machine learning & knowledge extraction

MDPI

Review

Evaluation of Regression Models: Model Assessment, Model Selection and Generalization Error

Frank Emmert-Streib [1,2,*] **and Matthias Dehmer** [3,4,5]

1 Predictive Society and Data Analytics Lab, Faculty of Information Technolgy and Communication Sciences, Tampere University, 33100 Tampere, Finland
2 Institute of Biosciences and Medical Technology, 33520 Tampere, Finland
3 Institute for Intelligent Production, Faculty for Management, University of Applied Sciences Upper Austria, Steyr Campus, 4400 Steyr, Austria; matthias.dehmer@umit.at
4 Department of Mechatronics and Biomedical Computer Science, University for Health Sciences, Medical Informatics and Technology, 6060 Hall in Tirol, Austria
5 College of Computer and Control Engineering, Nankai University, Tianjin 300071, China
* Correspondence: v@bio-complexity.com; Tel.: +358-50-301-5353

Received: 9 February 2019; Accepted: 18 March 2019; Published: 22 March 2019

Abstract: 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, one needs to decide which model to select from candidate model families based on performance evaluations. In this paper, we review the theoretical framework of model selection and model assessment, including error-complexity curves, the bias-variance tradeoff, and learning curves for evaluating statistical models. We discuss criterion-based, step-wise selection procedures and resampling methods for model selection, whereas cross-validation provides the most simple and generic means for computationally estimating all required entities. To make the theoretical concepts transparent, we present worked examples for linear regression models. However, our conceptual presentation is extensible to more general models, as well as classification problems.

Keywords: machine learning; statistics; model selection; model assessment; regression models; high-dimensional data; data science; bias-variance tradeoff; generalization error

1. Introduction

Nowadays, "data" are at the center of our society, regardless of whether one looks at the science, industry or entertainment [1,2]. The availability of such data makes it necessary for them to be analyzed adequately, which explains the recent emergence of a new field called *data science* [3–6]. For instance, in biology, the biomedical sciences, and pharmacology, the introduction of novel sequencing technologies enabled the generation of high-throughput data from all molecular levels for the study of pathways, gene networks, and drug networks [7–11]. Similarly, data from social media can be used for the development of methods to address questions of societal relevance in the computational social sciences [12–14].

For the analysis of supervised learning models, such as regression or classification methods [15–20], allowing to estimate a prediction error model selection and model assessment are key concepts for finding the best model for a given data set. Interestingly, regarding the definition of a best model, there are two complementary approaches with a different underlying philosophy [21,22]. One is defining best model as predictiveness of a model, and the other as descriptiveness. The latter approach aims at identifying the true model, whose interpretation leads to a deeper understanding of the generated data and the underlying processes that generated the data.

Despite the importance of all these concepts, there are few reviews available on the intermediate level that formulate the goals and approaches of model selection and model assessment in a clear way. For instance, advanced reviews are presented by [21,23–27] that are either comprehensive presentations without much detail, or detailed presentations of selected topics. Furthermore, there are elementary introductions to these topics, such as by [28,29]. While accessible for beginners, these papers focus only on a small subset of the key concepts, making it hard to recognize the wider picture of model selection and model assessment.

In contrast, the focus of our review is different, with respect to the following points. First, we present the general conceptual ideas behind model selection, model assessment, and their interconnections. For this, we also present theoretical details as far as they are helpful for a deeper understanding. Second, we present practical approaches for their realization and demonstrate these by worked examples for linear polynomial regression models. This allows to close the gap between theoretical understanding and practical application. Third, our explanations aim at an intermediate level of the reader by providing background information frequently omitted in advanced texts. This should ensure that our review is useful for a broad readership with a general interest in data science. Finally, we will give information about the practical application of the methods by providing information about the availability of implementations for the statistical programming language R [30]. We focus on R because it is a widely used programming language which is freely available and forms the gold standard of the literature on statistics.

This paper is organized as follows. In the next section, we present general preprocessing steps we use before a regression analysis. Thereafter, we discuss the ordinary least squares regression, linear polynomial regression, and ridge regression, because we assume that not all readers are familiar with these models, but an understanding is necessary for the following sections. Then, we discuss the basic problem of model diagnosis, as well as its key concepts model selection and model assessment, including methods for their analysis. Furthermore, we discuss cross-validation as a flexible, generic tool that can be applied to both problems. Finally, we discuss the meaning of learning curves for model diagnosis. The paper finishes with a brief summary and conclusions.

2. Preprocessing of Data and Regression Models

In this section, we briefly review some statistical preliminaries as needed for the models discussed in the following sections. Firstly, we discuss some preprocessing steps used for standardizing the data for all regression models. Secondly, we discuss different basic regression models, with and without regularization. Thirdly, we provide information about the practical realization of such regression models by using the statistical programming language R.

2.1. Preprocessing

Let's assume we have data of the form $(\mathbf{x}_i, y_i)$ with $i \in \{1, \ldots, n\}$, where n is the number of samples. The vector $\mathbf{x}_i$ corresponds to the predictor variables for sample i, whereas $\mathbf{x}_i = (X_{i1}, \ldots, X_{ip})^T$ and p is the number of predictors; furthermore, y_i is the response variable. We denote by $\boldsymbol{y} \in \mathbb{R}^n$ the vector of response variables and by $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ the predictor matrix. The vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ corresponds to the regression coefficients.

The predictors and response variable shall be standardized in the following way:

$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^{n} X_{ij} = 0 \quad \text{for all } j \tag{1}$$

$$\bar{s}_j^2 = \frac{1}{n} \sum_{i=1}^{n} X_{ij}^2 = 1 \quad \text{for all } j \tag{2}$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i = 0 \tag{3}$$

Here, $\bar{x}_j$ and $\bar{s}_j^2$ are the mean and variance of the predictor variables, and $\bar{y}$ is the mean of the response variables.

2.2. Ordinary Least Squares Regression and Linear Polynomial Regression

The general formulation of a multiple regression model [17,31] is given by

$$y_i = \sum_{j=1}^{p} X_{ij}\beta_j + \epsilon_i. \tag{4}$$

Here, X_{ij} are p predictor variables that are linearly mapped onto the response variable y_i for sample i. The mapping is defined by the p regression coefficients β_j. Furthermore, the mapping is affected by a noise term ϵ_i assuming values in $\sim N(0, \sigma^2)$ which are normally distributed. The noise term summarizes all kinds of uncertainties, such as measurement errors.

In order to write Equation (4) more compactly but also to see the similarity between a multiple linear regression model, having p predictor variables, and a simple linear regression model, having one predictor variable, one can rewrite Equation (4) in the form:

$$y_i = f(\boldsymbol{x}_i, \boldsymbol{\beta}) + \epsilon_i = \boldsymbol{x}_i^T \boldsymbol{\beta} + \epsilon_i \tag{5}$$

Here, $\boldsymbol{x}_i^T \boldsymbol{\beta}$ is the inner product (scalar product) between the two p-dimensional vectors $\mathbf{x}_i = (X_{i1}, \ldots, X_{ip})^T$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$. One can further summarize Equation (5) for all samples $i \in \{1, \ldots, n\}$ by:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{6}$$

Here, the noise terms assumes the form $\boldsymbol{\epsilon} \sim N(0, \sigma^2 \boldsymbol{I}_n)$, whereas $\boldsymbol{I}_n$ is the $\mathbb{R}^{n \times n}$ identity matrix.

In this paper, we will show worked examples for linear polynomial regressions. The general form of this model can be written as:

$$f(\boldsymbol{x}_i, \boldsymbol{\beta}) = \sum_{i=0}^{d} \beta_i x_i^i = \beta_0 + \beta_1 x_1 + \cdots + \beta_d x_d^d. \tag{7}$$

Equation (7) is a sum of polynomials with a maximal degree of d. Interestingly, despite the fact that Equation (7) is non-linear in x_i, it is linear in the regression coefficients β_i and, hence, it can be fitted in the same way as OLS regression models. That means the linear polynomial regression model shown in Equation (7) is a linear model.

2.3. Regularization: Ridge Regression

For studying the regularization of regression models, one needs to solve optimization problems. These optimization problems are formulated in terms of norms. For a real vector $\boldsymbol{x} \in \mathbb{R}^n$ and $q \geq 1$, the Lq-norm is defined by

$$\|\boldsymbol{x}\|_q = \Big(\sum_{i=1}^{n} |x_i|^q \Big)^{\frac{1}{q}}. \tag{8}$$

For the special case $q = 2$, one obtains the L2-norm (also known as the Euclidean norm) used for ridge regression and for $q = 1$ the L1-norm, which is, for instance, used by the LASSO [32].

The motivation for improving OLS comes from the observation that OLS models often have a low bias but large variance; put simply, this means the models are too complex for the data. In order to reduce the complexity of models, regularized regressions are used. The regularization leads either to a shrinking of the values of the regression coefficients, or to a vanishing of the coefficients (i.e., a value of zero) [33].

A base example for regularized regression is Ridge regression, introduced in [34]. Ridge regression can be formulated as follows:

$$\hat{\boldsymbol{\beta}}^{RR} = arg\min \left\{ \frac{1}{2n} \text{RSS}(\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_2^2 \right\} \quad (9)$$

$$= arg\min \left\{ \frac{1}{2n} \|(\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \right\} \quad (10)$$

Here, $\text{RSS}(\boldsymbol{\beta})$ is the residual sum of squares (RSS) called the loss of the model, $\lambda\|\boldsymbol{\beta}\|_2^2$ is the regularization term or penalty, and λ is the tuning or regularization parameter. The parameter λ controls the shrinkage of coefficients. The L2-penalty in Equation (10) is also sometimes called the Tikhonov regularization.

Overall, the advantage of a ridge regression and general regularized regression model is that regularization can reduce the variance by increasing the bias. Interestingly, this can improve the prediction accuracy of a model [19].

2.4. R Package

OLS regression is included in the base functionality of R. In order to preform regularized regression, the package *glmnet* [35] can be used. This package is very flexible, allowing to perform a variety of different regularized regression models, including ridge regression, LASSO, adaptive LASSO [36], and elastic net [37].

3. Overall View on Model Diagnosis

Regardless of what statistical model one is studying, e.g., for classification or regression, there are two basic questions one needs to address: (1) How can one choose between competing models, and (2) how can one evaluate them? Both questions aim at the diagnosis of models.

The above informal questions are formalized by the following two statistical concepts [18]:

> Model selection: Estimate the performance of different models in order to choose the best model.
> Model assessment: For the best model, estimate its generalization error.

Briefly, model selection refers to the process of optimizing a model family or model candidate. This includes the selection of a model itself from a set of potentially available models, and the estimation of its parameters. The former can relate to deciding which regularization method (e.g., ridge regression, LASSO, or elastic net) should be used, whereas the latter corresponds to estimating the parameters of the selected model. On the other hand, model assessment means the evaluation of the generalization error (also called test error) of the finally selected model for an independent data set. This task aims at estimating the "true prediction error" as could be obtained from an infinitely large test data set. What both concepts have in common is that they are based on the utilization of data to quantify properties of models numerically.

For simplicity, let's assume that we have been given a very large (or arbitrarily large) data set, D. The best approach for both problems would be to randomly divide the data into three non-overlapping sets:

1. Training data set: D_{train}
2. Validation data set: D_{val}
3. Test data set: D_{test}

By "very large data set", we mean a situation where the sample sizes—that is, n_{train}, n_{val}, and n_{test} for all three data sets are large without necessarily being infinite, but where an increase in their sizes

would not lead to changes in the model evaluation. Formally, the relation between the three data sets can be written as:

$$D = D_{train} \cup D_{val} \cup D_{test} \tag{11}$$
$$\emptyset = D_{train} \cap D_{val} \tag{12}$$
$$\emptyset = D_{train} \cap D_{test} \tag{13}$$
$$\emptyset = D_{val} \cap D_{test} \tag{14}$$

Based on these data, the training set would be used to estimate or learn the parameters of the models. This is called "model fitting". The validation data would be used to estimate a selection criterion for model selection, and the test data would be used for estimating the generalization error of the final chosen model.

In practice, the situation is more complicated due to the fact that D is typically not arbitrarily large. In the following sections, we discuss first model assessment and then model selection in detail. The order of our discussion is reversed to the order in which one would perform a practical analysis. However, for reasons of understanding the concepts, this order is beneficial.

4. Model Assessment

Let's assume we have a general model of the form:

$$y = f(x, \boldsymbol{\beta}) + \epsilon \tag{15}$$

mapping the input x to the output y as defined by the function f. The mapping varies by a noise term $\epsilon \sim N(0, \sigma^2)$ representing, for example, measurement errors. We want to approximate the true (but unknown) mapping function f by a model g that depends on parameters $\boldsymbol{\beta}$, that is,

$$\hat{y} = g(x, \hat{\boldsymbol{\beta}}(D)) = \hat{g}(x, D). \tag{16}$$

Here, the parameters $\boldsymbol{\beta}$ are estimated from a training data set D (strictly denoted by D_{train}), making the parameters a function of the training set $\hat{\boldsymbol{\beta}}(D)$. The "hat" indicates that the parameters $\boldsymbol{\beta}$ are estimates of the data D. As a short-cut, we are writing $\hat{g}(x, D)$ instead of $g(x, \hat{\boldsymbol{\beta}}(D))$.

Based on these entities, we can define the following model evaluation measures:

$$\text{SST} = \text{TSS} = \sum_{i=1}^{n} (y_i - \bar{y})^2 = \|\boldsymbol{Y} - \bar{\boldsymbol{Y}}\|^2 \tag{17}$$
$$\text{SSR} = \text{ESS} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 = \|\hat{\boldsymbol{Y}} - \bar{\boldsymbol{Y}}\|^2 \tag{18}$$
$$\text{SSE} = \text{RSS} = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 = \sum_{i=1}^{n} e_i^2 = \|\hat{\boldsymbol{Y}} - \boldsymbol{Y}\|^2 \tag{19}$$

Here, $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$ is the mean value of the predictor variable, and $e_i = \hat{y}_i - y_i$ are the residuals; furthermore:

- SST is the **sum of squares total**, also called the **total sum of squares** (TSS);
- SSR is the **sum of squares due to regression** (variation explained by linear model), also called the **explained sum of squares** (ESS);
- SSE is the **sum of squares due to errors** (unexplained variation), also called the **residual sum of squares** (RSS).

There is a remarkable property for the sum of squares given by:

$$\underbrace{\text{SST}}_{\text{total deviation}} = \underbrace{\text{SSR}}_{\text{deviation of regression from mean}} + \underbrace{\text{SSE}}_{\text{deviation of regression}} \tag{20}$$

This relation is called *partitioning of the sum of squares* [31].

Furthermore, for summarizing the overall predictions of a model, the **mean squared error** (MSE) is useful, given by

$$\text{MSE} = \frac{\text{SSE}}{n}. \tag{21}$$

The general problem when dealing with predictions is that we would like to know about the generalization abilities of our model. Specifically, for a given training data set D_{train}, we can estimate the parameters of our model $\boldsymbol{\beta}$ leading to estimates $g(x, \hat{\boldsymbol{\beta}}(D_{train}))$. Ideally, we would like to have that $y \approx \hat{g}(x, D_{train})$ for any data point (x, y). In order to assess this quantitatively, a loss function, simply called "loss", is defined. Frequent choices are the absolute error

$$L(y, \hat{g}(x, D_{train})) = \left| y - \hat{g}(x, D_{train}) \right| \tag{22}$$

or the squared error

$$L(y, \hat{g}(x, D_{train})) = \left(y - \hat{g}(x, D_{train}) \right)^2. \tag{23}$$

If one would use only the data points from a training set, i.e., $(x, y) \in D_{train}$ to assess the loss, these estimates are usually overly optimistic and lead to much smaller estimates than if data points are used from all possible values (i.e., $(x, y) \sim P$) whereas P is the distribution of all possible values. Formally, we can write this as expectation values of the respective data,

$$E_{test}(D_{train}, n_{train}) = \mathbb{E}_P \left[L(y, \hat{g}(x, D_{train})) \right]. \tag{24}$$

The expectation value in Equation (24) is called the generalization error of the model given by $\hat{\boldsymbol{\beta}}(D_{train})$. This error is also called *out-of-sample error*, or simply test error. The latter name emphasizes the important fact that test data are used for the evaluation of the prediction error (as represented by the distribution P) of the model, but training data are used to learn its parameters (as indicated by D_{train}).

From Equation (24), one can see that we have an unwanted dependency on the training set D_{train}. In order to remove this, we need to assess the generalization error of the model given by $\hat{\boldsymbol{\beta}}(D_{train})$ by forming the expectation value with respect to all training sets, i.e.,

$$E_{test}(n_{train}) = \mathbb{E}_{D_{train}} \mathbb{E}_P \left[L(y, \hat{g}(x, D_{train})) \right]. \tag{25}$$

This is the expected generalization error of the model, which is no longer dependent on any particular estimates of $\hat{\beta}(D_{train})$. Hence, this error provides the desired assessment of a model. Equation (25) is also called *expected out-of-sample error* [38]. It is important to emphasize that the training sets D_{train} are not infinitely large, but all have the same finite sample size n_{train}. Hence, the expected generalization error in Equation (25) is independent of a particular training set but dependent on the size of these sets. This dependency will be explored in Section 7 when we discuss learning curves.

On a practical note, we would like to say that in practice, we do not have *all* data available—instead, we have one (finite) data set, D, which we need to utilize in an efficient way to approximate P for estimating the generalization error of the model in Equation (25). The gold-standard

approach for this is cross-validation (CV), and we discuss practical aspects thereof in Section 6. However, in the following, we focus first on theoretical aspects of the generalization error of the model.

4.1. Bias-Variance Tradeoff

It is interesting that the above generalization error of the model in Equation (25) can be decomposed into different components. In the following, we derive this decomposition which is known as the bias–variance tradeoff [39–42]. We will see that this decomposition provides valuable insights for understanding the influence of the model complexity on the prediction error.

In the following, we denote the training set briefly by D to simplify the notation. Furthermore, we write the expectation value with respect to distribution P as $\mathbb{E}_{x,y}$, and not as $\mathbb{E}_P$ as in Equation (25), because this makes the derivation more explicit. This argument will become clear when discussing the Equations (31) and (34).

$$\begin{aligned}
\mathbb{E}_D\,\mathbb{E}_{x,y}\left[\left(y-\hat{g}(x,D)\right)^2\right] &= \underbrace{\mathbb{E}_D\,\mathbb{E}_{x,y}}_{\text{independent}}\left[\left(y-E_D\left[\hat{g}(x,D)\right]+E_D\left[\hat{g}(x,D)\right]-\hat{g}(x,D)\right)^2\right] && (26)\\
&= \mathbb{E}_{x,y}\,\mathbb{E}_D\left[\left(y-\mathbb{E}_D\left[\hat{g}(x,D)\right]\right)^2\right]+\mathbb{E}_{x,y}\,\mathbb{E}_D\left[\left(\mathbb{E}_D\left[\hat{g}(x,D)\right]-\hat{g}(x,D)\right)^2\right]+\\
&+ 2\,\mathbb{E}_{x,y}\,\mathbb{E}_D\left[\left(y-\mathbb{E}_D\left[\hat{g}(x,D)\right]\right)\left(\mathbb{E}_D\left[\hat{g}(x,D)\right]-\hat{g}(x,D)\right)\right] && (27)\\
&= \mathbb{E}_{x,y}\,\mathbb{E}_D\left[\left(y-\mathbb{E}_D\left[\hat{g}(x,D)\right]\right)^2\right]+\mathbb{E}_{x,y}\,\mathbb{E}_D\left[\left(\mathbb{E}_D\left[\hat{g}(x,D)\right]-\hat{g}(x,D)\right)^2\right]+\\
&+ 2\,\mathbb{E}_{x,y}\left[\underbrace{\left(y-\mathbb{E}_D\left[\hat{g}(x,D)\right]\right)}_{\text{independent of } D}\mathbb{E}_D\left[\left(\mathbb{E}_D\left[\hat{g}(x,D)\right]-\hat{g}(x,D)\right)\right]\right] && (28)\\
&= \mathbb{E}_{x,y}\left[\left(y-\mathbb{E}_D\left[\hat{g}(x,D)\right]\right)^2\right]+\mathbb{E}_{x,y}\,\mathbb{E}_D\left[\left(\mathbb{E}_D\left[\hat{g}(x,D)\right]-\hat{g}(x,D)\right)^2\right] && (29)\\
&= \mathbb{E}_{x,y}\left[\left(y-\bar{g}(x)\right)^2\right]+\mathbb{E}_{x,y}\,\mathbb{E}_D\left[\left(\bar{g}(x)-\hat{g}(x,D)\right)^2\right] && (30)\\
&= \mathbb{E}_{x,y}\left[\left(y-\bar{g}(x)\right)^2\right]+\mathbb{E}_D\,\mathbb{E}_x\,\mathbb{E}_{y|x}\left[\left(\underbrace{\bar{g}(x)-\hat{g}(x,D)}_{\text{independent of } y}\right)^2\right] && (31)\\
&= \mathbb{E}_{x,y}\left[\left(y-\bar{g}(x)\right)^2\right]+\mathbb{E}_D\,\mathbb{E}_x\left[\left(\bar{g}(x)-\hat{g}(x,D)\right)^2\right] && (32)\\
&= \text{bias}^2+\text{variance}
\end{aligned}$$

In Equations (28) and (31) we used the independence of the sampling processes for D and x, y to change the order of the expectation values. This allowed us to evaluate the conditional expectation value $\mathbb{E}_{y|x}$, because the argument is independent of y.

In Equation (30), we used the short form

$$\bar{g}(x)=\mathbb{E}_D\left[\hat{g}(x,D)\right] \tag{33}$$

to write the expectation value of $\hat{g}$ with respect to D, giving a mean model $\bar{g}$ over all possible training sets D. Due to the fact that this expectation value integrates over all possible values of D, the resulting $\bar{g}(x)$ no longer depends on it.

By utilizing the conditional expectation value

$$\mathbb{E}_{x,y}=\mathbb{E}_x\,\mathbb{E}_{y|x} \tag{34}$$

we can further analyze the first term of the above derivation (highlighted in green) by making use of

$$\mathbb{E}_{x,y}\,y=\mathbb{E}_x\,\mathbb{E}_{y|x}\,y=\mathbb{E}_x\,\bar{y}(x)=\bar{y}. \tag{35}$$

Here, it is important to note that $\bar{y}(x)$ is a function of x, whereas $\bar{y}$ is not because the expectation value $\mathbb{E}_x$ integrates over all possible values of x. For reasons of clarity, we want to note that y actually means $y(x)$, but for notational simplicity we suppress this argument in order to make the derivation more readable.

Specifically, by utilizing this term, we obtain the following decomposition:

$$\begin{aligned}
\mathbb{E}_{x,y}\left[(y-\bar{g}(x))^2\right] &= \mathbb{E}_{x,y}\left[(y-\mathbb{E}_D\left[\hat{g}(x,D)\right])^2\right] && (36)\\
&= \mathbb{E}_{x,y}\left[(y-\bar{y}(x)+\bar{y}(x)-\mathbb{E}_D\left[\hat{g}(x,D)\right])^2\right] && (37)\\
&= \mathbb{E}_{x,y}\left[(y-\bar{y}(x))^2\right]+\mathbb{E}_{x,y}\left[(\bar{y}(x)-\mathbb{E}_D\left[\hat{g}(x,D)\right])^2\right]+\\
&+ 2\,\mathbb{E}_{x,y}\left[(y-\bar{y}(x))(\bar{y}-\mathbb{E}_D\left[\hat{g}(x,D)\right])\right] && (38)\\
&= \mathbb{E}_{x,y}\left[(y-\bar{y}(x))^2\right]+\mathbb{E}_{x,y}\left[(\underbrace{\bar{y}(x)-\mathbb{E}_D\left[\hat{g}(x,D)\right]}_{\text{independent of } y})^2\right]+\\
&+ 2\,\mathbb{E}_x\,\mathbb{E}_{y|x}\left[(y-\bar{y}(x))(\underbrace{\bar{y}(x)-\mathbb{E}_D\left[\hat{g}(x,D)\right]}_{\text{independent of } y})\right] && (39)\\
&= \mathbb{E}_{x,y}\left[(y-\bar{y}(x))^2\right]+\mathbb{E}_{x,y}\left[(\bar{y}(x)-\mathbb{E}_D\left[\hat{g}(x,D)\right])^2\right]+\\
&+ 2\,\mathbb{E}_x\left[(\bar{y}(x)-\bar{y}(x))(\bar{y}(x)-\mathbb{E}_D\left[\hat{g}(x,D)\right])\right] && (40)\\
&= \mathbb{E}_{x,y}\left[(y-\bar{y}(x))^2\right]+\mathbb{E}_x\left[(\bar{y}(x)-\mathbb{E}_D\left[\hat{g}(x,D)\right])^2\right] && (41)\\
&= \text{Noise}+\text{Bias}^2
\end{aligned}$$

Taken together, we obtain the following combined result:

$$\mathbb{E}_D\,\mathbb{E}_{x,y}\left[(y-\hat{g}(x,D))^2\right] = \tag{42}$$

$$\begin{aligned}
&\mathbb{E}_{x,y}\left[(y-\bar{y}(x))^2\right]+\mathbb{E}_x\,\mathbb{E}_D\left[(\mathbb{E}_D\left[\hat{g}(x,D)\right]-\hat{g}(x,D))^2\right]+\mathbb{E}_x\left[(\bar{y}(x)-\mathbb{E}_D\left[\hat{g}(x,D)\right])^2\right]\\
&= \mathbb{E}_{x,y}\left[(y-\bar{y}(x))^2\right]+\mathbb{E}_x\,\mathbb{E}_D\left[(\bar{g}(x)-\hat{g}(x,D))^2\right]+\mathbb{E}_x\left[(\bar{y}(x)-\bar{g}(x))^2\right] && (43)\\
&= \text{Noise}+\text{Variance}+\text{Bias}^2
\end{aligned}$$

- Noise: This term measures the variability within the data, not considering any model. The noise cannot be reduced because it does not depend on the training data D or g, or any other parameter under our control; hence, it is a characteristic of the data. For this reason, this component is also called "irreducible error".
- Variance: This term measures the model variability with respect to changing training sets. This variance can be reduced by using less complex models, g. However, this can increase the bias (underfitting).
- Bias: This term measures the inherent error that you obtain from your model, even with infinite training data. This bias can be reduced by using more complex models, g. However, this can increase the variance (overfitting).

Figure 1 shows a visualization of the model assessment problem and the bias-variance tradeoff. In Figure 1A, the blue curve corresponds to a model family—that is, a regression model with a fixed number of covariates—and each point along this line corresponds to a particular model obtained from estimating the parameters of the model from a data set. The dark-green point corresponds to the true (but unknown) model and a data set generated by this model. Specifically, this data set has been obtained in the error-free case, i.e., $\epsilon_i = 0$ for all samples, i. If another data set is generated from the

true model, this data set will vary to some extent because of the noise term ϵ_i, which is usually not zero. This variation is indicated by the large (light) green circle around the true model.

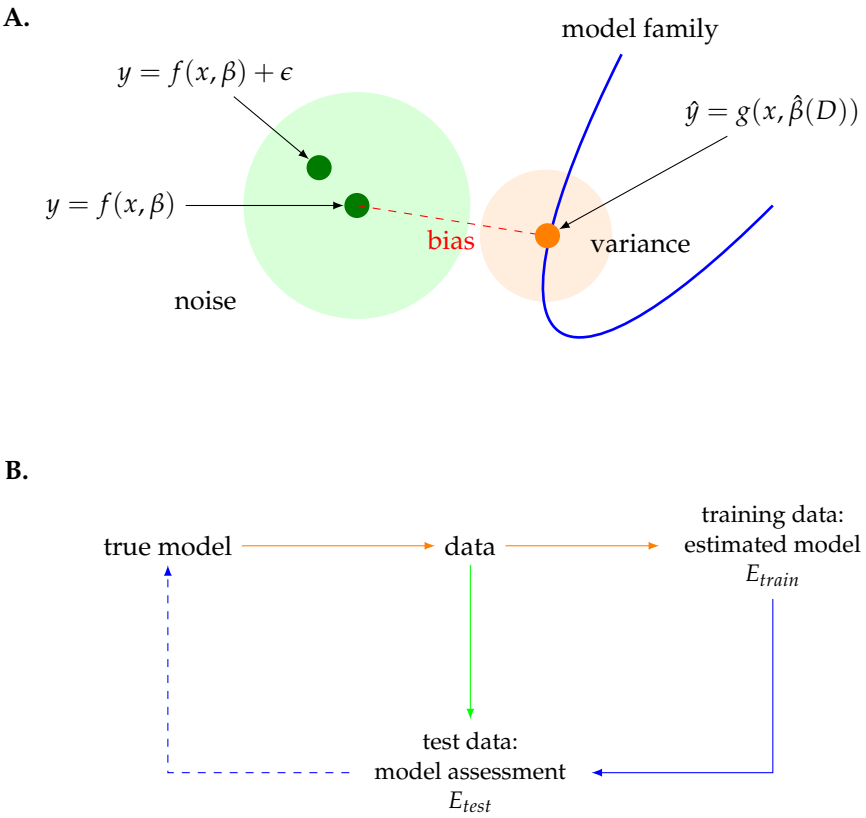

Figure 1. Idealized visualization of the model assessment and bias–variance tradeoff.

In case the model family does not include the true model, there will be a bias corresponding to the distance between the true model and the estimated model, indicated by the orange point along the curve of the model family. Specifically, this bias is measured between the error-free data set generated by the true model and the estimated model based on this data set. Also the estimated model will have some variability indicated by the (light) orange circle around the estimated model. This corresponds to the variance of the estimated model.

It is important to realize that there is no possibility of directly comparing the true model and the estimated model with each other because the true model is usually unknown. Instead, this comparison is carried out indirectly via data that have been generated by the true model. Hence, these data are serving two purposes. Firstly, they are used to estimate the parameters of the model, where the training data are used. If one uses the same training data to evaluate the prediction error of this model, the prediction error is called training error

$$E_{train} = E_{train}(D_{train}). \tag{44}$$

E_{train} is also called in-sample error. Secondly, they are used to assess the estimated model by quantifying its prediction error, and for this estimation the test data are used. For this reason, the prediction error is called test error

$$E_{test} = E_{test}(D_{test}). \tag{45}$$

In order to emphasize this, we visualized this process in Figure 1B.

It is important to note that a prediction error is always evaluated with respect to a given data set. For this reason, we emphasized this explicitly in Equations (44) and (45). However, usually this information is omitted whenever it is clear which data set has been used.

We want to emphasize that the training error is only defined as a sample estimate but not as a population estimate, because the training data set is always finite. That means Equation (44) is estimated by

$$E_{train} = \frac{1}{n_{train}} \sum_{i=1}^{n_{train}} L(y, \hat{g}(x, D_{train})) \tag{46}$$

assuming the sample size of the training data is n_{train}. In contrast, the test error in Equation (45) corresponds to the population estimate given in Equation (25). In practice, this can be approximated by a sample estimate, similar to Equation (46), of the form

$$E_{test} = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} L(y, \hat{g}(x, D_{train})) \tag{47}$$

for a test data set with n_{test} samples.

4.2. Example: Linear Polynomial Regression Model

Figure 2 presents an example. Here, the true model is shown in blue, corresponding to

$$f(x, \boldsymbol{\beta}) = 25 + 0.5x + 4x^2 + 3x^3 + x^4 \tag{48}$$

whereas $\boldsymbol{\beta} = (25, 0.5, 4, 3, 1)^T$ (see Equation (15)). The true model is a mixture of polynomials of different degrees, whereas the highest degree is 4, corresponding to a linear polynomial regression model. From this model, we generate training data with a sample size of $n = 30$ (shown by black points) that we use to fit different regression models.

The general model family we use for the regression model is given by

$$g(x, \boldsymbol{\beta}) = \sum_{i=0}^{d} \beta_i x^i = \beta_0 + \beta_1 x + \cdots + \beta_d x^d. \tag{49}$$

That means we are fitting linear polynomial regression models with a maximal degree of d. The highest degree corresponds to the model complexity of the polynomial family. For our analysis, we are using polynomials with degree d from 1 to 10, and we fit these to the training data. The results of these regression analyses are shown as red curves in Figure 2A–J.

In Figure 2A–J, the blue curves show the true model, the red curves the fitted models, and the black points correspond to the training data. These results correspond to individual model fits—that is, no averaging has been performed. Furthermore, for all results, the sample size of the training data was kept fixed (varying sample sizes are studied in Section 7). Because the model degree indicates the complexity of the fitted model, the shown models correspond to different model complexities, from low-complexity ($d = 1$) to high-complexity ($d = 10$) models.

One can see that for both low and high degrees of the polynomials, there are clear differences between the true model and the fitted models. However, these differences have a different origin. For low-degree models, the differences come from the low complexity of the models which are not flexible enough to adapt to the variability of the training data. Put simply, the model is too simple. This behavior corresponds to an underfitting of the data (caused by high bias, as explained in detail below). In contrast, for high degrees, the model is too flexible for the few available training samples. In this case, the model is too complex for the training data. This behavior corresponds to an overfitting of the data (caused by high variance, as explained in detail below).

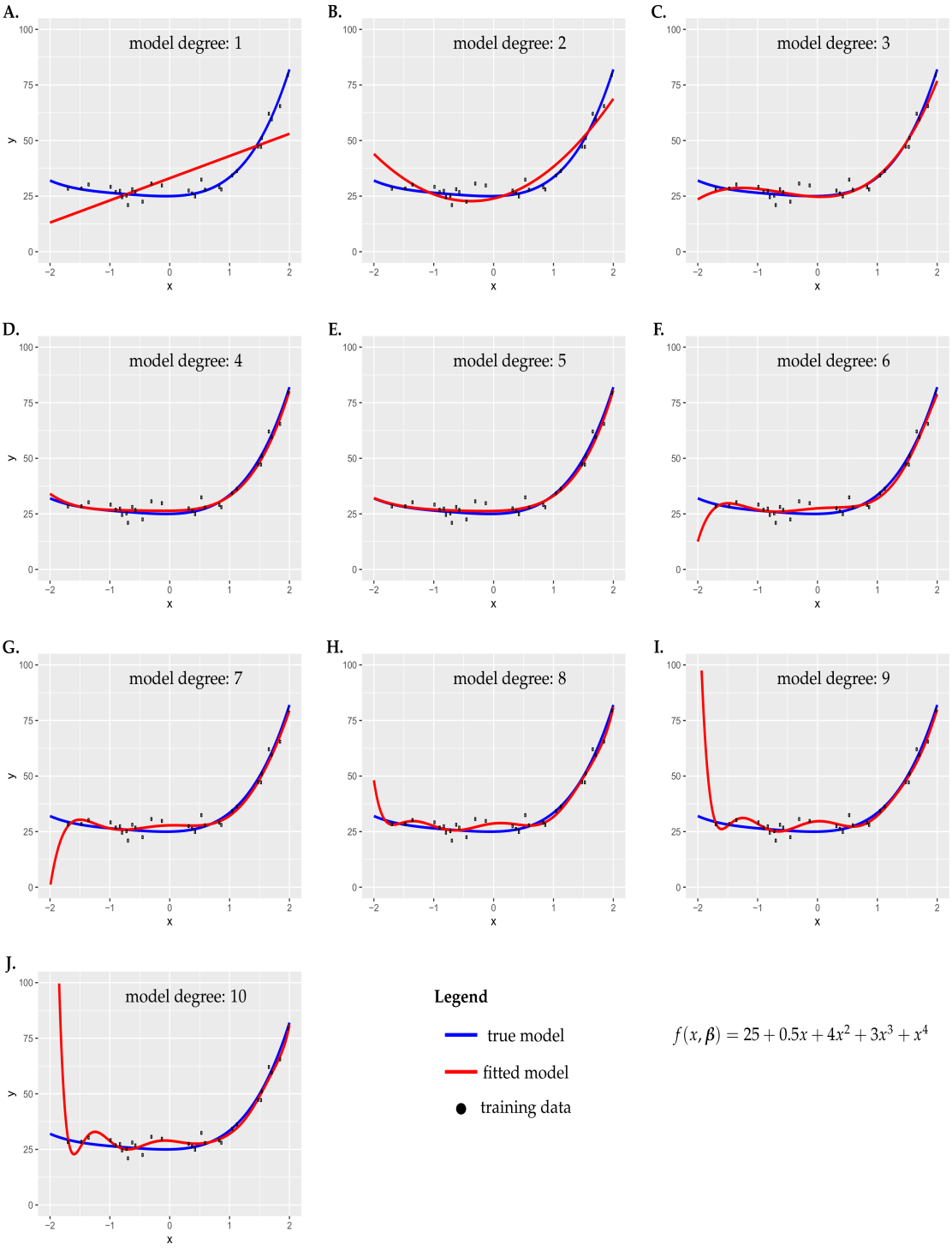

Figure 2. Different examples for fitted linear polynomial regression models of varying degree d, ranging from 1 to 10. The model degree indicates the highest polynomial degree of the fitted model. These models correspond to different model complexities, from low-complexity ($d = 1$) to high-complexity ($d = 10$) models. The blue curves show the true model, the red curves show the fitted models, and the black points correspond to the training data. The shown results correspond to individual fits—that is, no averaging has been performed. For all results, the sample size of the training data was kept fixed.

A different angle to the above results can be obtained by showing the expected training and test errors for the different polynomials. This is shown in Figure 3.

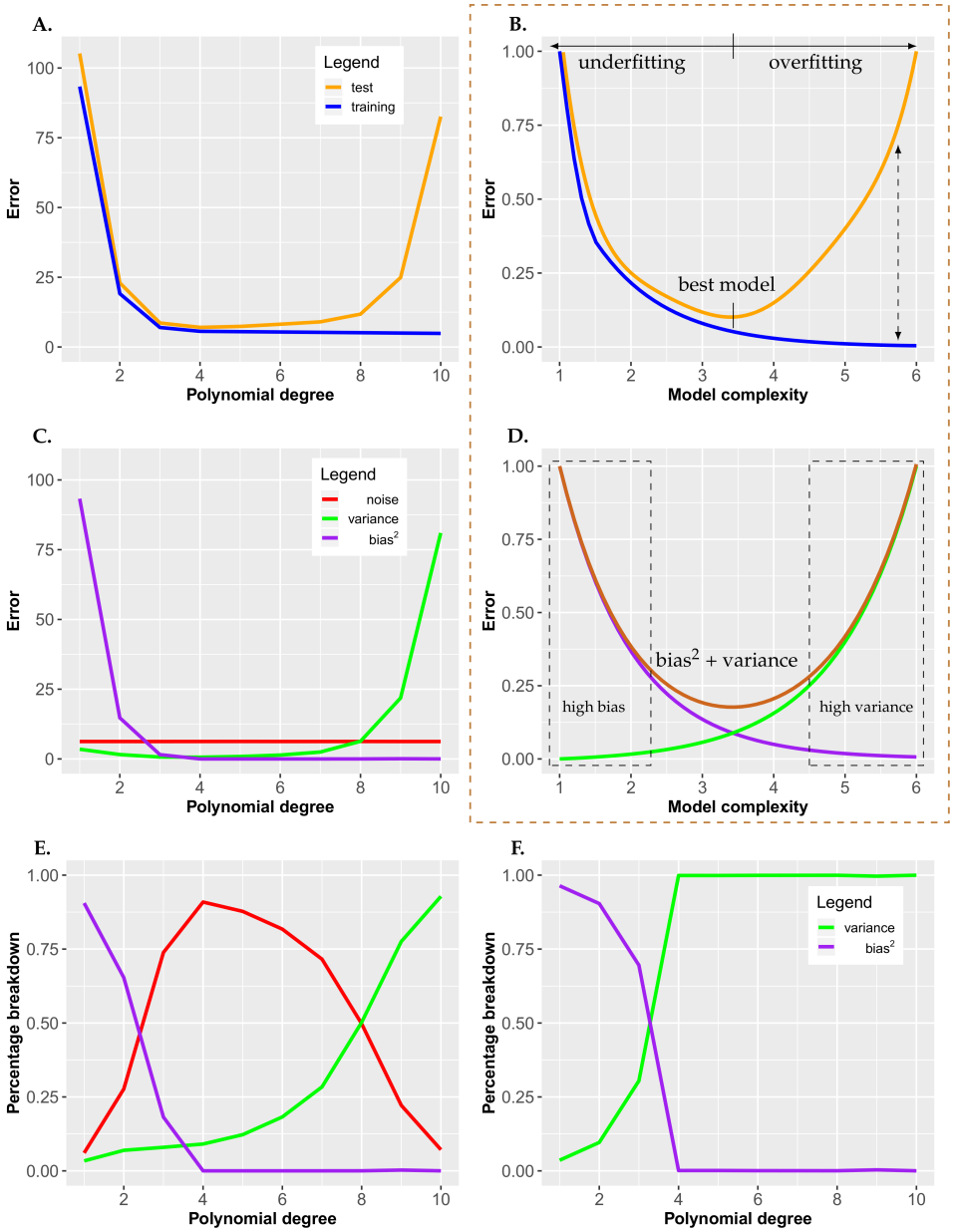

Figure 3. Error-complexity curves showing the prediction error (training and test error) in dependence on the model complexity. (**A**,**C**,**E**,**F**) show numerical simulation results for a linear polynomial regression model. The model complexity is expressed by the degree of the highest polynomial. For this analysis, the training data set was fixed. (**B**) Idealized error curves for general statistical models. (**C**) Decomposition of the expected generalization error (test error) into noise, bias, and variance. (**D**) Idealized decomposition in bias and variance. (**E**) Percentage breakdown of the noise, bias, and variance shown in C relative to the polynomial degrees. (**F**) Percentage breakdown for bias and variance.

Here, we show two different types of results. The first type, shown in Figure 3A,C,E,F, corresponds to numerical simulation results fitting a linear polynomial regression to training data, whereas the second type, shown in Figure 3B,D (emphasized by the dashed red rectangle), corresponds to idealized results that hold for general statistical models beyond our studied examples. The numerical simulation results in Figure 3A,C,E,F have been obtained by averaging over an ensemble of repeated model fits. For all these fits, the sample size of the training data was kept fixed.

The plots shown in Figure 3A,B are called **error-complexity curves**. They are important for evaluating the learning behavior of models.

Definition 1. Error-complexity curves *show the training error and test error in dependence on the model complexity. The models underlying these curves are estimated from training data with a fixed sample size.*

From Figure 3A, one can see that the training error decreases with an increasing polynomial degree, while in contrast, the test error is U-shaped. Intuitively, it is clear that more complex models fit the training data better, but there should be an optimal model complexity, and going beyond could worsen the prediction performance. The training error alone clearly does not reflect this, and for this reason, estimates of the test error are needed. Figure 3B shows idealized results for characteristic behavior of the training and test error for general statistical models.

In Figure 3C, we show the decomposition of the test error into its noise, bias, and variance components. The noise is constant for all polynomial degrees, whereas the bias is monotonously decreasing and the variance is increasing. Also, this behavior is generic beyond the shown examples. For this reason, we show in Figure 3D the idealized decomposition (neglecting the noise because of its constant contribution).

In Figure 3E, we show the percentage breakdown of the noise, bias, and variance for each polynomial degree. In this representation, the behavior of the noise is not constant because of the non-linear decomposition for different complexity values of the model. The numerical values of the percentage breakdown depend on the degree of the polynomial and can vary, as is evident from the Figure. Figure 3F shows the same as in Figure 3E, but without the noise part. From these representations, one can see that simple models have a high bias and a low variance, and complex models have a low bias and a high variance. This characterization is also generic and not limited to the particular model we studied.

4.3. Idealized Error-Complexity Curves

From the idealized error-complexity curves in Figure 3B, one can summarize and clarify a couple of important terms. We say **a model is overfitting** if its test error is higher than those of a *less complex* model. That means to decide whether a model is overfitting, it is necessary to compare it with a simpler model. Hence, overfitting is detected from a comparison, and it is not an absolute measure. Figure 3B shows that all models with a model complexity larger than 3.5 are overfitting, with respect to the best model having a model complexity of $c_{opt} = 3.5$ leading to the lowest test error. One can formalize this by defining an overfitting model as follows.

Definition 2 (model overfitting)**.** *A model with complexity c is called* **overfitting** *if, for the test error of this model, the following holds:*

$$E_{test}(c) - E_{test}(c_{opt}) > 0 \quad \forall c > c_{opt} \tag{50}$$

with

$$c_{opt} = \arg\min_{c} \left\{ E_{test}(c) \right\} \tag{51}$$

$$E_{test}(c_{opt}) = \min_{c} \left\{ E_{test}(c) \right\} \tag{52}$$

From Figure 3B we can also see that for all these models, the difference between the test error and the training error increases for increasing complexity values—that is,

$$\Big(E_{test}(c) - E_{train}(c) \Big) > \Big(E_{test}(c') - E_{train}(c') \Big) \quad \forall c > c' \text{ and } c, c' > c_{opt}. \tag{53}$$

Similarly, we say **a model is underfitting** if its test error is higher than those of a *more complex* model. In other words, to decide whether a model is underfitting, it is necessary to compare it

with a more complex model. In Figure 3B, all models with a model complexity smaller than 3.5 are underfitting, with respect to the best model. The formal definition of this can be given as follows.

Definition 3 (model underfitting). *A model with complexity c is called* **underfitting** *if, for the test error of this model, the following holds:*

$$E_{test}(c) - E_{test}(c_{opt}) > 0 \quad \forall c < c_{opt}. \tag{54}$$

Finally, the **generalization capabilities of a model** are assessed by its predictive performance of the test error in comparison with the training error. If the distance between the test error and the training error is small (has a small gap), such as

$$E_{test}(c) - E_{train}(c) \approx 0, \tag{55}$$

the model has good generalization capabilities [38]. From Figure 3B, one can see that models with $c > c_{opt}$ have bad generalization capabilities. In contrast, models with $c < c_{opt}$ have good generalization capabilities, but not necessarily small error. This makes sense considering the fact that the sample size is kept fixed.

In Definition 4 we formally summarize these characteristics.

Definition 4 (generalization). *If a model with complexity c holds*

$$E_{test}(c) - E_{train}(c) < \delta \text{ with } \delta \in \mathbb{R}^+, \tag{56}$$

we say the model has good generalization capabilities.

In practice, one needs to decide what a reasonable value of δ is, because $\delta = 0$ is usually too strict. This makes the definition of generalization problem specific. Put simply, if one can conclude from the training error to the test error (because they are of similar value), a model generalizes to new data.

Theoretically, for increasing the sample size of the training data, we obtain

$$\lim_{n_{train} \to \infty} E_{test}(c) - E_{train}(c) = 0 \tag{57}$$

for all model complexities c, because Equations (46) and (47) become identical, assuming an infinite large test data set—that is, $n_{test} \to \infty$.

From the idealized decomposition of the test error shown in Figure 3D, one can see that a simple model with low variance and high bias generally has good generalization capabilities, whereas for a complex model, its variance is high and the model's generalization capabilities are poor.

5. Model Selection

The expected generalization error provides the most complete information about the generalization abilities of a model. For this reason, the expected generalization error is used for model assessment [43–45]. It would appear natural to also perform model selection based on model assessment of the individual models. If it is possible to estimate the expected generalization error for each individual model, this is the best you can do. Unfortunately, it is not always feasible to estimate the expected generalization error, and for this reason, alternative approaches have been introduced. The underlying idea of these approaches is to estimate an auxiliary function that is different to the expected generalization error, but suffices to order different models in a similar way as could be done with the help of the expected generalization error. This means that the measure used for model selection just needs to result in the same ordering of models as if the generalization errors of the models would have been used for the ordering. Hence, model selection is actually a model ordering

problem, and the best model is selected without necessarily estimating the expected generalization error. This explains why model assessment and model selection are generally two different approaches.

There are two schools of thought in model selection, and they differ in the way in which one defines "best model". The first defines a best model as the "best prediction model", and the second as the "true model" that generated the data [21,22,46]. For this reason, the latter is referred to as *model identification*. The first definition fits seamlessly into our above discussion, whereas the second one is based on the assumption that the true model also has the best generalization error. For very large sample sizes ($n_{train} \to \infty$), this is uncontroversial; however, for finite sample sizes (as is the case in practice), this may not be the case.

In Figure 4, we visualize the general problem of model selection. In Figure 4A we show three model families indicated by the three curves in blue, red, and green. Each of these model families correspond to a statistical model—that is, a linear regression model with covariates of p_1, p_2, and p_3. Similarly to Figure 1A, each point along these lines correspond to a particular model obtained from estimating the parameters of the models from a data set. These parameter estimates are obtained by using a training data set. Here, $\hat{y}_1 = g_1(x_1, \hat{\beta}_1(D))$, $\hat{y}_2 = g_2(x_2, \hat{\beta}_2(D))$, and $\hat{y}_3 = g_3(x_3, \hat{\beta}_3(D))$ are three examples.

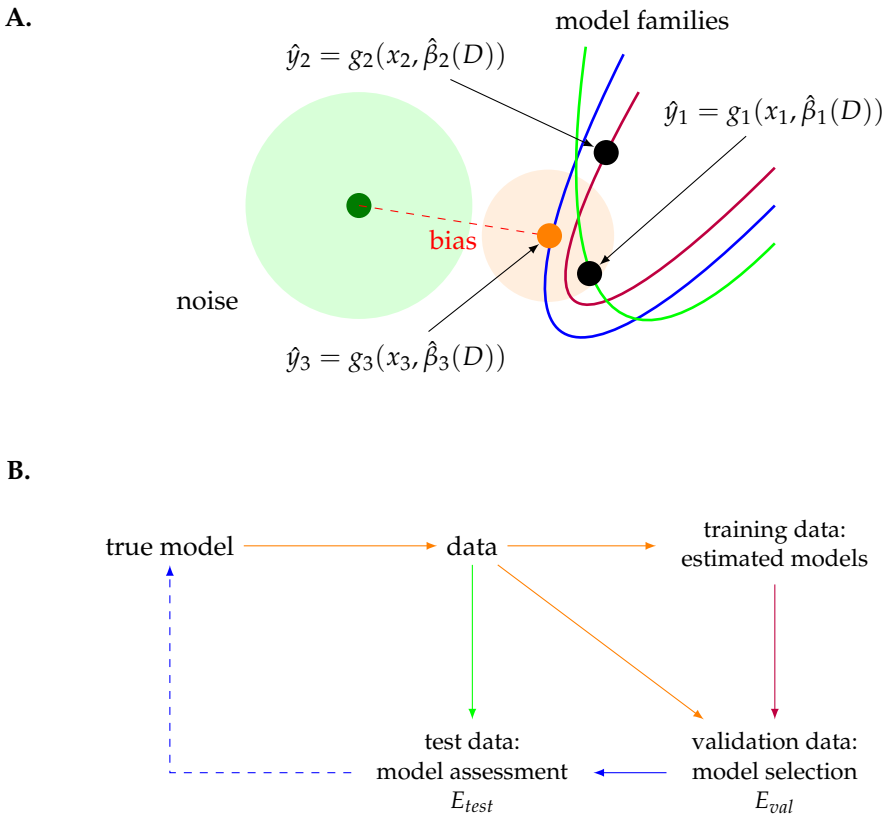

Figure 4. Idealized visualization of the model selection process. (**A**) Three model families are shown, and estimates of three specific models were obtained from training data. (**B**) A summary combining model selection and model assessment, emphasizing that different data sets are used for different analysis steps.

After the parameters of the three models have been estimated, one performs a model selection for identifying the best model according to a criterion. For this, a validation data set is used. Finally, one performs a model assessment of the best model by using a test data set.

In Figure 4B, a summary of the above process is shown. Here, we emphasize that different data (training data, validation data, or test data) are used for the corresponding analysis step. Assuming an ideal (very large) data set D, there are no problems with the practical realization of this step. However,

practically, we have no ideal data set, but one with a finite sample size. This problem will be discussed in detail in Section 6.

In the following, we discuss various evaluation criteria for model selection that can be used for model ranking.

5.1. R^2 and Adjusted R^2

The first measure we discuss is called the *coefficient of determination* (COD) [47,48]. The COD is defined as

$$R^2 = \frac{\text{SSR}}{\text{SST}} = 1 - \frac{\text{SSE}}{\text{SST}}. \tag{58}$$

This definition is based on SSR and SST in Equations (17) and (19). The COD is a measure of how well the model explains the variance of the response variables. A disadvantage of R^2 is that a submodel of a full model always has a smaller value, regardless of its quality.

For this reason, a modified version of R^2 has been introduced, called the *adjusted coefficient of determination* (ACOD). The ACOD is defined as

$$R^2_{adj} = 1 - \frac{\text{SSE}(n-1)}{\text{SST}(n-p)}. \tag{59}$$

It can also be written in dependence on R^2, as

$$R^2_{adj} = 1 - \frac{(n-1)}{(n-p-1)}(1-R^2). \tag{60}$$

The ACOD adjusts for sample size n of the training data and mode complexity, as measured by the number of covariates, p.

5.2. Mallows' Cp Statistic

For a general model and in-sample data $\{(x_i, y_i)\}$ used for training and out-sample data $\{(x_i, y'_i)\}$ used for testing, one can show that

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2\right] < \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(y'_i - \hat{y}_i\right)^2\right]. \tag{61}$$

Furthermore, if the model is linear having p predictors and an intercept one can show that

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(y'_i - \hat{y}_i\right)^2\right] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2\right] + \underbrace{\frac{2}{n}\sigma^2(p+1)}_{\text{optimism}}. \tag{62}$$

The last term in Equation (62) is called *optimism*, because it is the amount by which the in-sample error underestimates the out-sample error. Hence, a large value of the optimism indicates a large discrepancy between both errors. It is interesting to note that:

1. The optimism increases with σ^2;
2. The optimism increases with p;
3. The optimism decreases with n.

Explanations for the above factors are given by:

1. Adding more noise (indicated by increasing σ^2) and leaving n and p fixed makes it harder for a model to be learned;

2. Increasing the complexity of the model (indicated by increasing p) and leaving σ^2 and n fixed makes it easier for a model to fit the test data but is prune to overfitting;
3. Increasing the test data set (indicated by increasing n) and leaving σ^2 and p fixed reduces the chances for overfitting.

The problem with Equation (62) is that σ^2 corresponds to the true value of the noise which is unknown. For this reason, one needs to use an estimator to obtain a reasonable approximation. One can show that by estimating $\hat{\sigma}^2$ from the largest model, this will be an unbiased estimator of σ^2 if the true model is smaller.

Using this estimate for σ^2 leads to Mallows' Cp statistic [49,50],

$$Cp = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2\right] + \frac{2}{n}\hat{\sigma}^2(p+1). \tag{63}$$

Alternatively, we can write Equation (63) as:

$$Cp = \text{MSE} + \frac{2}{n}\hat{\sigma}^2(p+1). \tag{64}$$

For model selection, one needs to choose the model that minimizes Cp. Mallows' Cp is only used for linear regression models that are evaluated with the squared error.

5.3. Akaike's Information Criterion (AIC), Schwarz's BIC, and the Bayes Factor

The next two model selection criteria are similar to Equation (64). Specifically, Akaike's information criterion (AIC) [24,51,52] for model $\mathcal{M}$ is defined by

$$\text{AIC}(\mathcal{M}) = -2\log\left(L_{\mathcal{M}}\right) + 2dim(\mathcal{M}). \tag{65}$$

Here, $L_{\mathcal{M}}$ is the likelihood of model $\mathcal{M}$ evaluated at the maximum likelihood estimate, and $dim(\mathcal{M})$ is the dimension of the model corresponding to the number of free parameters. In contrast to Mallows' Cp, the Akaike's information criterion selects the model that maximizes $\text{AIC}(\mathcal{M})$.

For a linear model, one can show that the log likelihood is given by

$$\log\left(L_{\mathcal{M}}\right) = -\frac{n}{2}\log\left(\text{MSE}\right) + C' \tag{66}$$

where C' is a model independent constant, and the dimension of the model is

$$dim(\mathcal{M}) = p + 2. \tag{67}$$

Taken together, this gives

$$\text{AIC}(\mathcal{M}) = n\log\left(\text{MSE}\right) + 2p + C \tag{68}$$

with $C = -2C' + 4$. For model comparisons, the parameter C is irrelevant.

The BIC (Bayesian Information Criterion) [53,54], also called the Schwarz criterion, has a similar form as the AIC. The BIC is defined by

$$\text{BIC}(\mathcal{M}) = -2\log\left(L_{\mathcal{M}}\right) + p\log\left(n\right). \tag{69}$$

For a linear model with normal distributed errors, this simplifies to

$$\text{BIC}(\mathcal{M}) = n\log\left(\text{MSE}\right) + p\log\left(n\right). \tag{70}$$

Also, BIC selects the model that maximizes $\text{BIC}(\mathcal{M})$.

Another model selection criterion is the Bayes' factor [55–58]. Suppose we have a finite set of models $\{\mathcal{M}_i\}$ with $i \in 1 \dots M$, which we can use for fitting the data D. In order to select the best model from a Bayesian perspective, we need to evaluate the posterior probability of each model,

$$Pr(\mathcal{M}_i|D), \tag{71}$$

for the available data. Using Bayes' theorem, one can write this probability as:

$$Pr(\mathcal{M}_i|D) = \frac{p(D|\mathcal{M}_i)Pr(\mathcal{M}_i)}{\sum_{j=1}^{M} p(D|\mathcal{M}_i)Pr(\mathcal{M}_j)}. \tag{72}$$

Here, the term $p(D|\mathcal{M}_i)$ is called the **evidence for the model** $\mathcal{M}_i$, or simply "evidence".

The ratio of the posterior probabilities for model $\mathcal{M}_A$ and $\mathcal{M}_B$ corresponding to the **posterior odds** of the models is given by:

$$\frac{Pr(\mathcal{M}_A|D)}{Pr(\mathcal{M}_B|D)} = \frac{p(D|\mathcal{M}_A)Pr(\mathcal{M}_A)}{p(D|\mathcal{M}_B)Pr(\mathcal{M}_B)} = \frac{p(D|\mathcal{M}_A)}{p(D|\mathcal{M}_B)} \times \text{prior odds} = BF_{AB} \times \text{prior odds}. \tag{73}$$

That means the Bayes' factor of the models is the ratio of the posterior probabilities and the prior probabilities.

$$BF_{AB} = \frac{\text{posterior odds}}{\text{prior odds}} = \frac{\frac{Pr(\mathcal{M}_A|D)}{Pr(\mathcal{M}_B|D)}}{\frac{Pr(\mathcal{M}_A)}{Pr(\mathcal{M}_B)}}. \tag{74}$$

If one uses non-informative priors, such as $Pr(\mathcal{M}_A) = Pr(\mathcal{M}_A) = 0.5$, then the Bayes' factor simplifies to

$$BF_{AB} = \text{posterior odds} = \frac{Pr(\mathcal{M}_A|D)}{Pr(\mathcal{M}_B|D)}. \tag{75}$$

Assuming the parameter dependency of a model $\mathcal{M}_i$ on θ, then the evidence can be written as

$$p(D|\mathcal{M}_i) = \int p(D|\theta, \mathcal{M}_i)p(\theta|\mathcal{M}_i)d\theta. \tag{76}$$

A serious problem with this expression is that it can be very hard to evaluate—especially in high dimensions—if no closed-form solution is available. This makes the Bayes' factor problematic to apply.

Interestingly, there is a close connection between the BIC and the Bayes' factor. Specifically, in [55] it has been proven that for $n \to \infty$, the following holds:

$$2 \ln BF_{BA} \approx \text{BIC}_A - \text{BIC}_B. \tag{77}$$

Note that the relation $-\ln BF_{AB} = \ln BF_{BA}$ is a negative symmetric. Hence, model comparison results for the BIC and the Bayes' factor can approximate each other.

For a practical application for interpreting the BIC and Bayes factors, [26] suggested the following evaluation of a comparison of two models—see Table 1. Here, "min" indicates the model with smaller BIC or posterior probability.

The common idea of AIC and BIC is to penalize larger models. Because $log(n = 8) > 2$, the BIC penalizes more harshly than AIC (usually, data sets have more than 8 samples). Hence, BIC selects smaller models than AIC. BIC has a consistency property which means that when the true unknown model is one of the models under consideration and, for the sample size, holds $n \to \infty$, BIC selects the correct model. In contrast, AIC does not have this consistency property.

Table 1. Interpretation of model comparisons with the BIC and Bayes' factor.

Evidence	Δ BIC $=$ BIC$_k$ $-$ BIC$_{min}$	BF$_{min,k}$
weak	0–2	1–3
positive	2–6	3–20
strong	6–10	20–150
very strong	>10	>150

In general, AIC and BIC are considered to have a different view on model selection [28]. Whereas BIC assumes that the true model is among the studied ones, its goal is to identify the true model. In contrast, AIC does not assume this; instead, the goal of AIC is to find the model that maximizes predictive accuracy. In practice, the true model is rarely among the model families studied, and for this reason the BIC cannot select the true model. For such a case, AIC is the appropriate approach for finding the best approximating model. Several studies suggest to prefer the AIC over BIC for practical applications [24,28,54]. For instance, in [59] it was found that AIC can select a better model than BIC even for the case when the true model is among the studied models. Specifically for regression models, in [60] it has been demonstrated that AIC is asymptotically efficient, selecting the model with the least MSE while when the true model is not among the studied models, BIC does not.

In summary, the AIC and BIC have the following characteristics:

- BIC selects smaller models (more parsimonious) than AIC and tends to perform underfitting;
- AIC selects larger models than BIC and tends to perform overfitting;
- AIC represents a frequentist point of view;
- BIC represents a Bayesian point of view;
- AIC is asymptotically efficient but not consistent;
- BIC is consistent but not asymptotically efficient;
- AIC should be used when the goal is prediction accuracy of a model;
- BIC should be used when the goal is model interpretability.

The AIC and BIC are generic in their applications not limited to linear models, and can be applied whenever we have a likelihood of a model [61].

5.4. Best Subset Selection

So far, we discussed evaluation criteria which one can use for model selection. However, we did not discuss how these criteria are actually used. In the following, we provide this information, discussing best subset selection (Algorithm 1), forward stepwise selection (Algorithm 2), and backward stepwise selection (Algorithm 3) [47,62,63]. All of these approaches are computational.

Algorithm 1: Best subset selection.

Input: A model family $\mathcal{M}$ to be fitted.

Let $\hat{\mathcal{M}}_0$ denote the fitted model with zero parameters.

for $k = 1, \ldots, p$ **do**

  Fit all $\binom{p}{k}$ models having k parameters.

  Select the best model with k parameters and call it $\hat{\mathcal{M}}_k$. The evaluation of this is based on minimizing the MSE or on maximizing R^2.

Select the best model from $\{\hat{\mathcal{M}}_0, \ldots, \hat{\mathcal{M}}_p\}$ by using C_p, AIC, or BIC as evaluation criterion, or cross-validation.

Algorithm 2: Forward stepwise selection.

Input: A model family $\mathcal{M}$ to be fitted.

1 Let $\hat{M}_0$ denote the fitted model with zero parameters.
2 **for** $k = 0, \ldots, p-1$ **do**
3 Fit all $p-k$ models having $k+1$ parameters.
4 Select the best of these $p-k$ models and call it $\hat{\mathcal{M}}_{k+1}$. The evaluation of this is based on minimizing the MSE, or on maximizing R^2, or cross-validation.
5 Select the best model from $\{\hat{\mathcal{M}}_0, \ldots, \hat{\mathcal{M}}_p\}$ by using C_p, AIC, or BIC as evaluation criterion.

Algorithm 3: Backward stepwise selection.

Input: A model M to be fitted.

1 Let $\hat{M}_p$ denote the fitted model with p parameters.
2 **for** $k = p, \ldots, 1$ **do**
3 Fit all k models having $k-1$ parameters from the parameters of model $\hat{M}_k$.
4 Select the best of these k models and call it $\hat{M}_{k-1}$. The evaluation of this is based on minimizing the MSE or on maximizing R^2.
5 Select the best model from $\{\hat{M}_0, \ldots, \hat{M}_p\}$ by using C_p, AIC, or BIC as evaluation criterion.

The most brute-force model selection strategy is evaluating each possible model. This is the idea of best subset selection (Best).

Best subset selection evaluates each model with k parameters by the MSE or R^2. Due to the fact that each of these models have the same complexity (a model with k parameters), measures considering the model complexity are not needed. However, when comparing the $p+1$, different models having different parameters (see line 5 in Algorithm 1) a complexity penalizing measure, such as the C_p, AIC, or BIC, needs to be used.

For a linear regression model, one needs to fit all combinations with p predictors. A problem with the best subset selection is that in total, one needs to evaluate $\sum_{k=0}^{p} \binom{p}{k} = 2^p$ different models. For $p = 20$, this already gives over 10^6 models, leading to computational problems in practice. For this reason, approximations to the best subset selection are needed.

5.5. Stepwise Selection

Two such approximations are discussed in the following. Both of these follow a greedy approach, whereas forward stepwise selection does this in a bottom-up manner, and backward stepwise selection does it in a top-down manner.

5.5.1. Forward Stepwise Selection

The idea of forward stepwise selection (FSS) is to start with a null model without parameters and successively add one parameter at a time, that is best done according to a selection criterion.

For a linear regression model with p predictors, this gives

$$1 + \sum_{k=0}^{p-1} (p-k) = 1 + \frac{p(p+1)}{2} \tag{78}$$

models. For $p = 20$, this gives only 211 different models one needs to evaluate, which is a great improvement compared to best subset selection.

5.5.2. Backward Stepwise Selection

The idea of backward stepwise selection (BSS) is to start with a full model with p parameters and successively remove one parameter at a time that is worst according to a selection criterion.

The number of models that need to be evaluated with backward stepwise selection is the exact same as for forward stepwise selection.

Both stepwise selection strategies are not guaranteed to find the best model containing a subset of the p predictors. However, when p is large, both approaches may be the only ones which are practically feasible. Despite the apparent symmetry of the forward stepwise selection and the backward stepwise selection, there is a difference in situations when $p > n$, or when we have more parameters than samples in our data. In this case, the forward stepwise selection approach can still be applied because the procedure may be systematically limited to n parameters.

6. Cross-Validation

A cross-validation (CV) approach is the most practical and flexible approach one can use for model selection [23,64,65]. The reasons for this are because (A) it is conceptually simple, (B) it is intuitive, and (C) it can be applied to any statistical model family regardless of its technical details (for instance, to parametric and non-parametric models). Conceptually, cross-validation is a resampling method [66–68] and its basic idea is to repeatedly split the data into training and validation data for estimating the parameters of the model and for its evaluation—see Figure 5 for a visualization of the base functioning of a five-fold cross-validation. Importantly, the test data used for model assessment (MA) are not resampled during this process.

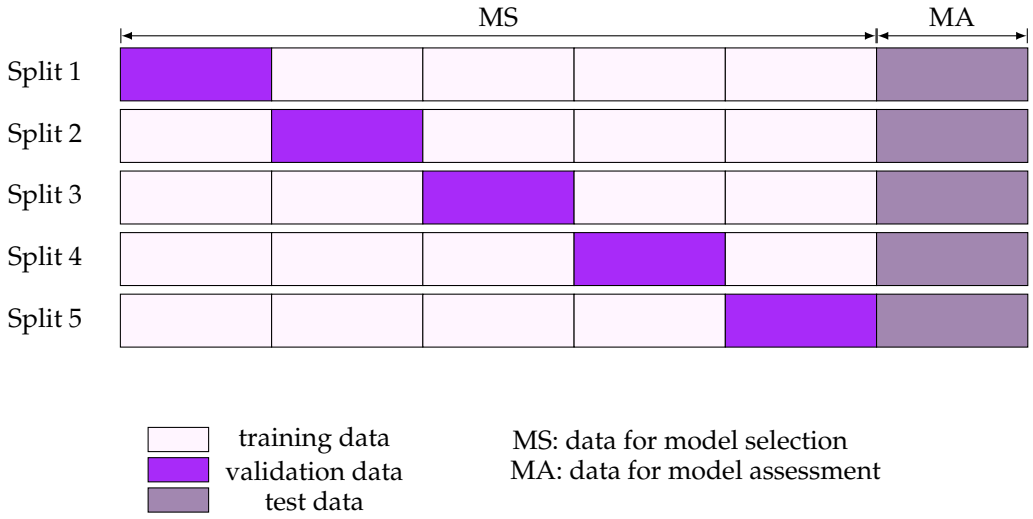

Figure 5. Visualization of cross-validation. Before splitting the data, the data points are randomized, but then kept fixed for all splits. Neglecting the column MA shows a standard five-fold cross validation. Consideration of the column MA shows a five-fold cross validation with holding-out of a test set.

Formally, cross-validation works the following way. For each split k ($k \in \{1, \ldots, K\}$), the parameters of model m ($m \in \{1, \ldots, M\}$) are estimated using the training data, and the prediction error is evaluated using the validation data—that is:

$$E_{val}(k, m) = \frac{1}{n_{val}} \sum_{i=1}^{n_{val}} L(y_i, \hat{g}_m(x_i, D_{train})) \tag{79}$$

After the last split, the errors are summarized by

$$E_{val}(m) = \frac{1}{K} \sum_{k=1}^{K} E_{val}(k, m). \tag{80}$$

This gives estimates of the prediction error for each model m. The best model can now be selected by

$$m_{opt} = \arg\min_{m} \left\{ E_{val}(m) \right\}. \tag{81}$$

Compared to other approaches for model selection, cross-validation has the following advantages:

- Cross-validation is a computational method that is simple in its realization;
- Cross-validation makes few assumptions about the true underlying model;
- Compared with AIC, BIC, and the adjusted R^2, cross-validation provides a direct estimate of the prediction error;
- Every data point is used for both training and testing.

Some drawbacks of cross-validation are:

- The computation time can be long because the whole analysis needs to be repeated K times for each model;
- The number of folds (K) needs to be determined;
- For a small number of folds, the bias of the estimator will be large.

There are many technical variations of cross-validation and other resampling methods (e.g., Boostrap [69,70]) to improve the estimates [23,71,72]. We just want to mention that in the case of very limited data, *leave-one-out cross-validation* (LOOCV) has some advantages [72]. In contrast to cross-validation, LOOCV splits the data into $K = n$ folds, whereas n corresponds to the total number of samples. The rest of the analysis proceeds like CV.

Using the same idea as for model selection, cross-validation can also be used for model assessment. In this case, the prediction error is estimated by using the test data, instead of the validation data used for model selection—see Figure 5. That means we estimate the prediction error for each split by

$$E_{test}(k, m_{opt}) = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} L(y_i, \hat{g}_{m_{opt}}(x_i, D_{train})) \tag{82}$$

and summarize these errors by the sample average

$$E_{test}(m_{opt}) = \frac{1}{K} \sum_{k=1}^{K} E_{test}(k, m_{opt}). \tag{83}$$

7. Learning Curves

Finally, we discuss learning curves as another way of model diagnosis. A learning curve shows the performance of a model for different sample sizes of the training data [73,74]. The performance of a model is measured by its prediction error. For extracting the most information, one needs to compare the learning curve of the training error and the test error with each other. This leads to complementary information to the error-complexity curves. Hence, learning curves are playing an important role in model diagnosis, but are not strictly considered as part of model assessment methods.

Definition 5. Learning curves *show the training error and test error in dependence on the sample size of the training data. The models underlying these curves all have the same complexity.*

In the following, we first present numerical examples for learning curves for linear polynomial regression models. Then, we discuss the behavior of idealized learning curves that can correspond to any type of statistical model.

7.1. Learning Curves for Linear Polynomial Regression Models

In Figure 6, we show results for the linear polynomial regression models discussed earlier. It is important to emphasize that each figure shows results for a fixed model complexity, but varying sample sizes of the training data. This is in contrast to the results shown earlier (see Figure 3) which varied the model complexity but kept the sample size of the training data fixed. We show six examples for six different model degrees. The horizontal red dashed line corresponds to the optimal error $E_{test}(c_{opt})$ attainable by the model family. The first two examples (Figure 6A,B) are qualitatively different to all others because neither the training nor the test error converge to $E_{test}(c_{opt})$, yet are much higher. This is due to a high bias of the models, because these models are too simple for the data.

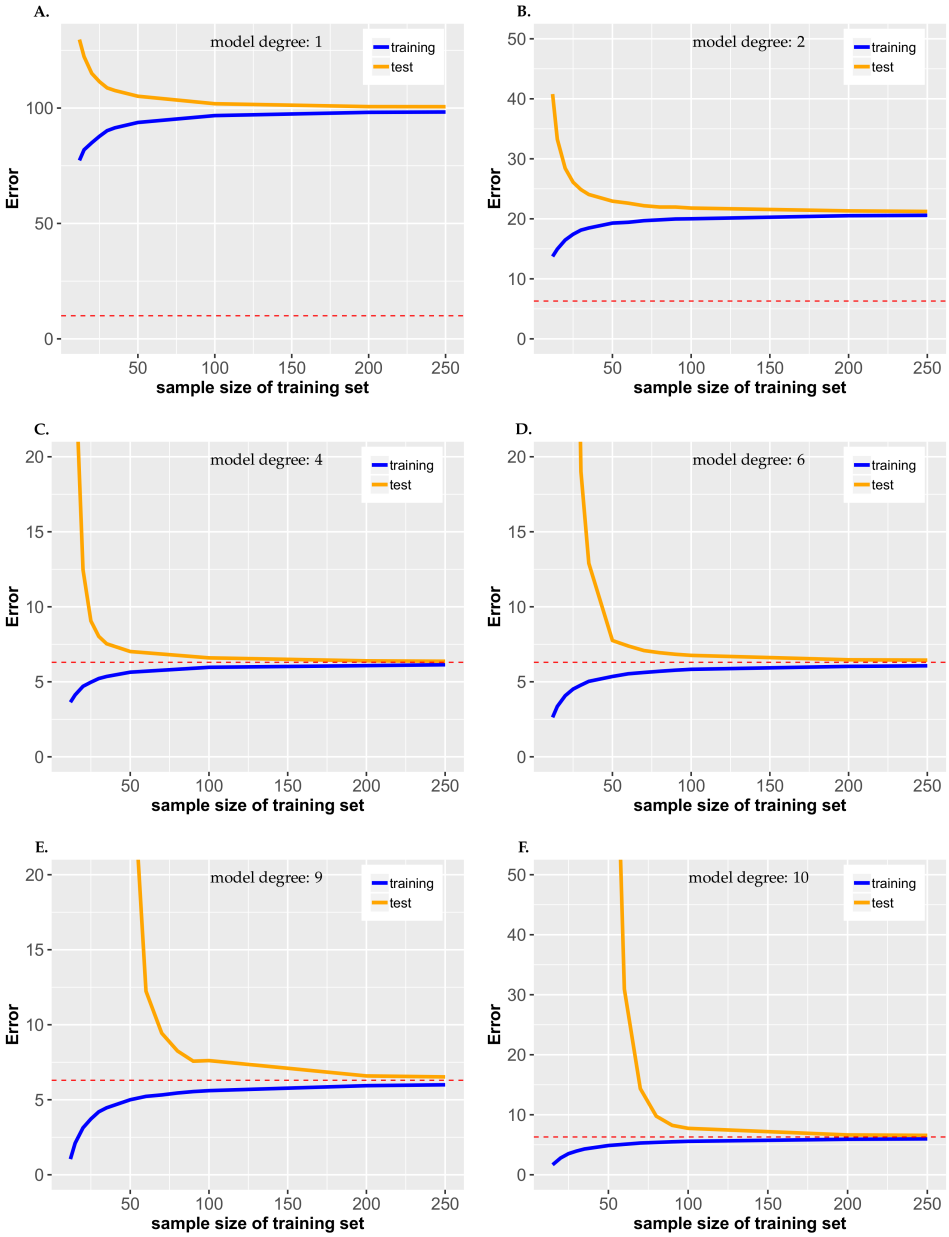

Figure 6. Estimated learning curves for training and test errors for six linear polynomial regression models. The model degree indicates the highest polynomial degree of the fitted model, and the horizontal dashed red line corresponds to the optimal error $E_{test}(c_{opt})$ attainable by the model family for the optimal model complexity $c_{opt} = 4$.

Figure 6E exhibits some different extreme behavior. Here, for sample sizes of the training data smaller than ≈ 60, one can obtain very high test errors and a large difference to the training error. This is due to a high variance of the models, because those models are too complex for the data. In contrast, Figure 6C shows results for $c_{opt} = 4$ which are the best results obtainable for this model family and the data.

In general, learning curves can be used to answer the following two questions:

1. How much training data is needed?
2. How much bias and variance is present?

For (1): The learning curves can be used to predict the benefits one can obtain from increasing the number of samples in the training data.

- If the curve is still changing (increasing for training error and decreasing for test error) rapidly $\rightarrow$ need larger sample size;
- If the curve is completely flattened out $\rightarrow$ sample size is sufficient;
- If the curve is gradually changing $\rightarrow$ a much larger sample size is needed.

This assessment is based on evaluating the tangent of a learning curve toward the highest available sample size.

For (2): In order to study this point, one needs to generate several learning curves for models of different complexity. From this, one obtains information about the smallest attainable test error. In the following, we call this the optimal attainable error $E_{test}(c_{opt})$.

For a specific model, one can evaluate its learning curves as follows.

- A model has **high bias** if the training and test error converge to a value much larger than E_{test}. In this case, increasing the sample size of the training data will not improve the results. This indicates an underfitting of the data because the model is too simple. In order to improve this, one needs to increase the complexity of the model.
- A model has **high variance** if the training and test error are quite different from each other, with a large gap between both. Here, a gap is defined as $E_{test}(n) - E_{train}(n)$ for sample size n of the training data. In this case, the training data are fitted much better than the test data, indicating problems with the generalization capabilities of the model. In order to improve the sample size of the training data, needs to be increased.

These assessments are based on evaluating the gap between the test error and the training error toward the highest available sample size of the training data.

7.2. Idealized Learning Curves

In Figure 7, we show idealized learning curves for the four cases one obtains from combining high/low bias and high/low variance with each other. Specifically, the first/second column shows low/high bias cases, and the first/second row shows low/high variance cases. Figure 7A shows the ideal case when the model has a low bias and a low variance. In this case, the training and test error both converge to the optimal attainable error $E_{test}(c_{opt})$ that is shown as a dashed red line.

In Figure 7B, a model with a high bias and a low variance is shown. In this case, the training and test error both converge to values that are distinct from the optimal attainable error, and an increase in the sample size of the training data will not solve this problem. The small gap between the training and test error is indicative of a low variance. A way to improve the performance is to increase the model complexity, such as by allowing more free parameters or boosting approaches. This case is the ideal example for an **underfitting model**.

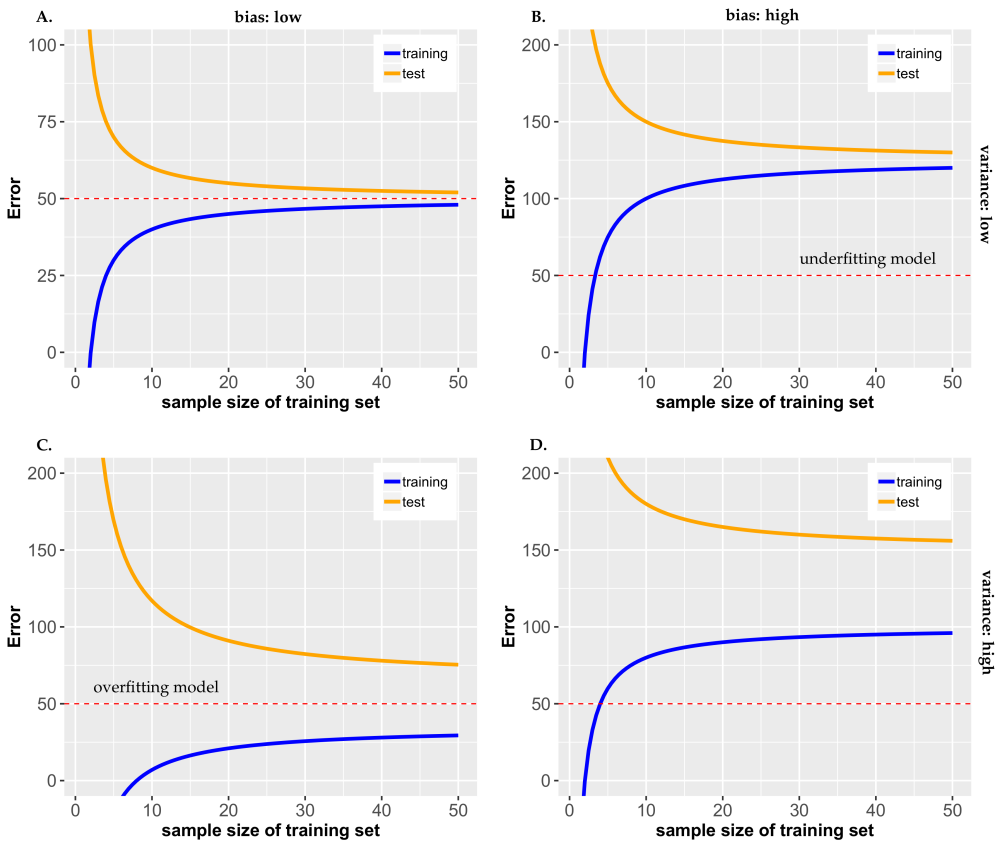

Figure 7. Idealized learning curves. The horizontal red dashed line corresponds to the optimal attainable error $E_{test}(c_{opt})$ by the model family. Shown are the following four cases. (**A**) Low bias, low variance; (**B**) high bias, low variance; (**C**) low bias, high variance; (**D**) high bias, high variance.

In Figure 7C, a model with a low bias and a high variance is shown. In this case, the training and test error both converge to the optimal attainable error. However, the gap between the training and test error is large, indicating a high variance. In order to reduce this variance, the sample size of the training data needs to be increased to possibly much larger values. Also, the model complexity can be reduced, such as by regularization or bagging approaches. This case is the ideal example for an **overfitting model**.

In Figure 7D, a model with a high bias and a high variance is shown. This is the worst-case scenario. In order to improve the performance, one needs to increase the model complexity and possibly the sample size of the training data. This means improving such a model is the most demanding case.

Also, the learning curves allow an evaluation of the generalization capabilities of a model. Only the low variance cases have a small distance between the test error and the training error, indicating the model has good generalization capabilities. Hence, a model with low variance generally has good generalization capabilities, irrespective of the bias. However, models with a high bias perform badly, and may only be considered in exceptional situations.

8. Summary

In this paper, we presented theoretical and practical aspects of model selection, model assessment, and model diagnosis [75–77]. The error-complexity curves, the bias–variance tradeoff, and the learning curves provide means for a theoretical understanding of the core concepts. In order to utilize error-complexity curves and learning curves for a practical analysis, cross-validation offers a flexible approach to estimate the involved entities for general statistical models which are not limited to linear models.

In practical terms, model selection is the task of selecting the best statistical model from a model family, given a data set. Possible model selection problems include, but are not limited to:

- Selecting predictor variables for linear regression models;
- Selecting among different regularization models, such as ridge regression, LASSO, or elastic net;
- Selecting the best classification method from a list of candidates, such as random forest, logistic regression, or the support vector machine of neural networks;
- Selecting the number of neurons and hidden layers in neural networks.

The general problems one tries to counteract with model selection are overfitting and underfitting of data.

- An underfitting model: Such a model is characterized by high bias, low variance, and poor test error. In general, such a model is too simple;
- The best model: For such a model, the bias and variance are balanced and the test error makes good predictions;
- An overfitting model: Such a model is characterized by low bias, high variance, and poor test error. In general, such a model is too complex.

It is important to realize that these terms are defined for a given data set with a certain sample size. Specifically, the error-complexity curves are estimated from training data with a fixed sample size and, hence, these curves can change if the sample size changes. In contrast, the learning curves investigate the dependency on the sample size of the training data.

We also discussed more elegant methods for model selection, such as AIC or BIC; however, the applicability of these depends on the availability of the analytical results of models, such as about their maximum likelihood. Such results can usually be obtained for linear models, as discussed in our paper, but may not be known for more complex models. Hence, for practical applications, these methods are far less flexible than cross-validation.

The bias-variance tradeoff providing a frequentist view-point of model complexity is for practical problems, for which the true model is unknown, not accessible. Instead, it offers a conceptual framework to think about a problem theoretically. Interestingly, the balancing of bias and variance reflects the underlying philosophy of Ockham's razor [78], stating that from two similar models, the simpler one should be chosen. On the other hand, for simulations, the true model is known and the decomposition into noise, bias, and variance is feasible.

In Figure 8 we summarize different model selection approaches. In this figure, we highlight two important characteristics of such methods. The first characteristic distinguishes methods regarding data-splitting, and the second regarding model complexity. Neither best subset selection (Best), forward stepwise selection (FSS), nor backward stepwise selection (BSS) apply data-splitting, but they use the entire data for evaluation. Furthermore, each of these approaches is a two-step procedure that employs, in its first step, a measure that does not consider the model complexity. For instance, either the MSE or R^2 is used in this step. In the second step, a measure considering model complexity is used, such as AIC, BIC, or C_p.

Another class of model selection approaches uses data-splitting. Data-splitting is typically based on resampling of the data, and in this paper we focused on cross-validation. Interestingly, CV can be used without (MSE) or with (regularization) model complexity measures. Regularized regression models, such as ridge regression, LASSO, or elastic net, consider the complexity by varying the value of λ (regularization parameter).

In practice, the most flexible approach that can be applied to any type of statistical model is cross-validation. Assuming the computations can be completed within an acceptable time frame, it is advised to base the decisions for model selection and model assessment on the estimates of the error-complexity curves and the learning curves. Depending on the data and the model family, there can be technical issues which may require the application of other resampling methods in order

to improve the quality of the estimates. However, it is important to emphasize that all of these issues are purely of numerical nature, not conceptual.

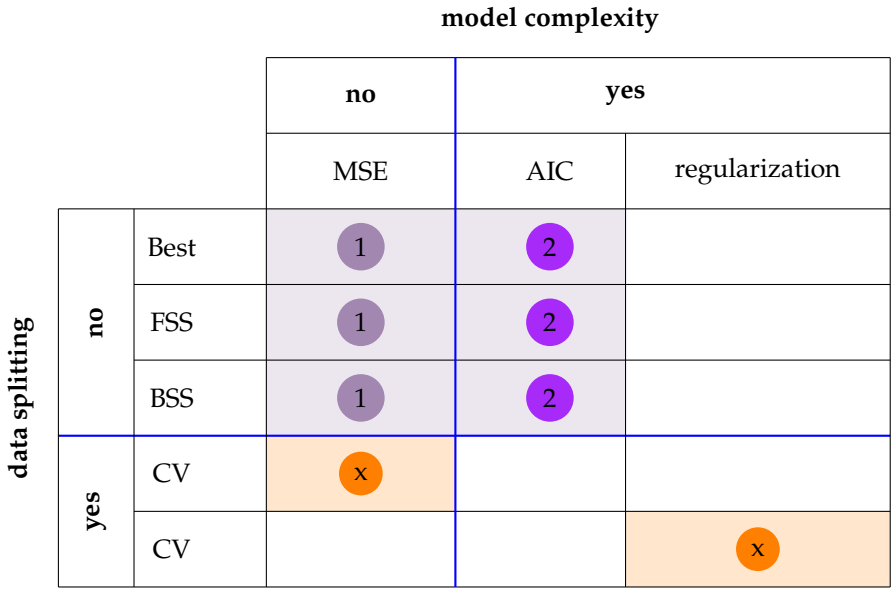

Figure 8. Summary of different model selection approaches. Here, AIC stands for any criterion considering model complexity, such as BIC or C_p, and regularization is any regularized regression model, such as LASSO or elastic net.

In summary, cross-validation, AIC, and C_p all have the same goal—trying to find a model that predicts best. They all tend to choose similar models. On the other hand, BIC is quite different, and tends to choose smaller models. Also, its goal is different because it tries to identify the true model. In general, smaller models are easier to interpret, and obtain an understanding of the underlying process. Overall, cross-validation is the most general approach and can be used for parametric, as well as non-parametric models.

9. Conclusions

Data science is currently receiving much attention across various fields because of the big data-wave which is flooding all areas of science and our society [79–83]. Model selection and model assessment are two important concepts when studying statistical inference, and every data scientist needs to be familiar with this in order to select the best model and to assess its prediction capabilities fairly in terms of the generalization error. Despite the importance of these topics, there is a remarkable lack of accessible reviews on the intermediate level in the literature. Given the interdisciplinary character of data science, this level is particularly needed for scientists interested in applications. We aimed to fill this gap with a particular focus on the clarity of the underlying theoretical framework and its practical realizations.

Author Contributions: F.E.-S. conceived the study. All authors contributed to the writing of the manuscript and approved the final version.

Funding: M.D. thanks the Austrian Science Funds for supporting this work (project P30031).

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Chang, R.M.; Kauffman, R.J.; Kwon, Y. Understanding the paradigm shift to computational social science in the presence of big data. *Decis. Support Syst.* **2014**, *63*, 67–80. [CrossRef]

2. Provost, F.; Fawcett, T. Data science and its relationship to big data and data-driven decision making. *Big Data* **2013**, *1*, 51–59. [CrossRef]
3. Hardin, J.; Hoerl, R.; Horton, N.J.; Nolan, D.; Baumer, B.; Hall-Holt, O.; Murrell, P.; Peng, R.; Roback, P.; Lang, D.T.; et al. Data science in statistics curricula: Preparing students to 'think with data'. *Am. Stat.* **2015**, *69*, 343–353. [CrossRef]
4. Emmert-Streib, F.; Moutari, S.; Dehmer, M. The process of analyzing data is the emergent feature of data science. *Front. Genet.* **2016**, *7*, 12. [CrossRef] [PubMed]
5. Emmert-Streib, F.; Dehmer, M. Defining data science by a data-driven quantification of the community. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 235–251. [CrossRef]
6. Dehmer, M.; Emmert-Streib, F. *Frontiers Data Science*; CRC Press: Boca Raton, FL, USA, 2017.
7. Ansorge, W. Next-generation DNA sequencing techniques. *New Biotechnol.* **2009**, *25*, 195–203. [CrossRef]
8. Emmert-Streib, F.; de Matos Simoes, R.; Mullan, P.; Haibe-Kains, B.; Dehmer, M. The gene regulatory network for breast cancer: Integrated regulatory landscape of cancer hallmarks. *Front. Genet.* **2014**, *5*, 15. [CrossRef]
9. Musa, A.; Ghoraie, L.; Zhang, S.D.; Glazko, G.; Yli-Harja, O.; Dehmer, M.; Haibe-Kains, B.; Emmert-Streib, F. A review of connectivity mapping and computational approaches in pharmacogenomics. *Brief. Bioinf.* **2017**, *19*, 506–523.
10. Mardis, E.R. Next-generation DNA sequencing methods. *Ann. Rev. Genom. Hum. Genet.* **2008**, *9*, 387–402. [CrossRef]
11. Tripathi, S.; Moutari, S.; Dehmer, M.; Emmert-Streib, F. Comparison of module detection algorithms in protein networks and investigation of the biological meaning of predicted modules. *BMC Bioinf.* **2016**, *17*, 1–18. [CrossRef] [PubMed]
12. Conte, R.; Gilbert, N.; Bonelli, G.; Cioffi-Revilla, C.; Deffuant, G.; Kertesz, J.; Loreto, V.; Moat, S.; Nadal, J.P.; Sanchez, A.; et al. Manifesto of computational social science. *Eur. Phys. J.-Spec. Top.* **2012**, *214*, 325–346. [CrossRef]
13. Lazer, D.; Pentland, A.S.; Adamic, L.; Aral, S.; Barabasi, A.L.; Brewer, D.; Christakis, N.; Contractor, N.; Fowler, J.; Gutmann, M.; et al. Life in the network: The coming age of computational social science. *Science* **2009**, *323*, 721. [CrossRef] [PubMed]
14. Emmert-Streib, F.; Yli-Harja, O.; Dehmer, M. Data analytics applications for streaming data from social media: What to predict? *Front. Big Data* **2018**, *1*, 1. [CrossRef]
15. Breiman, L. Bagging Predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
16. Clarke, B.; Fokoue, E.; Zhang, H.H. *Principles and Theory for Data Mining and Machine Learning*; Springer: Dordrecht, The Netherlands; New York, NY, USA, 2009.
17. Harrell, F.E. *Regression Modeling Strategies*; Springer: New York, NY USA, 2001.
18. Haste, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*; Springer: New York, NY, USA, 2009.
19. Emmert-Streib, F.; Dehmer, M. High-dimensional LASSO-based computational regression models: Regularization, shrinkage, and selection. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 359–383. [CrossRef]
20. Schölkopf, B.; Smola, A. *Learning with Kernels: Support Vector Machines, Regulariztion, Optimization and Beyond*; The MIT Press: Cambridge, MA, USA, 2002.
21. Ding, J.; Tarokh, V.; Yang, Y. Model selection techniques: An overview. *IEEE Signal Process. Mag.* **2018**, *35*, 16–34. [CrossRef]
22. Forster, M.R. Key concepts in model selection: Performance and generalizability. *J. Math. Psychol.* **2000**, *44*, 205–231. [CrossRef]
23. Arlot, S.; Celisse, A. A survey of cross-validation procedures for model selection. *Stat. Surv.* **2010**, *4*, 40–79. [CrossRef]
24. Burnham, K.P.; Anderson, D.R. Multimodel inference: Understanding AIC and BIC in model selection. *Sociol. Methods Res.* **2004**, *33*, 261–304. [CrossRef]
25. Kadane, J.B.; Lazar, N.A. Methods and criteria for model selection. *J. Am. Stat. Assoc.* **2004**, *99*, 279–290. [CrossRef]
26. Raftery, A.E. Bayesian model selection in social research. *Sociol. Methodol.* **1995**, *25*, 111–163. [CrossRef]
27. Wit, E.; van der Heuvel, E.; Romeijn, J.W. 'All models are wrong...': An introduction to model uncertainty. *Stat. Neerl.* **2012**, *66*, 217–236. [CrossRef]

28. Aho, K.; Derryberry, D.; Peterson, T. Model selection for ecologists: The worldviews of AIC and BIC. *Ecology* **2014**, *95*, 631–636. [CrossRef] [PubMed]
29. Zucchini, W. An introduction to model selection. *J. Math. Psych.* **2000**, *44*, 41–61. [CrossRef]
30. R Development Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2008; ISBN 3-900051-07-0.
31. Sheather, S. *A Modern Approach to Regression With R*; Springer Science & Business Media: New York, NY, USA, 2009.
32. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B* **1996**, *58*, 267–288. [CrossRef]
33. Hastie, T.; Tibshirani, R.; Wainwright, M. *Statistical Learning with Sparsity: The Lasso And Generalizations*; CRC Press: Boca Raton, FL, USA, 2015.
34. Hoerl, A.E.; Kennard, R.W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **1970**, *12*, 55–67. [CrossRef]
35. Friedman, J.; Hastie, T.; Tibshirani, R. Glmnet: Lasso and elastic-net regularized generalized linear models. *R Package Version* **2009**, *1*.
36. Zou, H. The adaptive lasso and its oracle properties. *J. Am. Stat. Assoc.* **2006**, *101*, 1418–1429. [CrossRef]
37. Zou, H.; Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2005**, *67*, 301–320. [CrossRef]
38. Abu-Mostafa, Y.S.; Magdon-Ismail, M.; Lin, H.T. *Learning from Data*; AMLBook: New York, NY, USA, 2012; Volume 4.
39. Geman, S.; Bienenstock, E.; Doursat, R. Neural networks and the bias/variance dilemma. *Neural Comput.* **1992**, *4*, 1–58. [CrossRef]
40. Kohavi, R.; Wolpert, D.H. Bias plus variance decomposition for zero-one loss functions. In Proceedings of the 13th International Conference on Machine Learning, Bari, Italy, 3–6 July 1996; Volume 96, pp. 275–283.
41. Geurts, P. Bias vs. variance decomposition for regression and classification. In *Data Mining and Knowledge Discovery Handbook*; Springer: Boston, MA, USA, 2009; pp. 733–746.
42. Weinberger, K. Lecture Notes in Machine Learning (CS4780/CS5780). 2017. Available online: http://www.cs.cornell.edu/courses/cs4780/2017sp/lectures/lecturenote11.html (accessed on 1 January 2019).
43. Nicholson, A.M. Generalization Error Estimates and Training Data Valuation. Ph.D. Thesis, California Institute of Technology, Pasadena, CA, USA, 2002.
44. Wang, J.; Shen, X. Estimation of generalization error: Random and fixed inputs. *Stat. Sin.* **2006**, *16*, 569.
45. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006.
46. Forster, M.R. Predictive accuracy as an achievable goal of science. *Philos. Sci.* **2002**, *69*, S124–S134. [CrossRef]
47. Draper, N.R.; Smith, H. *Applied Regression Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2014; Volume 326.
48. Wright, S. Correlation of causation. *J. Agric. Res.* **1921**, *20*, 557–585.
49. Gilmour, S.G. The interpretation of Mallows's C_p-statistic. *J. R. Stat. Soc. Ser. D (Stat.)* **1996**, *45*, 49–56.
50. Zuccaro, C. Mallows? Cp statistic and model selection in multiple linear regression. *Mark. Res. Soc. J.* **1992**, *34*, 1–10. [CrossRef]
51. Akaike, H. A new look at the statistical model identification. In *Selected Papers of Hirotugu Akaike*; Springer: New York, NY, USA, 1974; pp. 215–222.
52. Symonds, M.R.; Moussalli, A. A brief guide to model selection, multimodel inference and model averaging in behavioural ecology using Akaike's information criterion. *Behav. Ecol. Sociobiol.* **2011**, *65*, 13–21. [CrossRef]
53. Schwarz, G. Estimating the dimension of a model. *Ann. Stat.* **1978**, *6*, 461–464. [CrossRef]
54. Neath, A.A.; Cavanaugh, J.E. The Bayesian information criterion: Background, derivation, and applications. *Wiley Interdiscip. Rev. Comput. Stat.* **2012**, *4*, 199–203. [CrossRef]
55. Kass, R.E.; Raftery, A.E. Bayes factors. *J. Am. Stat. Assoc.* **1995**, *90*, 773–795. [CrossRef]
56. Morey, R.D.; Romeijn, J.W.; Rouder, J.N. The philosophy of Bayes factors and the quantification of statistical evidence. *J. Math. Psychol.* **2016**, *72*, 6–18. [CrossRef]
57. Lavine, M.; Schervish, M.J. Bayes factors: What they are and what they are not. *Am. Stat.* **1999**, *53*, 119–122.
58. Jaynes, E.T. *Probability Theory: The Logic of Science*; Cambridge University Press: Cambridge, UK, 2003.
59. Vrieze, S.I. Model selection and psychological theory: A discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). *Psychol. Methods* **2012**, *17*, 228. [CrossRef] [PubMed]

60. Yang, Y. Can the strengths of AIC and BIC be shared? A conflict between model indentification and regression estimation. *Biometrika* **2005**, *92*, 937–950. [CrossRef]
61. Kuha, J. AIC and BIC: Comparisons of assumptions and performance. *Sociol. Methods Res.* **2004**, *33*, 188–229. [CrossRef]
62. Beale, E.; Kendall, M.; Mann, D. The discarding of variables in multivariate analysis. *Biometrika* **1967**, *54*, 357–366. [CrossRef] [PubMed]
63. Derksen, S.; Keselman, H.J. Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables. *Br. J. Math. Stat. Psychol.* **1992**, *45*, 265–282. [CrossRef]
64. Geisser, S. The predictive sample reuse method with applications. *J. Am. Stat. Assoc.* **1975**, *70*, 320–328. [CrossRef]
65. Stone, M. Cross-validatory choice and assessment of statistical predictions. *J. R. Stat. Soc. Ser. B (Methodol.)* **1974**, *36*, 111–147. [CrossRef]
66. Good, P.I. *Resampling Methods*; Springer: Boston, MA, USA, 2006.
67. Schumacher, M.; Holländer, N.; Sauerbrei, W. Resampling and cross-validation techniques: A tool to reduce bias caused by model building? *Stat. Med.* **1997**, *16*, 2813–2827. [CrossRef]
68. Efron, B. *The Jackknife, the Bootstrap, and Other Resampling Plans*; Siam: Philadelphia, PA, USA, 1982; Volume 38.
69. Efron, B.; Tibshirani, R. *An Introduction to the Bootstrap*; Chapman and Hall/CRC: New York, NY, USA, 1994.
70. Wehrens, R.; Putter, H.; Buydens, L.M. The bootstrap: A tutorial. *Chemometr. Intel. Lab. Syst.* **2000**, *54*, 35–52. [CrossRef]
71. Krstajic, D.; Buturovic, L.J.; Leahy, D.E.; Thomas, S. Cross-validation pitfalls when selecting and assessing regression and classification models. *J. Cheminform.* **2014**, *6*, 10. [CrossRef]
72. Molinaro, A.M.; Simon, R.; Pfeiffer, R.M. Prediction error estimation: A comparison of resampling methods. *Bioinformatics* **2005**, *21*, 3301–3307. [CrossRef] [PubMed]
73. Amari, S.I.; Fujita, N.; Shinomoto, S. Four types of learning curves. *Neural Comput.* **1992**, *4*, 605–618. [CrossRef]
74. Amari, S.I. A universal theorem on learning curves. *Neural Netw.* **1993**, *6*, 161–166. [CrossRef]
75. Cawley, G.C.; Talbot, N.L. On over-fitting in model selection and subsequent selection bias in performance evaluation. *J. Mach. Learn. Res.* **2010**, *11*, 2079–2107.
76. Guyon, I.; Saffari, A.; Dror, G.; Cawley, G. Model selection: Beyond the bayesian/frequentist divide. *J. Mach. Learn. Res.* **2010**, *11*, 61–87.
77. Piironen, J.; Vehtari, A. Comparison of Bayesian predictive methods for model selection. *Stat. Comput.* **2017**, *27*, 711–735. [CrossRef]
78. Good, I. Explicativity: A mathematical theory of explanation with statistical applications. *Proc. R. Soc. Lond. A* **1977**, *354*, 303–330. [CrossRef]
79. Chen, H.; Chiang, R.H.; Storey, V.C. Business intelligence and analytics: From big data to big impact. *MIS Q.* **2012**, *36*, 1165–1188. [CrossRef]
80. Erevelles, S.; Fukawa, N.; Swayne, L. Big Data consumer analytics and the transformation of marketing. *J. Bus. Res.* **2016**, *69*, 897–904. [CrossRef]
81. Jin, X.; Wah, B.W.; Cheng, X.; Wang, Y. Significance and challenges of big data research. *Big Data Res.* **2015**, *2*, 59–64. [CrossRef]
82. Holzinger, A.; Kieseberg, P.; Weippl, E.; Tjoa, A.M. Current advances, trends and challenges of machine learning and knowledge extraction: From machine learning to explainable ai. In *Proceedings of the International Cross-Domain Conference for Machine Learning and Knowledge Extraction, Hamburg, Germany, 27–30 August 2018*; Springer: Cham, Switzerland, 2018; pp. 1–8.
83. Lynch, C. Big data: How do your data grow? *Nature* **2008**, *455*, 28–29. [CrossRef] [PubMed]