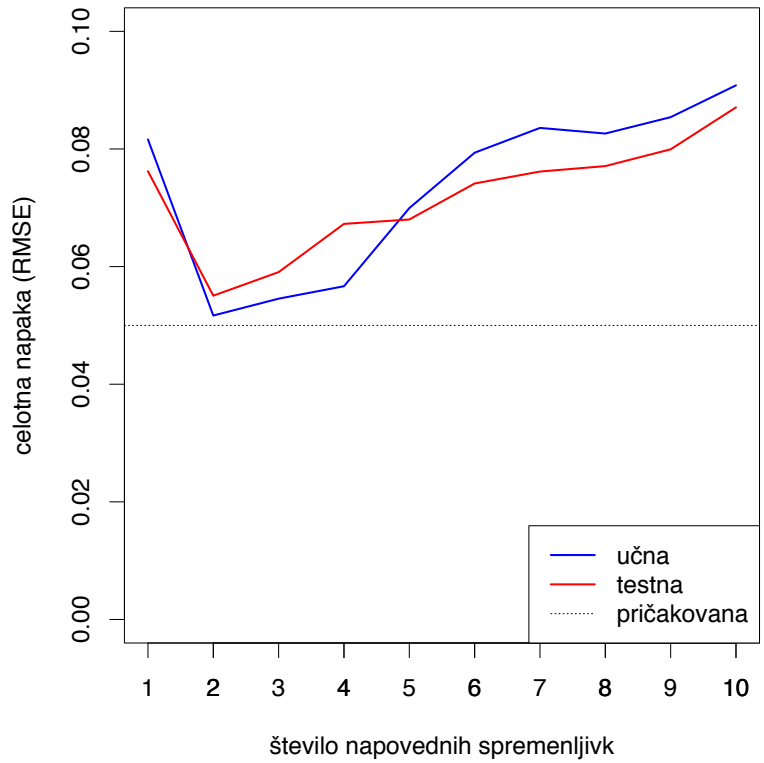




$$Y = (1 + X_1 + X_1X_2)/3 + \mathcal{N}(0, 0.05),\ D_i = [0, 1], i = 1..10, D_Y = [0, 1]$$

The Change of Radius/Volume with Dimensions ($k = 10$)





$$Y = (1 + X_1 + X_1X_2)/3 + \mathcal{N}(0, 0.05)$$

Distance D of a Random Point

Distance is observed from the center of a unit ball.

Volume of a p-dimensional ball with radius r

$$V = r^p \pi^{\frac{p}{2}} / \Gamma(p/2 + 1)$$

Note: only the first term r^p depends on the radius

The distribution of the random variable D

- $F_D(x) = P(D \leq x) = x^p / 1^p = x^p,\ 0 \leq x \leq 1$
- $f_D(x) = P(D = x) = F'_D(x) = p\, x^{p-1}$

Minimal distance M of n random points

Distance is observed from the center of a unit ball.

The distribution of the random variable M

- $f_M(x) = P(M = x) = n\,(1 - F_D(x))^{n-1}\, f_d(x) = n\,(1 - x^p)^{n-1}\, p\, x^{p-1}$
- By integrating the above function wrt x, we get

$$F_M(x) = P(M \leq x) = 1 - (1 - x^p)^n$$

The integration from the previous slide

$$
\begin{aligned}
F_M(x) &= \int_{-\infty}^{x} n\,(1-y^p)^{n-1}\,p\,y^{p-1}\,dy \\
&=_{x\geq 0} \int_{0}^{x} n\,(1-y^p)^{n-1}\,p\,y^{p-1}\,dy \\
&=_{z=y^p,\,dz=py^{p-1}\,dy} \int_{0}^{x^p} n\,(1-z)^{n-1}\,dz \\
&= -(1-z)^n\Big|_0^{x^p} \\
&= 1-(1-x^p)^n
\end{aligned}
$$

Median x_m of the minimal distance M

$$
\begin{aligned}
F_M(x) = 1 - (1 - {x_m}^p)^n &= \frac{1}{2} \\
x_m &= \left(1 - \frac{1}{2^{\frac{1}{n}}}\right)^{\frac{1}{p}}
\end{aligned}
$$

Curse of dimensionality

The median of the minimal distance of a random sample of n points increases (exponentially) with increasing dimensionality p.

Addressing the Challenges

Bias-Variance Trade-Off	Curse of Dimensionality
Ensembles (reducing variance)	Dealing with Complex Data
Support Vector Machines (reducing bias)	Embeddings
Neural networks, Deep Learning	Autoencoders (dim reduction)

Lecture Overview

Overfitting (Underfitting)

- Bias-variance decomposition
- Trade-off between bias and variance

Curse of Dimensionality

- Optimal model and nearest neighbors
- Empirical illustration and mathematical results

Software platforms for practical exercises

- R environment for statistical computing
- Python and Jypyter notebooks

R Environment for Statistical Computing

www.r-project.org

Recommended environment for using R: RStudio, rstudio.com.

Exercises involve various CRAN packages

- Implementing the machine learning methods in the course
- Ensembles, Support Vector Machines, Kernels, Neural Networks
- Second part of the course
- Specific packages to be announced later on the Web site

Python and Jypyter Notebooks

www.python.org

And support for using Jypyter notebooks, jupyter.org.

Exercises involve prepared Jypyter notebooks

- The notebooks include example running code
- Can be reused in your own future projects
- First part of the course
- Links to the notebooks will be published on the Web site