

High-Dimensional Data and the Bias Variance Tradeoff in Model Selection

Eligo Workineh Menna

Northwest University, School of Mathematics, Xi'an 710127, China
Email: worke2003@gmail.com

ABSTRACT

The exploration of function estimation from a specific dataset was primarily theoretical until the 1990s, despite the development of statistical learning theory in the late 1960s. This shift allowed statistical learning theory to be applied in practical multidimensional function estimation methods alongside theoretical evaluations. Statistical learning represents a contemporary approach to machine learning that employs mathematical models, algorithms, and statistical techniques to detect patterns and make predictions from large datasets. The extensive use of statistical learning in fields ranging from genetics and biology to economics and social datasets is clear from a thorough examination of the literature. This methodology has recently been used in a number of disciplines, including computer science, biology, finance, and social science. Model selection is one of the main obstacles to solving each of these studies' problems. Because the influence of bias-variance tradeoff in model selection plays a vital role especially in analyzing high-dimensional data. This comprehensive scientific review aims to investigate the role of the bias-variance tradeoff in model selection for high-dimensional datasets. We examine the concepts of bias and variance, their interrelationship, and their effects on model performance regarding prediction accuracy and generalization. Additionally, we highlight numerous strategies and techniques used to balance the bias-variance tradeoff during model selection. This review provides insights into the importance of understanding and managing the bias-variance tradeoff for effective model selection in high-dimensional data analysis.

Key words: Biasvariance trade-off, Model selection, Regularization, Cross-validation, Ensemble Methods, Statistical Inference

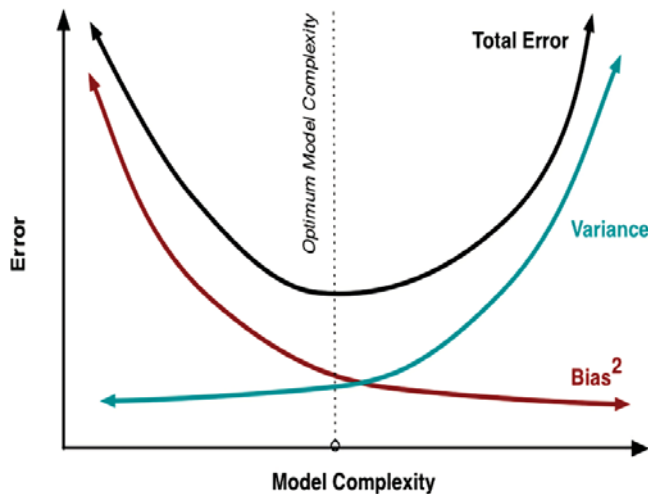
Mathematics Subject Classification: 62J12, 62G99

Computing Classification System: I.4

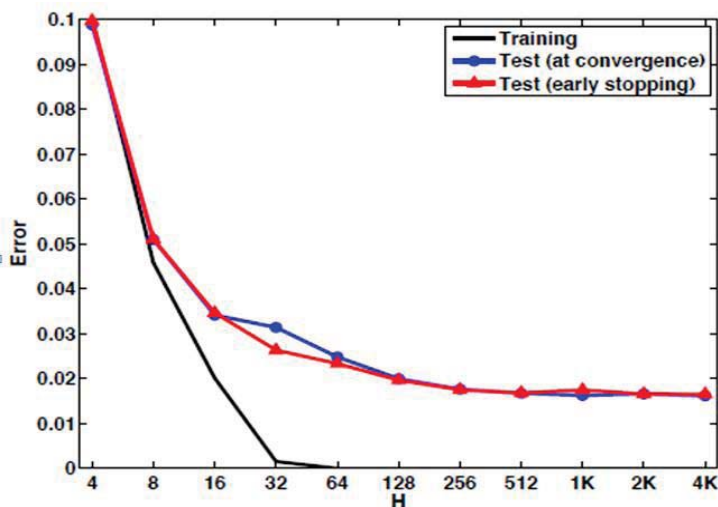
1. INTRODUCTION

Bias-variance tradeoff is the intricate balance between a model's bias (which leads to underfitting) and its variance (which causes overfitting). This idea is essential in machine learning and statistics, significantly influencing model performance. In highdimensional dataset scenarios, where the number of features larger than the number of samples, this tradeoff becomes even more crucial. The core idea is that for any given machine learning model, the complexity of it is inversely related to its bias and variance. As a model becomes more complex, its bias decreases while its variance increases. Conversely, a simpler model tends to have lower variance and higher bias. The objective is to search the maximum level of model complexity that decrease both variance and bias. The variance and bias tradeoff is key challenge in supervised learning, as machine learning algorithms aim for high accuracy and generalization of new data. However, there are always a tradeoff between the two main sources of errors named variance and bias. This tradeoff is known as the bias and variance tradeoff and which is vital checke the effectiveness of machine learning models.

The bias variance problem refers to the challenge of minimizing these two error sources simultaneously, which hinders supervised machine learning algorithms [1-2] from generalizing beyond their validation data. Bias is the error stemming from assumptions about the learning problem or model, while variance is the error arising from sensitivity to minor changes in the training dataset. Generally, a model with large bias will underfit the dataset and have poor performance in predictions, while model with large variance will overfit the dataset and struggle for generalize to new data. Thus, balancing bias and variance is crucial for achieving optimal performance [3]. The key principle is when model complexity varies, bias and variance interact, with overly simple models leading to underfitting (high bias) and overly complex models resulting in overfitting (high variance).



(a) Source—Fortmann-Roe [7], show U-shape test error curve of bias variance trade-off prediction.



(b) Source — Neyshabur [6] the test error truly decreased with neural network width.

Figure 1. Bias-variance tradeoff test errors predicted versus reality (Neural network)

The main implication of the variance bias tradeoff is test error typically follows U-shaped graph in relation to model complexity. Statistical learning theory [4] predicts this U-shaped test error curve for various traditional machine learning models by emphasizing model capacity as the primary factor influencing this trade-off. Empirical studies [5-11] suggest that larger networks frequently generalize better than smaller ones. However, sometime, a U-shaped test error graph may not be detected, (see the illustration in (Figure1 (b)) [6].

2. BIAS -VARIANCE AND MODEL SELECTION

Bias indicates the compromise between the variance and bias of the model. Bias is the error stemming from assumptions made about learning problem of model, while variance is the error caused by the model's sensitivity for minor changes in training dataset. Typically, a model having large bias will underfit the dataset, leading to poor predictive accuracy, whereas a model with large variance will overfit the dataset, resulting in inadequate generalization to new data. Thus, it is crucial to come up with balance between variance and bias to achieve best performance (see Figure 2). A significant challenge in supervised learning is managing the tradeoff between variance and bias.

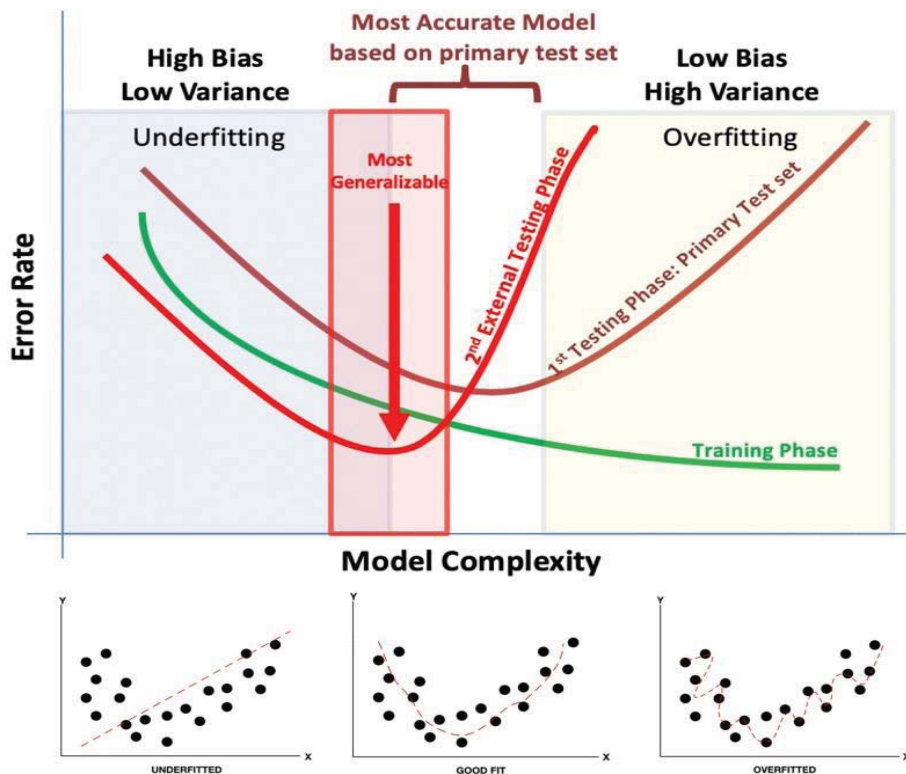


Figure 2. Illustrates the variance and bias tradeoff in machine learning. It shows that training dataset (green line) frequently does not fully reflect the outcomes from the testing phase. Underfitting results in low variability but big error rates and large bias (blue box). Conversely, overfitting leads to large variance and small bias (yellow box). The area between overfitting and underfitting is optimum, which may not be achieved till multiple testing efforts have been conducted (red line). (Figure is adapted from [78]).

Ideally, a model should be selected that accurately captures the patterns in the training dataset while also generalizing effectively to new or validation dataset. However, achieving both at the same time is most of the time not feasible. High variance learning techniques may represent for training dataset well but risk overfitting to noisy or unseen dataset. conversely, the algorithms with bias (large) tend to produce easy model that may underfit the data by not capturing important patterns [12]. High bias results in underfitting, which leads to poor model performance, while high variance indicates a model's sensitivity to variations in the training dataset. Whereas high variance can cause overfitting, where the model learns noise and struggles to generalize to new data [13-15].

In practice, the bias-variance trade-off can be managed through various techniques such as regularization and complexity control, cross-validation, holdout validation, grid search, model selection using information criteria, and ensemble methods among others. Regularization methods like L_1

and/or L_2 regularization will be used to minimize the involvedness of the model and thereby increase

the bias, while cross-validation can be used to select the ideal model hyperparameters in order to balance the bias and variance. Multiple models can be combined using ensemble approaches, including Random Forest and Gradient Boosting, to lower the final prediction's variance while preserving a low bias [16]. The idea of variance and bias tradeoff is vital in machine learning that must be handled to produce the best results in conclusion. One can create models that can precise and generalized prediction for a range of applications by comprehending and balancing a model's variance and bias. The variance and bias tradeoff can be further clear by decomposing the model total error into its variance and bias components. It can be represented mathematically as follows:

$$Error(x) = Bias^2 + Variance + Irreducible Error \text{ or}$$

$$E \left[(y - \hat{g}(x))^2 \right] = bias[\hat{g}(x)]^2 + var(\hat{g}(x)) + \sigma_\epsilon^2$$

where x is a sample of the data, as discussed above, bias represents the gap, between what the model predicts and the actual value variance indicates how much predictions might vary with deviations of training data and irreducible error cannot eliminate by model. This formula highlights the necessity to handle balance between bias and variance when selecting a model. Since variance and squared bias are always non negative it means that the total error can never be less than error.

Breaking it down further as we increase model complexity bias decreases but variance increases; on the hand reducing model complexity raises bias. Reduces variance. Therefore finding the sweet spot among bias and variance trade-off is crucial, in minimizing error and determining optimal model complexity [15].

3. STATISTICAL INFERENCE VERSUS BIAS AND VARIANCE

Suppose $S = \{(x_{(i)}, y_{(i)})\}_{i=1}^n$ be i.i.d. data set and it was produced from some data and resulting in

probability distribution having some constant new parameter $w^* \in \mathbb{R}^d$. And then $x_{(i)} \in \mathbb{R}^d$ and

$y_{(i)} \in \mathbb{R}^d$. Let $X \in \mathbb{R}^{n \times d}$ represent the data matrix, and $\vec{y} \in \mathbb{R}^n$ the labels vector. We consider that

in the case of regression X is given. Let the estimator denoted as $\hat{w}_n$ where n is the amount of the

dataset which used to fit the model. If we take into consideration that in linear regression, $\hat{w}_n = (X^T X)^{-1} X^T \vec{y}$. It is clear that $\hat{w}_n$ is a random variable even though w^* is not random.

Because $\hat{w}_n$ is known function of the noisy dataset S , where noisiness in the labels $\vec{y}$ is typical; since

it worth noting that the randomness in $\hat{w}_n$ which indirectly depends on w^* . Commonly $\hat{w}_n$ is called

the *Sampling distribution*. The 1st and 2nd moments (centred) are bias and variance estimator of $\hat{w}_n$.

The bias of the estimator $\hat{w}_n$ can be called, Bias ($\hat{w}_n$) $\equiv E[\hat{w}_n - w^*]$. It is unbiased if

$E[\hat{w}_n - w^*] = 0$ (i.e. $E[\hat{w}_n - w^*] = 0$) for all values of w^* . Similarly, if $\text{Var}(\hat{w}_n \equiv \text{Cov}[\hat{w}_n])$ the

variance of the estimator $\hat{w}_n$. It should be noted that, the variance of the estimator does not directly

depend on the true parameter w^* .

Similar to how the 1st and 2nd moments of any distribution are not always related, the variance and bias of an estimator are not always directly related [17]. It may have either large or small variance and large or small bias estimators. The relationship between an estimator's variance and bias under squared error is as follows:

$$\begin{aligned} \text{Mean Square Error } (\hat{w}_n) &= E[||\hat{w}_n - w^*||^2] \\ &= E[||\hat{w}_n - E[\hat{w}_n] + E[\hat{w}_n] - w^*||^2] \\ &= E \left[||\hat{w}_n - E[\hat{w}_n]||^2 + \underbrace{||E[\hat{w}_n] - w^*||^2}_{\text{Constant}} + 2 \underbrace{(\hat{w}_n - E[\hat{w}_n])^T (E[\hat{w}_n] - w^*)}_{\text{Zero mean}} \right] \\ &= E[||\hat{w}_n - E[\hat{w}_n]||^2] + ||E[\hat{w}_n] - w^*||^2 \end{aligned}$$

$$\begin{aligned}
 &= \frac{E \left[\text{tr}[(\hat{w}_n - E[\hat{w}_n])(\hat{w}_n - E[\hat{w}_n])^T] \right]}{\text{tr}[Var(\hat{w}_n)]} + \frac{||E[\hat{w}_n] - w^*||^2}{||Bias(\hat{w}_n)||^2} \\
 &= \text{tr}[Var(\hat{w}_n)] + ||Bias(\hat{w}_n)||^2.
 \end{aligned}$$

It is frequently happens that methods used to lessen variance also result in high variance, and vice versa, this occurrence gave the name bias variance trade-off. The secret to effective model creation is finding the best approach to balance the two undesirables, bias and variance.

4. TECHNIQUES FOR BALANCING BIAS-VARIANCE TRADEOFF AND MODEL SELECTION

Model selection in high-dimensional data poses significant difficulty due to the bias-variance tradeoff. This review explores that how to balance between variance and bias in the processes of model selection. We discuss recent advancements in this field, highlighting key methodologies and their applications.

In high-dimensional data analysis, choosing the appropriate model is the fundamental goal. However, selecting an optimal model often involves navigating the bias-variance tradeoff. High-dimensional datasets are characterized by a huge number of features as compared as the number of observations, which increased model complexity and potentially overfitting. To obtain the reliable and generalizable model, balancing variance and bias is crucial [18-19].

4.1 Prediction supervised machine learning

In the prediction processes the goals of supervised machine learning and statistical inference is quite difference. Rather than creating an estimator $\hat{w}_n$ for the unknown parameter, our goal is to develop

a function g that can make accurate predictions y given x , according to a specific loss function. It

was previously mentioned the training set $S = \{(x_{(i)}, y_{(i)})\}_{i=1}^n$ is provided. We make the assumption

that $y = g(x) + \varepsilon$, where ε satisfies $E(\varepsilon) = 0$ and $Var(\varepsilon) = \sigma^2$ (ε is not necessarily Gaussian).

Additional, we define (rather than assume) the “true” g as:

$$g(x') \equiv E[y|x = x'].$$

Now, given a fixed size training set S , our aim is to build a hypothesis $\hat{g}_n$ that accurately replicates g on all future unknown examples. Put differently, $\hat{g}_n$ must have a good generalization error. We restrict

our consideration to the scenario in which the generalization error on an unseen sample is equal to the expected squared error loss.

Suppose $\hat{g}_n$ is derived from a certain (not specified) training method applied to S . As previously mentioned, $\hat{g}_n$ is random, and this randomness arises from the $\varepsilon^{(i)}$'s embedded elements in the training set samples. Now, take into account a new, unseen example pair (y_*, x_*) and the associated generalization error, where the expectation is based on the randomness present in the embedded test example and in $\hat{g}_n$:

$$\begin{aligned}
 MSE(\hat{g}_n) &= E[(y_* - \hat{g}_n(x_*))^2] \\
 &= E[(\varepsilon + g(x_*) - \hat{g}_n(x_*))^2] \\
 &= E[\varepsilon^2] + E[(g(x_*) - \hat{g}_n(x_*))^2] + E[2\varepsilon(g(x_*) - \hat{g}_n(x_*))] \\
 &= E[\varepsilon^2] + E[(g(x_*) - \hat{g}_n(x_*))^2] + \underbrace{E[\varepsilon]E[2(g(x_*) - \hat{g}_n(x_*))]}_{=0} \text{ where (i.i.d. } \varepsilon) \\
 &= E[\varepsilon^2] + E[(g(x_*) - \hat{g}_n(x_*))^2] \\
 &= E[\varepsilon^2] + E[g(x_*) - \hat{g}_n(x_*)]^2 \\
 &\quad + Var[g(x_*) - \hat{g}_n(x_*)]; \text{ Recall } (E[X^2] = Var[X] + E[X]^2) \\
 &= \underbrace{\varphi^2}_{\text{Irreducible error}} + \underbrace{E[\hat{g}_n(x_*) - g(x_*)]^2}_{\text{Bias}^2} + \underbrace{Var[\hat{g}_n(x_*)]}_{\text{Variance}}; \text{ where } (Var[a - X] = Var[X])
 \end{aligned}$$

According to the decomposition above, the environment is similar to statistical inference. As before, the variance and bias consists of the 1st and 2nd moments of the (centred) function $\hat{g}_n$ (avoiding x_* in

the notation). Variance is $Var[\hat{g}_n]$, and the bias of $\hat{g}_n$ at input x_* is defined as $E[\hat{g}_n - g]$. Only for

the squared error loss does such a clear split into Bias and Variance terms exist. There have been suggestions for larger losses, but none have gained widespread acceptance.

However, the bias and variance tradeoff is too difficult issues at the time of model selection. But there are some techniques to overcome the trade-off supervised machine learning bias-variance is reviewed as follow:

4.2 Techniques for balancing bias-variance tradeoff

4.2.1 Regularization methods

The regularization methods like ridge regression and Lasso-Least Absolute Shrinkage and Selection Operator have gained popularity in high-dimensional model selection due to their ability to shrink less important predictors towards zero. These methods help control overfitting by reducing variance while introducing a controlled amount of bias [20-21].

L_1 and L_2 regularization are techniques used in machine learning to prevent overfitting of models.

These method encourage the model to have simpler weights by including penalty term(s) to the function of loss. L_1 regularization called Lasso, adds penalty term equal to the absolute value of the weights to the loss function. Mathematically, this can be represented as:

$$L_1 - \text{regularization} = \text{Loss function} + \lambda * \sum_{i=1}^n |w_i| \text{ (i. e. as follow)}$$

$$\text{Cost} = \underbrace{\sum_{j=1}^m \left(y_i - \sum_{i=1}^n x_{ij} w_i \right)^2}_{\text{Loss function}} + \underbrace{\lambda * \sum_{i=1}^n |w_i|}_{\text{Regularization term}}$$

where λ denoted as hyperparameter that prevents the strength of regularization.

On the other hand, L_2 regularization (or called ridge regularization), adds penalty term equal to the square of the weights to the function of loss. Mathematically, this can be represented as:

$$L_2 - \text{Regularization loss} = \text{Loss function} + \lambda * \sum_{i=1}^n w_i^2$$

$$\text{Cost} = \underbrace{\sum_{j=1}^m \left(y_i - \sum_{i=1}^n x_{ij} w_i \right)^2}_{\text{Loss function}} + \underbrace{\lambda * \sum_{i=1}^n w_i^2}_{\text{Regularization term}}$$

where λ is prevents the strength of regularization and refers to hyper-parameter.

The advantages of L_1 regularization are:

- (1) It can be used for feature selection, as it tends to set the weights of irrelevant or redundant features to zero. This can help simplify the model and improve its interpretability.
- (2) Sparse models can be created where only subsets of features are selected.

The disadvantages of L_1 regularization are:

- (1) It may not work well when there are highly correlated features, as it tends to randomly select one of them and ignore the others.
- (2) It is sensitive to the scale of features, so scaling of feature is necessary.

Advantages of L_2 regularization are:

- (1) It is more numerically stable than L_1 regularization and can handle highly correlated features well.
- (2) It used to prevent overfitting and improve the generalization accuracy of the model.

Disadvantages of L_2 regularization:

- (1) It does not perform feature selection like L_1 regularization.
- (2) It can shrink all the weights towards zero, which may not be desirable if some features are more important than others.

Remark: In general, L_1 regularization penalizes the sum of weight's absolute values, whereas L_2 regularization penalizes the squares of the weights sum. L_1 regularization results is sparse solution, whereas L_2 regularization results non-sparse. L_1 and L_2 regularizations have different strengths and weaknesses, they can be used to avert overfitting and increase model performance in different scenarios.

We now take linear regression as a case study with L_2 – regularization, where this tradeoff can be easily handle. It is quite common when employing techniques to minimize variance results increase bias, and vice versa. This processes is known as bias variance tradeoff. Let us tie all this back to the situation of L_2 – regularized linear regression.

Assume $g(x) = w^{*T}x$ where w^* the true unknown parameter is. Now $\hat{g}_n(x) = \hat{w}_n^T x$ where $\hat{w}_n$ is an regularized estimator of L_2 linear regression. The relationship of variance and bias terms of prediction and of inference as follows:

$$\begin{aligned} Bias(\hat{g}_n) &= E[\hat{g}_n(x) - g(x)] \\ &= E[\hat{w}_n^T x - w^{*T} x] = E[\hat{w}_n - w^*]^T x \\ &= Bias(\hat{w}_n)^T x. \end{aligned}$$

(and similarly)

$$\begin{aligned} Var(\hat{g}_n) &= E[(\hat{g}_n(x) - E[\hat{g}_n(x)])^2] \\ &= E[(\hat{w}_n^T x - E[\hat{w}_n^T x])^2] \\ Var(\hat{g}_n) &= x^T [Var(\hat{w}_n)] x. \end{aligned}$$

Because it is a result of the noise in the test example (in the inference setting there is no such test example), where irreducible error only manifests in the prediction setting. In other words, the test example noise expresses itself as the error term called irreducible, while noise in training data adds to the variance term.

We need to eliminate one or more of the decomposed components in order to lower the generalization error. Since irreducible error is essentially data noise (i.e., the same x may have

various y values in different situations), there is nothing we can do to reduce it. The Bias and

Variance terms must then be balanced. We might think about changing the value of λ in the setting of L_2 —regularized linear regression. In the context of inference, usually increasing λ decreases

variance while raising Bias. Based on the relationships mentioned above, this tradeoff also immediately translates into the prediction setting. At the time, it is unclear what the ideal range for the λ value could be. However, there is an easy solution in the prediction setting: in cross-validation processes choosing the value of λ can reduce the error of squared (generalization error) [17].

Note that we assume in linear regression, $y_{(i)} = w^{*T} x_{(i)} + \varepsilon_{(i)}$ where each $\varepsilon_{(i)} \sim N(0, \varphi^2)$

independently identically distributed (i.i.d.); and we assume X is constant and given. To simplify

notation, let $\vec{\varepsilon} \in \mathbb{R}^n \sim N(\vec{0}, \varphi^2 I)$ where $\vec{\varepsilon}_i = \varepsilon^{(i)}$. So $\vec{y} = Xw^* + \vec{\varepsilon}$. We can recall that the

parameter $\lambda > 0$ minimizes the linear regression cost function with L_2 —regularization.

$$J(w) = \frac{\lambda}{2} \|w\|_2^2 + \frac{1}{2} \sum_{i=1}^n (y_{(i)} - w^T x_{(i)})^2,$$

and enjoys a closed form solution.

$$\begin{aligned} \hat{w}_n &= \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} J(w) \\ &= \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \left[\frac{\lambda}{2} \|w\|_2^2 + \frac{1}{2} \sum_{i=1}^n (y_{(i)} - w^T x_{(i)})^2 \right] \\ &= \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \left[\frac{\lambda}{2} \|w\|_2^2 + \frac{1}{2} \|Xw - \vec{y}\|_2^2 \right] \\ &= (X^T X + \lambda I)^{-1} X^T \vec{y} \end{aligned}$$

Consider the eigen decomposition of matrix $X^T X$, which is the symmetric Positive Semi Definite (PSD) :

$$X^T X = V \underbrace{\begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_d^2 \end{bmatrix}}_{\text{diag}(\sigma_1^2, \dots, \sigma_d^2)} V^T,$$

where $V^T V = V V^T = I$, and $\{\sigma_1^2, \sigma_2^2, \dots, \sigma_d^2\}$ are the eigenvalues, and $\sum \sigma_i^2 = 0$. Even though X and $X^T X$ is not full rank, adding $\lambda > 0$ to all the diagonal elements of $X^T X$ making $(X^T X + \lambda I)$ is symmetric and positive definite (PD).

$$X^T X + \lambda I = V \begin{bmatrix} \sigma_1^2 + \lambda & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_d^2 + \lambda \end{bmatrix} V^T$$

This implies that

$$(X^T X + \lambda I)^{-1} = V \begin{bmatrix} \frac{1}{\sigma_1^2 + \lambda} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\sigma_d^2 + \lambda} \end{bmatrix} V^T$$

Therefore $\hat{w}_n$ is unique and always exists. Then we can start the analysis of variance bias of $\hat{w}_n$ estimator as follow:

$$\begin{aligned} \hat{w}_n &= (X^T X + \lambda I)^{-1} X^T \vec{y} \\ &= (X^T X + \lambda I)^{-1} X^T (X w^* + \vec{\varepsilon}) \\ &= [(X^T X + \lambda I)^{-1} X^T X] w^* + [(X^T X + \lambda I)^{-1} X^T] \vec{\varepsilon} \end{aligned}$$

we can take an expectation of the above model to compute the bias by considering X as constant in regression:

$$\begin{aligned} E[\hat{w}_n] &= E[(X^T X + \lambda I)^{-1} X^T X] w^* + [(X^T X + \lambda I)^{-1} X^T] E[\vec{\varepsilon}] \\ &= [(X^T X + \lambda I)^{-1} X^T X] w^* + [(X^T X + \lambda I)^{-1} X^T] E[\vec{\varepsilon}] \\ &= [(X^T X + \lambda I)^{-1} X^T X] w^* + [(X^T X + \lambda I)^{-1} X^T] \vec{0} \\ &= [(X^T X + \lambda I)^{-1} X^T X] w^* \\ &= V \begin{bmatrix} \frac{1}{\sigma_1^2 + \lambda} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\sigma_d^2 + \lambda} \end{bmatrix} V^T X^T X w^* \end{aligned}$$

$$\begin{aligned}
 &= V \begin{bmatrix} \frac{1}{\sigma_1^2 + \lambda} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\sigma_d^2 + \lambda} \end{bmatrix} V^T V \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_d^2 \end{bmatrix} V^T \mathbf{w}^* \\
 &= V \begin{bmatrix} \frac{1}{\sigma_1^2 + \lambda} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\sigma_d^2 + \lambda} \end{bmatrix} \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_d^2 \end{bmatrix} V^T \mathbf{w}^* \\
 &= V \begin{bmatrix} \frac{\sigma_1^2}{\sigma_1^2 + \lambda} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\sigma_d^2}{\sigma_d^2 + \lambda} \end{bmatrix} V^T \mathbf{w}^*.
 \end{aligned}$$

We can draw a few conclusions from the above. First, when $\lambda = 0$, we see that $E[\hat{\mathbf{w}}_n] = \mathbf{w}^*$. This suggests that estimator is unbiased when using regularization of standard linear regression. Secondly, the above formula is essentially minimize $\mathbf{w}^*$ since the matrix of eigenvalues all are less than one. If we add large λ for more regularization, it yields the coefficients shrink toward 0 and the eigen values will be small. In this case the estimator $\hat{\mathbf{w}}_n$ of L_2 —regularized linear regression is biased and towards to 0.

While we incur the cost of introducing regularization by using a biased estimator, we also benefit from a decrease in variance. To examine the variance of the estimator $\hat{\mathbf{w}}_n$, let's first remember this

characteristic of multivariate Gaussians:

$$\text{If } \vec{\varepsilon} \sim N(\vec{0}, \varphi^2 I), \text{ then } A\vec{\varepsilon} \sim N(\vec{0}, (A\varphi^2 I)A^T).$$

This yields us (remember in regression X is constant and given) :

$$\begin{aligned}
 \text{Cov}[\hat{\mathbf{w}}_n] &= \text{Cov}[(X^T X + \lambda I)^{-1} X^T \vec{y}] \\
 &= \text{Cov}[(X^T X + \lambda I)^{-1} X^T (X \mathbf{w}^* + \vec{\varepsilon})] \\
 &= \text{Cov} \left[\underbrace{(X^T X + \lambda I)^{-1} X^T X \mathbf{w}^*}_{\text{Constant}} + \underbrace{((X^T X + \lambda I)^{-1} X^T) \vec{\varepsilon}}_{\text{Constant}} \right] \\
 &= \text{Cov}[(X^T X + \lambda I)^{-1} X^T \vec{\varepsilon}] \\
 &= [(X^T X + \lambda I)^{-1} X^T] \text{Cov}[\vec{\varepsilon}] [(X^T X + \lambda I)^{-1} X^T]^T \\
 &\quad \text{(using above property)}
 \end{aligned}$$

$$\begin{aligned}
 &= [(X^T X + \lambda I)^{-1} X^T] \varphi^2 I [X(X^T X + \lambda I)^{-1}] \\
 &= \varphi^2 (X^T X + \lambda I)^{-1} (X^T X) (X^T X + \lambda I)^{-1} \\
 &= \varphi^2 V \begin{bmatrix} \frac{1}{\sigma_1^2 + \lambda} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\sigma_d^2 + \lambda} \end{bmatrix} V^T V \begin{pmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_d^2 \end{pmatrix} \\
 &\quad V^T V \begin{bmatrix} \frac{1}{\sigma_1^2 + \lambda} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\sigma_d^2 + \lambda} \end{bmatrix} V^T \\
 &= V \begin{bmatrix} \frac{\varphi^2 \sigma_1^2}{(\sigma_1^2 + \lambda)^2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\varphi^2 \sigma_d^2}{(\sigma_d^2 + \lambda)^2} \end{bmatrix} V^T.
 \end{aligned}$$

From the aforementioned expression, we may deduce that all the eigenvalues of the covariance of the estimator $\hat{w}_n$ is reduced as regularization is increased (i.e. larger value of λ), resulting in smaller $\text{tr}[\text{Var}(\hat{w}_n)]$. As a result, we can clearly observe how affects the bias and variance tradeoff. The bias (unwanted) of $\hat{w}_n$ is higher when large value of λ , while the Variance (wanted) of $\hat{w}_n$ is smaller the larger the value of λ , and vice versa. Identifying the optimal point is best addressed in relation to prediction, which is covered in the section 4.1 above. There is a sweet spot for λ that minimizes the sum of the two evils.

4.2.2 Cross-validation

Crossvalidation is popular technique for assessing machine learning model performance while reducing the risk of over-fitting. In the crossvalidation method, the given data is divided into multiple groups of training and validation sets, and the model is trained on some of them while tested on the remainder iteratively. Techniques of cross-validation approaches (repeatedly dividing the data into training and validation sets), like leave-one-out crossvalidation and k-fold crossvalidation, produce reliable estimates of model performance. Cross-validation helps to balance bias and variance by figuring out the model's ideal level of complexity [22-23].

In other word, crossvalidation involves splits the dataset into different groups (called k —folds), where one fold is works as a validation set and the other $k - 1$ folds are works for model training. Using each fold once as test set and repeated for k times. The average performance of the model on the k

test sets is then used as an estimate of the model's generalizability. Cross-validation can be implemented using different strategies, such as k -fold, leave-one-out, and stratified variants. These

methods permits for a more accurate assessment of the model's performance on unobserved data. Recent studies have explored variations of cross validation, such as stratified sampling, nested cross validation, and leave-one-out cross-validation (LOOCV).

4.2.2.1 Stratified sampling

Stratified sampling is a variation of cross validation that is broadly applied in machine learning to ensure representative and unbiased evaluation of model performance, particularly in imbalanced dataset. Usually this method is preferable when the dataset is imbalanced (the sample samples in each group is significantly different) [17]. Lyons [24] used cross-validation to test different designs such as stratification by simple random block holdout by class and stratification by both space and class for the assessment of classification and accuracy on remote sensing. They discover that all of the resampling techniques produce accurate estimates of error compared to the conventional approach called overall accuracy. There are many research works regarding to stratify sampling in imbalanced data [25].

Distribution-balanced stratified cross-validation (DBSCV), first established by [26], enhances the quality of estimation by supplying distributions balanced intra-class when subdividing dataset into several groups. They used 9 real world dataset to test DBSCV and three domains for simulation using C4.5 decision trees classifier. The findings demonstrate that, in the majority of situations, DBSCV outperforms (has smaller biases than) conventional stratified cross-validation, particularly when the number of folds is low. Prusty [27] were used cross-validation (stratified k -fold) method to predict cervical cancer together to other classifier like KNN- k -Nearest Neighbours, SVM-support vector machine, extreme gradient boosting and random forest could be a good option for assisting clinical experts in classifying the cervical cancer disease in advance.

According to [28], the effect of the over-samplers was investigated somewhat classification results large F_1 and AUC values in using decision tree, SVM, MLP and KNN, classifiers among others and the findings also demonstrate that choosing sample classifier pair is more crucial for classification performance too when applying optimally balanced distribution SCV (DOB-SCV). Whereas poor validation methods can lead to false results (for example due to shifting data), which motivates the advancement of techniques for imbalanced datasets, like SCV-stratified crossvalidation and distribution optimally balanced SCV (DOB-SCV) [29]. Research results has proved that applying DOB-SCV instead of SCV on imbalanced data sets can lead to higher classification performance scores (AUC).

4.2.2.2 Nested cross validation

Nested crossvalidation is a method that improves estimation and ability of the model by splitting the dataset into multiple train-test splits. It involves inner loop (model evaluation) and an outer loop (tuning hyper-parameter). The outer loop each iteration performs model evaluation using a different

train-test split [30]. In the absence of hyperparameter optimization usually called model selection, simple crossvalidation understates ability of the model given after training using whole dataset. Simple crossvalidation overestimates the models ability when learning methods and hyper-parameter optimization are present. The method suggested by [31] to lessen the bias and estimate, as well as nested crossvalidation, it is model selection approach (which includes inner crossvalidation), are some additional options [32]. Raschka [33] was reviewed the bias variance tradeoff to how to select k, common crossvalidation methods including leave-one-out crossvalidation and k-fold crossvalidation, and useful advice for the best choice of k is provided based on experiential data. They describe various statistical tests for comparing algorithms and explore methods for handling different evaluations, such as omnibus-tests and multiplecomparison corrections. And finally, when working with small datasets, alternative approaches to selection of algorithm, like combined F-test 5x2 crossvalidation and nested crossvalidation were advised.

Many cross validation split methods were compared in detail by Tiittanen [34] who found that opt split produced more evenly spaced ruptures than the other approaches and was good enough to be applied to large Gene-Ontology (GO) data sets. In contrast the most popular cross validation split quality measures did not perform well when applied to multi-label data with significant class imbalance.

4.2.2.3 Cross validation leave-one-out (LOOCV)

LOOCV is types of variation of crossvalidation when each sample in the dataset serves as the test set once. In other words in LOOCV we create K validation sets each containing a single sample and use the rest samples as the training set. This approach yields an unbiased estimate of model but computationally expensive for large datasets [35].

Jiang and Simon [36] were introduced repeated leave-one-out bootstrap (RLOOB) approach to prediction and they proposed the adjusted bootstrap (ABS) to fit the learning curve, and ABS technique offers a small conservative estimation error of prediction and is reliable throughout the circumstances which they examined. It doesn't suffer from a significant upward bias like bootstrap (leave-one-out) and the 0.632+ bootstrap [37], nor does it have a significant variability like the leave-one-out crossvalidation in microarray implementation, even with tiny samples [38-40] was used LOOVC for classification by employing SVM, AdaBoost and novel clustering based classification technique.

The variance and bias tradeoff and its applications in cross-validation have recently been the subject of numerous studies. For example, [41] used cross-validation to examine the bias variance trade-off in deep learning techniques. They found that maximizing the model size results minimize the bias but an maximizin variance, suggesting importance of choosing an appropriate model complexity for achieving optimal results. Similarly, [42] explored the variance bias tradeoff in the context of classification tasks. Using cross-validation, the authors showed that maximizing the size of model parameters decreasing bias but increase variance, and the optimal level of complexity could be obtained by balancing these two factors.

4.3 Information criteria

Methods to select model, such as AIC- Akaike Information Criterion and BIC- Bayesian information criterion, offer quantitative measures to assess the tradeoff of model fit and complexity. These techniques penalize models with excessive complexity, discouraging overfitting and promoting parsimony [43 - 44].

The general approach of Akaike information criterion (AIC) is:

$$AIC = \frac{2K - 2\log(L)}{N}$$

where

K is the number of predictors.

L is the likelihood of model

N is the sample size (especially helpful in the case of small data set). And

$$BIC = \ln(n) k - 2\ln(\hat{L})$$

where

$\hat{L}$ - the maximum value of the likelihood function of model.

n - the sample size

k - the amount of free parameters to be estimated

Recent research has extended these criteria to handle complex models and high-dimensional data [45-46]. The three modified versions of BIC—mBIC2, EBIC, and mBIC—were analysed by [47], who concluded that EBIC and mBIC2 have greater influence and large wrong finding rate than mBIC. As a result, for all three variants of the BIC, various estimating error metrics typically assume values that are similar. The effectiveness of LSR-least squares regression used as selection of important predictors in GLM-generalized linear model(s) was also studied by [47]. LSR achieves well if the data produced using logistic regression and significantly poor when they are produced using poisson regression. Chen and Chen [48], were introduced Bayesian information criteria (extended) (EBIC) method for model selection and which consider both model space complexity and the amount of unknown parameters. Schöninge [49] were applied Bayesian model evidence (BME) on Bayes' theorem, averaging objectively ranks or Bayesian model selection. And the results demonstrate the value of BME from ICs are frequently severely prejudiced and the selection of approximation techniques has major effect on the model ranking's correctness. It's recommended that numerical methods (bias-free) should be favoured over ICs if it is computationally feasible for reliable model selection.

4.4 Grid search

Grid search technique is the most popular hyper-parameter method of optimization and it plays crucial role in achieving an optimal balance among bias and variance. According to [18], variance bias

tradeoff states to the inherent tradeoff among the model's performance fitting the training dataset (small bias) and its generalization capacity to validation set of data (small variance). Grid search approach, also known as parameter sweep, exhaustively searches across a predetermined hyperparameter space to discover the ideal combination that minimizes both bias and variation.

Recent studies have proposed efficient search strategies, like random search and Bayesian optimization [50] to overcome the limitations of exhaustive grid search. Grid search hyperparameter optimization (GSHPO) was introduced by [51] as a method for predicting HIV/AIDS test results utilizing different model performance metrics, like: precision, accuracy, recall, F1-score, AUC-ROC, R^2 , MAE, RMSE, and confusion matrix for comparing the outcomes of various tests. The results of

GSHPO and 10-fold cross validation approaches were promising. For each model parameter, a grid of hyper-parameter values is defined, and using cross validation on each combination the model performance was assessed. Grid search aids in finding the optimal collection of hyper-parameters that balance bias and variance by methodically examining the hyper-parameter space [52].

To achieve the ideal balance between bias and variance [53], practitioners can fine-tune hyper-parameters like learning rate, regularisation strength, and feature selection criteria using grid search. It helps in choosing the best hyper-parameter configuration by allowing for practical assessment of the model's ability in various conditions. In numerous fields [54], including computer vision, natural NLP-language processing, biology, and finance, grid search has found extensive uses. It has been used to optimise [55] the hyper-parameters for SVM- support vector machines, RF- random forests, deep learning architectures, and other machine learning methods.

4.5 Ensemble methods

Multiple models are combined in ensemble methods to enhance prediction performance. In order to improve model selection recent research has concentrated on ensemble structure optimization and the incorporation of various models [56]. Additionally, since there are numerous baseline models, ensemble techniques can lessen the chance of overfitting. Multiple fields and domains have effectively used ensemble learning, which outperforms single models [57-65]. Random forests, gradient boosting, and stacking are popular ensemble techniques among others [66-67].

Dietterich [68], outlined the statistical, computational, and representational factors as the three main drivers behind ensemble methods' effectiveness. Ensemble approaches' efficacy is also explained by bias-variance decomposition [69] and strength correlation [70]. Widely used strategies for ensemble learning include bagging, boosting, and stacking. Bagging decreases variance while boosting reduces bias. Bagging is meant to decrease the variance without increasing the bias. This method works best when a learner's training set experiences small changes that result in significant changes to the predicted output. The variance is decreased by bagging by combining different models. These models have dissimilar statistical properties like the means and standard deviations, among others [71].

On high variance models like decision trees, bagging performs well. It is not anticipated to have an impact on the learning process for models with smaller variance, such linear regression. However, according to different research [72-73], accuracy declines when bagging is done on models with high level of bias. When contrasting the model's performance with and without bagging, this becomes evident. The accuracy will be greater without bagging than when we apply bagging to such a model.

5. DISCUSSION AND CONCLUSION

Selecting a model that is good at both capturing the regularities of training dataset and making good generalizations to new data is the ultimate goal. It is usually not possible to do both at the same time unfortunately. Generalization of model to achieve well on validation dataset is a key issues in evaluating the efficiency of a model in machine learning. If the given model is too complex may over-fit the training dataset, meaning that it has learned patterns that are specific to that dataset and cannot be generalized to validation set. On contrary, if a model that is too simple may under-fit the dataset and fails to capture important patterns which leading to low performance on both training and test datasets [74]. To strike a stability between model complexity and generalization, it is vital to use techniques such as regularization, crossvalidation, and hyper-parameter tuning. Regularization penalizes complex models and encourages simpler models that generalize better [75]. Crossvalidation has dividing a given dataset into test and train to see the model's effectiveness on different subsets of data sections. Hyper-parameter tuning involves choosing the maximum values for the parameters that determine model's complexity [76].

This review provides an overview of bias variance tradeoff in the course of choosing model techniques of supervised machine learning. Even though, bias variance tradeoff is the main obstacle to the performance of algorithms through optimization process there are ways of escape. Recent advancements in regularization, cross-validation, holdout validation, grid search, ensemble methods are widely used techniques to improve model selection practices. During model selection, the bias variance tradeoff has a vital role, so that many techniques proposed to handle these issues in supervised machine learning. Regularization techniques, like: L_1 and L_2 regularization, are common

approach to control model difficulty and address the variance bias tradeoff. Cross-validation has been recently the subject of numerous studies to overcome bias variance trade-off. For example, Yan [75] used cross-validation to examine the variance bias tradeoff in deep learning models. They found that increasing the model size results low bias but high variance, suggesting the importance of choosing an appropriate model complexity for achieving optimal results. Another study by [77] explored the variance bias tradeoff in the setting of classification tasks. Using cross-validation, the authors showed that increasing the number of model parameters reduced Bias but increased Variance, and that the optimal level of complexity could be obtained by balancing these two factors. Ensemble methods is one of model selection technique, which combine multiple models to improve predictive performance- such as random forests, gradient boosting, and stacking are some of common technique to overcome the variance bias tradeoff. Even though choosing model in deep learning, with its complex

architectures and large parameter spaces, poses unique challenges, some methods such as learning rate scheduling, early stopping, and architecture search, have shown promise in enabling effective model selection in deep learning.

5. REFERENCES

- [1] Kohavi, R. and Wolpert, D.H., 1996, July. Bias plus variance decomposition for zero-one loss functions. In *ICML* (Vol. 96, pp. 275-283).
- [2] Von Luxburg, U. and Schölkopf, B., 2011. Statistical learning theory: Models, concepts, and results. In *Handbook of the History of Logic* (Vol. 10, pp. 651-706). North-Holland. <https://doi.org/10.1016/B978-0-444-52936-7.50016-1>
- [3] Hastie, T., Tibshirani, R., Friedman, J.H. and Friedman, J.H., 2009. *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2, pp. 193-224). New York: springer.
- [4] Vapnik, V.N. and Vapnik, V., 1998. Statistical learning theory. pp. 219-223.
- [5] Neyshabur, B., Tomioka, R. and Srebro, N., 2014. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*.
- [6] Fortmann-Roe, S., 2012. Understanding the bias-variance tradeoff. I: URL: <http://scott.fortmann-roe.com/docs.BiasVariance.html> (hämtad 2017-09-29).
- [7] Zagoruyko, S. and Komodakis, N., 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.
- [8] Novak, R., Bahri, Y., Abolafia, D.A., Pennington, J. and Sohl-Dickstein, J., 2018. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*.
- [9] Lee, J., Bahri, Y., Novak, R., Schoenholz, S.S., Pennington, J. and Sohl-Dickstein, J., 2017. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*.
- [10] Belkin, M., Hsu, D., Ma, S. and Mandal, S., 2019. Reconciling modern machine-learning practice and the classical bias-variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32), pp.15849-15854. DOI: <https://doi.org/10.1073/pnas.1903070116>
- [11] Spigler, S., Geiger, M., d'Ascoli, S., Sagun, L., Biroli, G. and Wyart, M., 2019. A jamming transition from under-to over-parametrization affects generalization in deep learning. *Journal of Physics A: Mathematical and Theoretical*, 52(47), p.474001. DOI: 10.1088/1751-8121/ab4c8b.
- [12] Geman, S., Bienenstock, E. and Doursat, R., 1992. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1), pp.1-58.
- [13] James, G.M., 2003. Variance and bias for general loss functions. *Machine learning*, 51, pp.115-135.
- [14] Valentini, G. and Dietterich, T.G., 2004. Bias-variance analysis of support vector machines for the development of SVM-based ensemble methods. *Journal of Machine Learning Research*, 5(Jul), pp.725-775.
- [15] James, G., Witten, D., Hastie, T. and Tibshirani, R., 2013. *An introduction to statistical learning* (Vol. 112, p. 18). New York: springer. DOI: 10.1007/978-1-0716-1418-1.
- [16] Mccrea, N., 2016. An introduction to machine learning theory and its applications: A visual tutorial with examples. Διαθέσιμο στην ηλεκτρονική διεύθυνση: <https://www.toptal.com/machine-learning/machine-learning-theoryan-introductory-primer> [Τελευταία πρόσβαση 19 Μαΐου 2020].
- [17] Avati, A., 2019. Bias-variance analysis: theory and practice.
- [18] Hastie, T., Tibshirani, R., Friedman, J.H. and Friedman, J.H., 2009. *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2, pp. 1-758). New York: springer.
- [19] Doroudi, S., 2020. The bias-variance tradeoff: How data science can inform educational debates. *AERA open*, 6(4), DOI: <https://doi.org/10.1177/2332858420977208>.

- [20] McNeish, D.M., 2015. Using lasso for predictor selection and to assuage overfitting: A method long overlooked in behavioral sciences. *Multivariate behavioral research*, 50(5), pp.471-484. DOI: <https://doi.org/10.1080/00273171.2015.1036965>.
- [21] Emmert-Streib, F. and Dehmer, M., 2019. High-dimensional LASSO-based computational regression models: regularization, shrinkage, and selection. *Machine Learning and Knowledge Extraction*, 1(1), pp.359-383. DOI: <https://doi.org/10.3390/make1010021>.
- [22] Raschka, S., 2018. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*.
- [23] Tantithamthavorn, C., McIntosh, S., Hassan, A.E. and Matsumoto, K., 2016. An empirical comparison of model validation techniques for defect prediction models. *IEEE Transactions on Software Engineering*, 43(1), pp.1-18.
- [24] Lyons, M.B., Keith, D.A., Phinn, S.R., Mason, T.J. and Elith, J., 2018. A comparison of resampling methods for remote sensing classification and accuracy assessment. *Remote Sensing of Environment*, 208, pp.145-153. DOI: <https://doi.org/10.1016/j.rse.2018.02.026>.
- [25] Zhu, Y., Yan, Y., Zhang, Y. and Zhang, Y., 2020. EHSO: Evolutionary Hybrid Sampling in overlapping scenarios for imbalanced learning. *Neurocomputing*, 417, pp. 333-346. DOI: <https://doi.org/10.1016/j.neucom.2020.08.060>.
- [26] Zeng, X. and Martinez, T.R., 2000. Distribution-balanced stratified cross-validation for accuracy estimation. *Journal of Experimental & Theoretical Artificial Intelligence*, 12(1), pp.1-12. DOI: <https://doi.org/10.1080/095281300146272>.
- [27] Prusty, S., Patnaik, S. and Dash, S.K., 2022. SKCV: Stratified K-fold cross-validation on ML classifiers for predicting cervical cancer. *Frontiers in Nanotechnology*, 4. DOI: <https://doi.org/10.3389/fnano.2022.972421>.
- [28] Szeghalmy, S. and Fazekas, A., 2023. A comparative study of the use of stratified cross-validation and distribution-balanced stratified cross-validation in imbalanced learning. *Sensors*, 23(4). DOI: <https://doi.org/10.3390/s23042333>.
- [29] Moreno-Torres, J.G., Sáez, J.A. and Herrera, F., 2012. Study on the impact of partition-induced dataset shift on k -fold cross-validation. *IEEE transactions on neural networks and learning systems*, 23(8), pp.1304-1312. DOI: 10.1109/TNNLS.2012.2199516.
- [30] Varma, S. and Simon, R., 2006. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7, pp.1-8. DOI: <https://doi.org/10.1186/1471-2105-7-91>.
- [31] Tibshirani, R.J. and Tibshirani, R., 2009. A bias correction for the minimum error rate in cross-validation. *Ann. Appl. Stat* 3(2). pp. 822-829. DOI: 10.1214/08-AOAS224.
- [32] Tsamardinos, I., Rakhshani, A. and Lagani, V., 2015. Performance-estimation properties of cross-validation-based protocols with simultaneous hyper-parameter optimization. *International Journal on Artificial Intelligence Tools*, 24(05), pp. 1-30. DOI: <https://doi.org/10.1142/S0218213015400230>.
- [33] Raschka, S., 2018. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*.
- [34] Tiittanen, H., Holm, L. and Törönen, P., 2021. Novel split quality measures for stratified multilabel cross validation with application to large and sparse gene ontology datasets. 2(1), pp. 49-62. DOI: <https://doi.org/10.3934/aci.222003>.
- [35] Zad, N., Tell, L.A., Ramachandran, R.A., Xu, X., Riviere, J.E., Baynes, R., Lin, Z., Maunsell, F., Davis, J. and Jaber-Douraki, M., 2023. Development of machine learning algorithms to estimate maximum residue limits for veterinary medicines. *Food and Chemical Toxicology*, 179. DOI: <https://doi.org/10.1016/j.fct.2023.113920>.
- [36] Jiang, W. and Simon, R., 2007. A comparison of bootstrap methods and an adjusted bootstrap approach for estimating the prediction error in microarray classification. *Statistics in medicine*, 26(29), pp.5320-5334. DOI: <https://doi.org/10.1002/sim.2968>

- [37] Efron, B., 1983. Estimating the error rate of a prediction rule: improvement on cross-validation. *Journal of the American statistical association*, 78(382), pp.316-331. DOI:10.1080/01621459.1983.10477973.
- [38] Efron, B. and Tibshirani, R., 1997. Improvements on cross-validation: the 632+ bootstrap method. *Journal of the American Statistical Association*, 92(438), pp.548-560. DOI: 10.1080/01621459.1997.10474007.
- [39] Efron, B. and Tibshirani, R.J., 1998. An Introduction to the Bootstrap, 1. reprint.
- [40] Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M. and Yakhini, Z., 2000, April. Tissue classification with gene expression profiles. In *Proceedings of the fourth annual international conference on Computational molecular biology* (pp. 54-64). DOI: <https://doi.org/10.1145/332306.332328>.
- [41] Yan, Y., Zhang, M., Jiang, K., Zhong, Z., and Cai, L., 2021. Investigating bias-variance tradeoff in deep learning models using cross-validation, *Journal of Ambient Intelligence and Humanized Computing* 12(6). pp. 5531-5541.
- [42] Zhang, Y., Zhou, Y., Cai, Y., Lin, J., and Chen, D., 2021. Exploring the bias-variance tradeoff in classification models using cross-validation, *Proceedings of the IEEE International Conference on Artificial Intelligence and Education (ICAIE)*, pp.152-156.
- [43] Sharma, P., Sarstedt, M., Shmueli, G., Kim, K.H. and Thiele, K.O., 2019. PLS-based model selection: The role of alternative explanations in information systems research. *Journal of the Association for Information Systems*, 20(4). DOI: 10.17005/1.jais.00538.
- [44] Białaszek, W., Marcowski, P. and Ostaszewski, P., 2017. Physical and cognitive effort discounting across different reward magnitudes: Tests of discounting models. *PloS one*, 12(7). DOI: <https://doi.org/10.1371/journal.pone.0182353>.
- [45] Dziak, J.J., Coffman, D.L., Lanza, S.T., Li, R. and Jermiin, L.S., 2020. Sensitivity and specificity of information criteria. *Briefings in bioinformatics*, 21(2), pp.553-565. DOI: <https://doi.org/10.1093/bib/bbz016>.
- [46] Bogdan, M., Ghosh, J.K. and Żak-Szatkowska, M., 2008. Selecting explanatory variables with the modified version of the Bayesian information criterion. *Quality and Reliability Engineering International*, 24(6), pp.627-641. DOI: <https://doi.org/10.1002/qre.936>.
- [47] Żak-Szatkowska, M. and Bogdan, M., 2011. Modified versions of the Bayesian information criterion for sparse generalized linear models. *Computational Statistics & Data Analysis*, 55(11), pp.2908-2924. DOI: <https://doi.org/10.1016/j.csda.2011.04.016>.
- [48] Chen, J. and Chen, Z., 2008. Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3), pp.759-771. DOI: <https://doi.org/10.1093/biomet/asn034>.
- [49] Schöniger, A., Wöhling, T., Samaniego, L. and Nowak, W., 2014. Model selection on solid ground: Rigorous comparison of nine ways to evaluate Bayesian model evidence. *Water resources research*, 50(12), pp.9484-9513. DOI: <https://doi.org/10.1002/2014WR016062>.
- [50] Xia, Y., Liu, C., Li, Y. and Liu, N., 2017. A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *Expert systems with applications*, 78, pp.225-241. DOI: <https://doi.org/10.1016/j.eswa.2017.02.017>.
- [51] Belete, D.M. and Huchaiah, M.D., 2022. Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. *International Journal of Computers and Applications*, 44(9), pp.875-886. DOI: <https://doi.org/10.1080/1206212X.2021.1974663>.
- [52] Bergstra, J. and Bengio, Y., 2012. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2). pp. 281- 305. DOI:10.5555/2503308.2188395
- [53] Bergstra, J., Casagrande, N., Erhan, D., Eck D., and Kégl, B., 2011. Foundations of hyperparameter optimization, In *International Conference on Machine Learning*. pp. 693-700.

- [54] LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. *nature*, 521(7553), pp.436-444.
- [55] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. *The Journal of machine Learning research*, 12, pp.2825-2830.
- [56] Kumar, V., Aydav, P.S.S. and Minz, S., 2022. Multi-view ensemble learning using multi-objective particle swarm optimization for high dimensional data classification. *Journal of King Saud University-Computer and Information Sciences*, 34(10), pp.8523-8537. DOI: <https://doi.org/10.1016/j.jksuci.2021.08.029>.
- [57] Rodriguez, J.J., Kuncheva, L.I. and Alonso, C.J., 2006. Rotation forest: A new classifier ensemble method. *IEEE transactions on pattern analysis and machine intelligence*, 28(10), pp.1619-1630. DOI: 10.1109/TPAMI.2006.211.
- [58] Banfield, R.E., Hall, L.O., Bowyer, K.W. and Kegelmeyer, W.P., 2006. A comparison of decision tree ensemble creation techniques. *IEEE transactions on pattern analysis and machine intelligence*, 29(1), pp.173-180. DOI: 10.1109/TPAMI.2007.250609.
- [59] Ekbal, A. and Saha, S., 2011. A multiobjective simulated annealing approach for classifier ensemble: Named entity recognition in Indian languages as case studies. *Expert Systems with Applications*, 38(12), pp.14760-14772. DOI: <https://doi.org/10.1016/j.eswa.2011.05.004>.
- [60] De Stefano, C., Fontanella, F., Folino, G. and Di Freca, A.S., 2011. A Bayesian approach for combining ensembles of GP classifiers. In *Multiple Classifier Systems: 10th International Workshop, MCS 2011, Naples, Italy, June 15-17, 2011. Proceedings 10* (pp. 26-35). Springer Berlin Heidelberg.
- [61] Shahzad, R.K. and Lavesson, N., 2013. Comparative analysis of voting schemes for ensemble-based malware detection. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 4(1), pp.98-117.
- [62] Anwar, H., Qamar, U. and Muzaffar Qureshi, A.W., 2014. Global optimization ensemble model for classification methods. *The Scientific World Journal*, 2014(1). DOI: <https://doi.org/10.1155/2014/313164>.
- [63] Prusa, J., Khoshgoftaar, T.M. and Dittman, D.J., 2015, August. Using ensemble learners to improve classifier performance on tweet sentiment data. In *2015 IEEE international conference on information reuse and integration* (pp. 252-257). IEEE. DOI: 10.1109/IRI.2015.49.
- [64] Moparthy, A.N.R. and Geethanjali, B.D.N., 2016, February. Design and implementation of hybrid phase based ensemble technique for defect discovery using SDLC software metrics. In *2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)* (pp. 268-274). IEEE. DOI: 10.1109/AEEICB.2016.7538287.
- [65] El-Sappagh, S., Ali, F., Abuhmed, T., Singh, J. and Alonso, J.M., 2022. Automatic detection of Alzheimer's disease progression: An efficient information fusion approach with heterogeneous ensemble classifiers. *Neurocomputing*, 512, pp.203-224. DOI: <https://doi.org/10.1016/j.neucom.2022.09.009>.
- [66] Sagi, O. and Rokach, L., 2018. Ensemble learning: A survey. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 8(4). DOI: <https://doi.org/10.1002/widm.1249>.
- [67] Dong, X., Yu, Z., Cao, W., Shi, Y. and Ma, Q., 2020. A survey on ensemble learning. *Frontiers of Computer Science*, 14, pp.241-258. DOI: <https://doi.org/10.1007/s11704-019-8208-z>.
- [68] Dietterich, T.G., 2000, June. Ensemble methods in machine learning. In *International workshop on multiple classifier systems* (pp. 1-15). Berlin, Heidelberg: Springer Berlin Heidelberg. https://link.springer.com/chapter/10.1007/3-540-45014-9_1
- [69] Kohavi, R. and Wolpert, D.H., 1996, July. Bias plus variance decomposition for zero-one loss functions. In *ICML* (Vol. 96, pp. 275-283).

- [70] Lalithnarayan, C., Ensemble Learning on Bias and Variance, 2023. URL: <https://www.section.io/engineering-education/ensemble-bias-var/>
- [71] Van Wezel, M. and Potharst, R., 2007. Improved customer choice predictions using ensemble methods. *European Journal of Operational Research*, 181(1), pp.436-452. DOI: <https://doi.org/10.1016/j.ejor.2006.05.029>.
- [72] Caruana, R., Niculescu-Mizil, A., Crew, G. and Ksikes, A., 2004, July. Ensemble selection from libraries of models. In *Proceedings of the twenty-first international conference on Machine learning* (p. 18). DOI: <https://doi.org/10.1145/1015330.1015432>
- [73] Yan, Y. Zhang, M. Jiang, K. Zhong Z. and Cai L., 2021. Investigating bias-variance tradeoff in deep learning models using cross-validation, *Journal of Ambient Intelligence and Humanized Computing* 12(6), pp. 5531-5541. DOI: <https://doi.org/10.48550/arXiv.2205.06720>.
- [74] Zhang, Y. Zhou, Y. Cai, Y. Lin J., and Chen, D. 2021. Exploring the bias-variance tradeoff in classification models using cross-validation, *Proceedings of the 2021 IEEE International Conference on Artificial Intelligence and Education (ICAIE)*, pp.152-156.
- [75] Goodfellow, I., Bengio, Y. and Courville, A., 2016. *Deep learning*. MIT press.
- [76] Brownlee, J., 2016. Overfitting and underfitting with machine learning algorithms. *Machine Learning Mastery*, 21, p. 575. URL: <https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/>.
- [77] R. Grosse, 2021. Lecture 9: Generalization, University of Toronto: Toronto, ON, Canada, URL: https://www.cs.toronto.edu/~rgrosse/courses/csc321_2018/slides/lec09.pdf.
- [78] Rashidi, H.H., Tran, N.K., Betts, E.V., Howell, L.P. and Green, R., 2019. Artificial intelligence and machine learning in pathology: the present landscape of supervised methods. *Academic pathology*, 6. DOI: <https://doi.org/10.1177/2374289519873088>.