

Országos Tudományos Diákköri Konferencia 2023



Neighborhood Ranking-based Model-Free Feature Selection

Készítette: Ipkovich Ádám

Témavezető: Dr. Abonyi János

Pannon Egyetem, Mérnöki Kar, Folyamatmérnöki Tanszék

A pályamunka lezárásának dátuma: 2022.04.19

Contents

1	Introduction	2
2	Neighborhood ranking-based feature selection	7
2.1	The theory behind the neighborhood-based approach	7
2.2	k-nearest neighbors with leave-one-out regression	9
2.3	The false nearest neighbors method	10
2.4	Trustworthiness and continuity	13
2.5	Neighbourhood ranking-based feature selection	15
2.6	Discussion on the similarities of the measures	17
2.7	Application of the proposed metrics	20
3	Application of the neighborhood ranking-based feature selection	22
3.1	Linear, monotonous and periodic equations	22
3.2	Friedman-1 model	23
3.3	Dynamic modeling - polymerization reactor	28
4	Discussion and conclusion	30

Abstract

This study analyzes and generalizes neighborhood ranking-based feature selection methods. The development of model-independent feature selection algorithms is based on the widely applied k-nearest neighbors (k-NN) regression algorithm, where several neighboring data are directly used for prediction. Even if k-NN is a model-free technique, its performance is dependent on the relevance of the selected features that are often determined by leave-one-out cross-validation. I demonstrate that the false nearest neighbors (FNN) feature selection method can be considered a special case of the k-NN-based feature selection algorithm that evaluates the local functional relationship between the independent and dependent variables. I also study the trustworthiness continuity (T&C) metrics developed to assess neighborhood-preserving properties of nonlinear feature transformation and highlight that these metrics can also be used for feature selection. The FNN can be considered a special case of the continuity metric. Based on the detailed analysis and the generalization of the methods mentioned above, a novel neighborhood ranking-based feature selection (NRFS) technique has been developed based on local rank value differences in the neighborhoods of both the dependent and independent variables. Lastly, the connection between the NRFS and distance correlation (DC) is established. The NRFS can be considered the special case of DC, the rank distance correlation. The proposed methods are validated in benchmark datasets, such as the Friedman database. The local neighborhood considers the ranking in the independent variables and compares that to the corresponding rank in the dependent variable. I also show that the novel method can determine the order of nonlinear dynamical models. Due to the complexity of the benchmark datasets, the measures are incorporated into genetic and forward selection algorithms to select the correct combination of the features. A neural network is also used to validate the neighborhood-based algorithms. The results illustrate that the proposed k-NN, T&C, and the NRFS-based methods are capable of dealing with nonlinear correlation in complex modeling problems and the identification of order in dynamic systems.

Keywords: Model-free Feature Selection; Ranking difference; False nearest neighbor; k-nearest neighbor; Distance correlation; MATLAB/Octave Toolbox

Chapter 1

Introduction

Machine learning models revolutionize the way economy and science work, which makes data the new black gold. Yet, the application of models has a prerequisite: the inputs of the model have to be determined as accurately as possible. Even a non-relevant input variable may significantly decrease the performance of the model. Modeling and data science in other disciplines are widely applied: their use in process engineering, social, economic, and environmental studies provide information that is not trivial to uncover. For example, the merits of Big Data in sustainability and climate change include exploring connections between thousands of observations in greenhouse gasses, helping recognize opportunities, and determining the relationship between environmental and social factors [1]. Therefore feature selection methods were developed with the task to select the relevant features for increasing the speed and accuracy of machine learning methods.

Feature selection algorithms differ in approach [2, 3]. Filter methods are used as a preprocessing step as these algorithms are independent of machine learning models. Such techniques include correlation and distance-based approaches to select only the relevant variables. Models are trained on the selected variables. Wrapper methods use a set of variables and train models, thus computationally expensive. Ranking-based filter methods are simple, fast, and efficient in selecting only the relevant variables [4].

The current work focuses on feature selection dealing with regression. Both model-based and model-free approaches are valid. Yet, in real-life problems, the true nature of a problem is most probably unknown. Thus, making assumptions about the model may hinder the prediction efficiency. Thus model-free approaches are preferable in these problems [5]. Both model-based and model-free methods require selecting relevant variables, especially in high dimensionality. Various model-free variable selection approaches exist including regression trees [6], Support Vector Machines [7], neural networks [8] and neighborhood-based methods as the k-nearest neighbors (k-NN) [9], false nearest neighbors (FNN) [10], trustworthiness and continuity (T&C) [11] as well as distance correlation

(DC) [12].

Model-free and non-parametric algorithms are an excellent approach to feature selection, as no costly model evaluation is required. One way is to examine local neighborhoods, which may describe the model without parameters. k -nearest neighbors (k -NN) is a well-known classification/regression technique that requires the Euclidean distance of a point to a k number of neighbors. k -NN performs well in a wide area of applications. In industrial time series modeling, a notable usage of k -NN is forecasting customer demands with fast and accurate local linear models [13]. Another feat is the estimation of the capacity of lithium-ion batteries that is predicted by a data-driven approach [14]. The k -NN model extracts the dependency of five features (charge capacities and voltages) on the capacity of the battery. Variable selection is also possible with the help of k -NN [15, 16]. Various methods include such approach, *e.g.* activity recognition in smart homes [17] and texture retrieval [18]. Moreover, neural activity can be modeled with the nearest neighbors method, which only finds the first neighbor [19]. k -NN has been researched as a potential feature selection algorithm. k -NN classification have been employed to accelerate the feature selection of wrapper method, reducing the overall time required based on a classifier distance matrix [20]. The Sequential Random k -nearest neighbors (SR k -NN) algorithm [21] is an iterative algorithm to select features based on majority voting of a nearest neighbors classifier. A distance and attribute weighted k -NN-based algorithm have been designed for feature selection [22]. Another method selects features based on similarity measures for k -NN, such as Euclidean distance, correlation distance, and cosine similarity [23]. The performance of k -NN -based feature selections have been examined, proposing that the randomized k -NN-based method yielded the highest accuracy results [24]. The choice of k has always been a question, as the same number of neighbors in one problem may not be sufficient in others; therefore, it requires careful tuning.

A special one neighbor case of k -NN, namely the False nearest neighbors (FNN) method, compares a point with its closest neighbor, and the ratio of the distances in the independent and dependent variables determines whether it is a false neighbor [10]. The number of false neighbors measures how well the variables correlate. The capability of the method for generating local models has already been proven [25]. FNN is employed in the selection of the appropriate embedding dimension [26], the detection of determinism [27] and variable selection. The FNN is considered a method capable of supporting feature selection due to its determinism and accurate selection of embedding dimensions, [28]. Practical applications include bearing fault diagnosis, where the FNN selects parameters that can be indicative of dysfunctional operation [29]. Batteries have been analyzed by determining the minimum embedding dimensions, which are forwarded to a hybrid neu-

ral network to calculate the remaining lifetime [30]. Moreover, FNN has been used in near-infrared spectroscopy to select characteristic wavelength [31]. It is important to note that k-NN and FNN are non-parametric methods. These are great for solving problems with nonlinear correlation. As FNN is a special case of k-NN, it is also capable of feature selection; A mixture of FNN and Supervised Locality Preserving Projection (SLPP) have been employed for feature selection, where FNN eliminated the weak features based on a false neighborhood ratio [32].

Another neighbor-based method is the trustworthiness and continuity (T&C) that plays an essential role in determining one-way correlation [11]. Trustworthiness qualifies the projection of the independent variables against the dependent variables by establishing the local neighborhood in the dependent variable and finding which neighbors cannot be considered the same in the independent variables. Continuity can be regarded as the measurement of the projection of the dependent variable against the independent variables. Therefore the cause-effect relationship between the variables can be detected. Other works featuring the T&C for feature selection are unknown to the authors.

This article aims to generalize the neighborhood-based methods and show the relationship between k-NN, FNN, and T&C. I provide a novel method that generalizes the neighborhood-based methods. The novel approach is for robust use that works even in nonlinear relationships.

The Neighborhood ranking-based feature selection (NRFS) method has been engineered for the above-mention goal. The NRFS ranks the distance matrices of both variables and ranks accordingly. The global variation takes the ranking differences of the matrices and sums them up in the Sum of Ranking Differences technique [33–35]. Local variations also consider a one-way nearest neighbor rank evaluation similar to the T&C. The connection between the two becomes evident, though the global variation of NRFS may fail to evaluate nonlinear correlation.

Inspired by the aforementioned methods, I developed a feature selection method for model-free variable selection based on local neighborhoods and ranking differences. The correct combination of variables can be selected by measuring the nonlinear correlation between the independent and dependent variables. I use the methods as cost functions to optimizer algorithms such as brute force, forward selection [36] and genetic algorithms [37].

The definition of neighborhoods derives from the ranked distance matrix of a variable. Both the independent and dependent variables have their neighborhoods, as their ranking difference may show their relationship. If the ranking difference of the points in the neighborhoods is high, then the relationship may be false or non-existent. The Ranking difference of local neighborhoods may also evaluate nonlinear correlation. Therefore it is vital to analyze the usage of neighborhoods for feature selection. nonlinear correlation

may also be evaluated through the distance correlation (DC) method [12]. Similar to the neighborhood-based methods, DC requires the distance matrices of both sets of variables. Yet, the significant difference is that ranking is not employed. The technique establishes correlation based on the Euclidean distance of the variables, whose distance covariance is divided by the product of the distance standard deviation. DC has been successfully employed in feature selection, where DC feature selection performed well on synthetic datasets compared to other methods [38]. The method has been tried on microarray classification problems providing accurate models for class prediction [39].

The contributions of this work can be defined as:

- I define the FNN as the special case of k-NN, where the number of neighbors is defined as one. With the employment of leave-one-out validation, I can determine a threshold value that can be used to measure nonlinear correlation robustly without model specificity (Section 2.2).
- I determine FNN as the special case to T&C, where the pairwise connection between two variables is measured. The methods present a robust approach to exploring causality between variables (Section 2.3).
- I explore the correctness of the novel method correlation by validation against the DC and neural networks (Chapter 3) .
- I propose a model-free feature selection method based on the FNN and rank correlation of local groups. The robust novel method can select the relevant variables and determines the correlation or the causality of these variables (Section 2.5).
- I employ brute force, forward selection, and genetic algorithms to select the correct variation of the features by incorporating the methods as cost functions into the optimization (Section 2.7).
- I benchmark on dynamic modeling example, proposing that the methods are capable of selecting the order of the model. I also provide simple and widely used datasets, such as the Friedman dataset [40] (Section 3.3).
- I introduce the theoretical background for the False k-th nearest neighbors (Fk-NN) technique, which is to generalize the False nearest neighbors for the k nearest neighbors (Section 2.3).

This work is to present the neighborhood ranking-based feature selection (NRFS) method. Chapter 2 debates the theoretical background of the neighbor-based methods. I first introduce the theoretical background of the neighborhood-based methods in Section

2.1. I define the k-NN algorithm in section 2.2 and its special case, the FNN in Section 2.3. Then T&C is described in Section 2.4, before the defining the generalized NRFS method (Section 2.5). Section (2.6) for the connection between the methods is also included, where the distance correlation is introduced to establish the novel method as its special case. The methods are applied to three datasets: Simple Linear (3.1), monotonous and nonlinear equations, the Friedman dataset (3.2), in which I describe the optimal amount of neighbors for the dataset, and use forward selection as well as genetic algorithms to optimize the feature selection. Another application of the algorithms, namely the identification of dynamic models, is described (3.3). Then I build a neural network to validate the results. Chapter 4 provides remarks on the methods and concludes the paper.

Chapter 2

Neighborhood ranking-based feature selection

2.1 The theory behind the neighborhood-based approach

Models assume a functional relationship $y_i = f(\mathbf{x}_i)$, where the values of n number of independent variables ($\mathbf{x}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n}]$) are capable of predicting the output y_i . Even though models can predict more than one dependent variable, for the sake of simplicity, this work only considers one output. Yet, if variables that provide no substantive contribution to the output are incorporated, the error in the prediction skyrockets. For this reason, selecting the proper features is an essential step to determine a good model. Model-free machine learning methods perform the same as models but without optimizing parameters or establishing the mathematical background of the model. As such, the model-free algorithms need training data to work, while models can predict the output after being trained without the need for other points. Neighborhood-based methods are a subset of model-free learning. The methods establish a relationship between the independent and dependent variables through their Euclidean (L^2 -norm) distances of the m th $\mathbf{x}_m$ points with regards to the i th points $\mathbf{x}_i$. The feature selection problem can be formalized with of n number of independent input variables each with N observations $\mathbf{X}_{N \times n}$. Models estimate the dependent (output) variables dependent variable $\mathbf{y} = [y_1, y_2, \dots, y_N]$.

$$d(\mathbf{x}_i, \mathbf{x}_m) = \|\mathbf{x}_i - \mathbf{x}_m\|_2; \quad i = 1, \dots, N; \quad m = 1, \dots, N, \quad (2.1)$$

where $d()$ denotes the Euclidean distance (L^2 -norm) between the two points, m represents the running index of datapoints whose distances are calculated against the i th point stands for the index of the reference point. A distance matrix can be constructed to represent the distances of each point against the others. The distance matrices of the independent

and dependent variables ($\mathbf{D}^x$ and $\mathbf{D}^y$) are symmetric:

$$\mathbf{D}^x = \begin{bmatrix} d(\mathbf{x}_1, \mathbf{x}_1) & \dots & d(\mathbf{x}_N, \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ d(\mathbf{x}_1, \mathbf{x}_N) & \dots & d(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \quad (2.2)$$

$$\mathbf{D}^y = \begin{bmatrix} d(y_1, y_1) & \dots & d(y_N, y_1) \\ \vdots & \ddots & \vdots \\ d(y_1, y_N) & \dots & d(y_N, y_N) \end{bmatrix} \quad (2.3)$$

According to the i th row or column, the neighbor rank can be defined based on the Euclidean distances. The rank determines which neighbor the m th point is concerning the i th point.

$$r_{i,m}^x = \text{rank}_i(d(\mathbf{x}_i, \mathbf{x}_m)), \quad r_{i,m}^x \neq r_{m,i}^x \quad (2.4)$$

where $r_{i,m}^x$ is the rank value of the m th point in relation to the i th point according to the ranking function ($\text{rank}_i()$). As the distance matrix is symmetric, the column and row ranking function yield the same ranks in the same combination. Out of the determined ranks, a rank matrix can be constructed similarly to the distance matrix ($\mathbf{D}$). The ranking matrix is the ordinality of the distance matrix and inherits its symmetricity. In this work, the ranking function ranks according to columns.

$$\mathbf{R}^x = \begin{bmatrix} r_{1,1}^x & \dots & r_{N,1}^x \\ \vdots & \ddots & \vdots \\ r_{1,N}^x & \dots & r_{N,N}^x \end{bmatrix} \quad (2.5)$$

where $\mathbf{R}^x$ denotes the ranking matrix.

As such, the m th data can be considered the j th closest neighbor of the i th point if:

$$i_x(j) = \left\{ m \in \{1, \dots, N\} \mid r_{i,m}^x = j \right\}; \quad j = 1, \dots, N; \quad m \neq i \quad (2.6)$$

where $i_x(j)$ denotes the j th neighbor of the i th point in the independent variables.

It is important to note that the ranking function is asymmetric. The i th neighbor of the j th neighbor will not result in the same rank as the j th neighbor of the i th point, only in some random cases. If the rank of the distance is less than or equal to k , then the point $i_x(j)$ is a k th neighbor of the sample i . Thus the set of the indices of neighbors $S_k(\mathbf{x}_i)$ can be defined:

$$S_k(\mathbf{x}_i) = \left\{ i_x(j) \in \{1, \dots, N\} \left| r_{i, i_x(j)}^x \leq k \right. \right\} \quad (2.7)$$

where $S_k(\mathbf{x}_i)$ denotes the set of k th closest neighbors according to the i th point.

Neighbor-based methods use the nearest neighbors for regression or correlation measurement, one example is the k -nearest neighbors (k -NN) algorithm [9].

2.2 k -nearest neighbors with leave-one-out regression

In k -Nearest Neighbors (k -NN)-based lazy regression with leave-one-out validation, for a given input vector $\mathbf{x}_i$, its output y_i is estimated through its Euclidean distance to k number of neighbors [9].

The predicted output $\hat{y}_i$ at $\mathbf{x}_i$ can be calculated by taking the mean of the outputs of the neighbors:

$$\hat{y}_i = \frac{1}{k} \sum_{i_x(j) \in S_k(\mathbf{x}_i)} w_{i_x(j)} y_{i_x(j)} \quad (2.8)$$

where $\hat{y}_i$ denotes the predicted dependent variable value, $y_{i_x(j)}$ represents the j th neighbor of the i th point in the independent variable. $w_{i_x(j)}$ stands for the weight of $y_{i_x(j)}$.

The principle behind k -NN is illustrated in Figure 2.1. The algorithm calculates the Euclidean distance in the independent variables, finds the k th closest neighbors with the least distance, and takes the mean of their values in the dependent variable.

The neighbors can be assigned either an equal $1/k$ weight, or their similarity to the selected value [41]:

$$w_{i_x(j)} = \frac{\frac{1}{d(\mathbf{x}_i, \mathbf{x}_{i_x(j)})}}{\sum_{i_x(j) \in S_k(\mathbf{x}_i)} \frac{1}{d(\mathbf{x}_i, \mathbf{x}_{i_x(j)})}} \quad (2.9)$$

The squared error value can be calculated as follows:

$$e_i = \|y_i - \hat{y}_i\|_2 = \left\| y_i - \frac{1}{k} \sum_{i_x(j) \in S_k(\mathbf{x}_i)} w_{i_x(j)} y_{i_x(j)} \right\|_2 \left| i_x(j) \in S_k(\mathbf{x}_i) \right. \quad (2.10)$$

where e_i is the squared error of the predicted $\hat{y}_i$ and the original y_i data.

The mean squared error value can be calculated as:

$$\epsilon = \frac{1}{N} \sum_{i=1}^N e_i \quad (2.11)$$

where ϵ is the mean squared error of the predicted and original data.

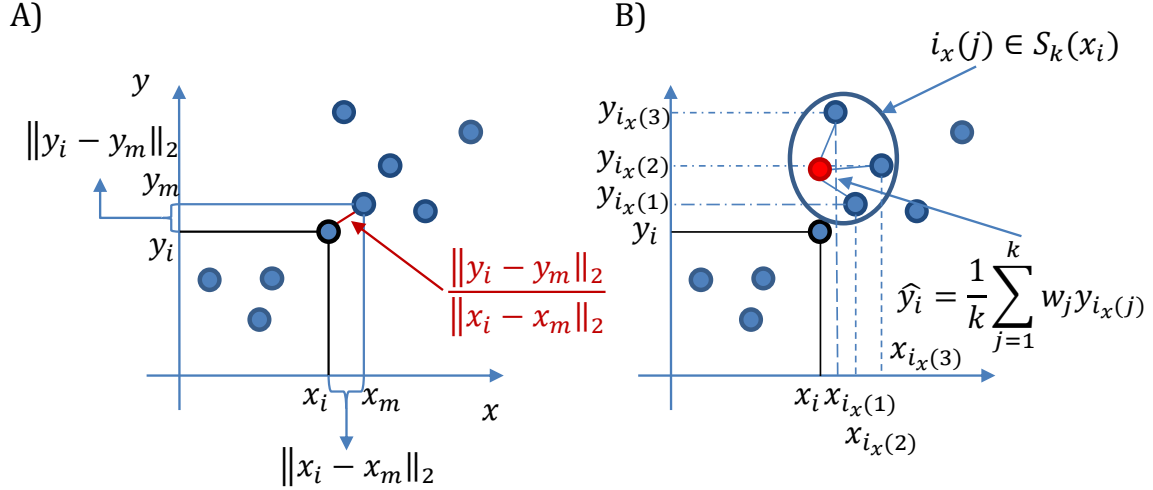


Figure 2.1: **Graphical illustration of the Euclidean distance and k-NN algorithm.** A) subfigure depicts the Euclidean distance between two points, while k-NN (B) evaluates the dependent variable based on the values of the closest (distance) neighbors in the independent variables.

k-NN is sensitive to features. If the set contains several irrelevant features, then the error may drastically increase. Thus it can be employed for feature selection; the less ϵ is, the better the current combination of features.

2.3 The false nearest neighbors method

The False nearest neighbor (FNN) technique was created to determine the minimum embedding dimension of models [10]. The method was later employed for the successful analysis of the input and output of models [26]. The FNN method shares a connection with the k-NN method. If the number of neighbors is selected to be one ($k = 1$), then the predicted value of $\mathbf{x}_i$ is the value of its first neighbor $\hat{y}_i = y_{i_x(1)}$, where $i_x(1)$ is defined as the closest neighbor.

Let us suppose that there is a connection between the independent and dependent variables, which can be modeled as [42]:

$$y_i - y_{i_x(1)} \approx \sum_{l=1}^n \frac{\partial f}{\partial x_{i,l}} (x_{i,l} - x_{i_x(1),l}) \quad (2.12)$$

where $y_i - y_{i_x(1)}$ denotes the change in the dependent variable, $x_{i,l}$ represent the i th observation of the l th variable, $\frac{\partial f}{\partial x_{i,l}}$ stands for the partial derivative of the underlying model in other words, $y_i - y_{i_x(1)}$ is approximated as the first order Taylor series of model $f()$. Thus, FNN is based on the linearization of $f()$ around the point, including the first neighbor $i_x(1)$.

With the help of the Cauchy-Schwarz inequality, Eq. 2.12 can be reorganized into an equality:

$$|y_i - y_{i_x(1)}| \leq \left\| \frac{\partial f}{\partial \mathbf{x}_i} \right\|_2 \|\mathbf{x} - \mathbf{x}_{i(1)}\|_2 \quad (2.13)$$

Which can be rearranged into:

$$\frac{|y_i - y_{i_x(1)}|}{\|\mathbf{x}_i - \mathbf{x}_{i(1)}\|_2} \leq \left\| \frac{\partial f}{\partial \mathbf{x}_i} \right\|_2 = \alpha \quad (2.14)$$

If this inequality is true, the nearest neighbor is a good neighbor. The selection of α is both an essential and gruesome task, as it is impossible to select a robust threshold for all databases. The threshold value directly influences the determination of false neighbors (neighbors that are not good neighbors); thus, overestimating it may deteriorate the accuracy of the method. The threshold may be predicted by maximizing the Jacobian matrix [43]:

$$\alpha = \max \left(\left\| \frac{\partial f}{\partial \mathbf{x}_i} \right\|_2 \right) \quad (2.15)$$

The FNN algorithm examines the ratio of the value of the closest neighbor in the independent variable and their corresponding dependent variable value. The more false neighbors are present, the worse the relationship between the variables is. The set of points/samples with false neighbors can be defined as in Eq. 2.16:

$$F_1(\mathbf{x}_i, y_i) = \left\{ \frac{\|y_i - y_{i_x(1)}\|_2}{\|\mathbf{x}_i - \mathbf{x}_{i_x(1)}\|_2} > \alpha \mid i_x(1) \in S_1(\mathbf{x}_i) \right\} \quad (2.16)$$

$$(2.17)$$

where the set $F_1(\mathbf{x}_i)$ denotes whether the nearest neighbor is located in the false neighbor set.

The inequality determines what kind of neighbor a point has. Neighborhoods are asymmetric as a neighbor in one variable may not have the same rank in the other, which may indicate false relationship. Percentage of false neighbors to all determines the validity of the model:

$$\text{FNN} = 100 \frac{1}{N} \sum_{i=1}^N |F_1(\mathbf{x}_i, y_i)| \quad (2.18)$$

where the length of $F_1(\mathbf{x}_i)$ can only be zero or one.

FNN is based on selecting those pairs whose Euclidean distance in both dimensions are

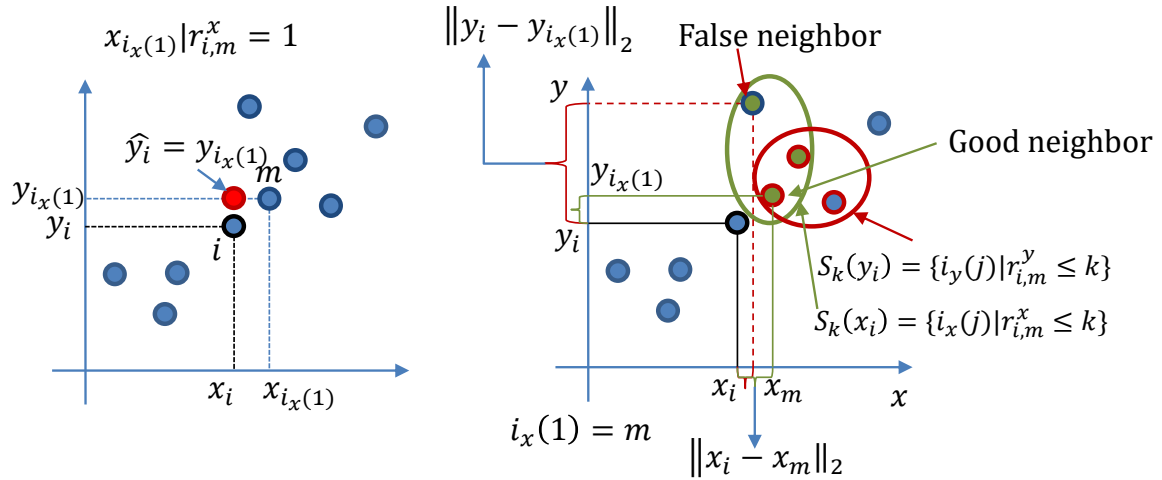


Figure 2.2: **Graphical illustration of the nearest neighbors and false nearest neighbors methods.** A) subfigure depicts the nearest neighbors method, where the closest neighbor determines the dependent variable value. The FNN (B) algorithm evaluates the relationship of two variables by individually iterating through each point in the variables determining which neighbors in one set of variables are neighbors in the others. The green and red circles depict the neighborhood in one variable (similarly illustrated with the red edges for the independent, green filling for the dependent). If a neighbor can only be found in one, it is only a false neighbor. However, if a point is located in both sets, then it is considered a good neighbor (both red edges and green filling).

within an acceptable value against each other and the threshold value. Figure 2.2 depicts the nearest neighbor and the false nearest neighbor algorithms. Subfigure A) illustrates the k -NN algorithm with $k = 1$. Here leave-one-out cross validation is performed to evaluate the correctness of the predicted data. In the FNN algorithm however, the closest neighbor is evaluated and determined whether it is located in the good neighbors set or the false neighbors set. A neighbor is considered a good neighbor if the m th point is the neighbor of the i th point in both independent and dependent variables $\{i(j) \in GNN(\mathbf{x}_i) | i_x(j) \in S_1(\mathbf{x}_i) \wedge i_y(1) \in S_1(y_i)\}$ (depicted by green and red circles), however one-way false neighbors may appear $\{i(1) \in FNN(\mathbf{x}_i) | i_x(1) \in S_1(\mathbf{x}_i) \oplus i_y(1) \in S_1(y_i)\}$ (green or red circles). As such, points that are not neighbors with the i th point can also be defined $\{i_x(1) \notin S_1(\mathbf{x}_i) \wedge i_y(1) \notin S_1(y_i)\}$ (blue circles).

I also generalized the FNN method. If the number of neighbors increases, then the FNN can be modified for more than one neighbor, essentially a false k th nearest neighbor (Fk-NN). For one point with k member neighborhood:

$$F_k(\mathbf{x}_i, y_i) = \left\{ \frac{\|y_i - y_{i_x(j)}\|_2}{\|\mathbf{x}_i - \mathbf{x}_{i_x(j)}\|_2} > \alpha \mid i_x(j) \in S_k(\mathbf{x}_i) \right\} \quad (2.19)$$

where the set $F_k(\mathbf{x}_i)$ denotes the the set of false neighbors in a k member neighborhood.

Then the $F_k(\mathbf{x}_i)$ sets are calculated for each point, and their normalized length is taken its mean:

$$\text{Fk-NN} = 100 \frac{1}{N} \sum_{i=1}^N \frac{1}{k} |F_k(\mathbf{x}_i, y_i)| \quad (2.20)$$

The Fk-NN algorithm may be more robust than the FNN algorithm, as the underlying model is trained on local neighborhoods rather than the nearest neighbor.

2.4 Trustworthiness and continuity

Trustworthiness and continuity (T&C) is another neighbor-based algorithm that measures the accuracy of the projection from an independent to a dependent variable, and *vice versa* [11]. Trustworthiness measures the accuracy of the projection from the independent $\mathbf{x}_i$ to the dependent y_i variables. If the neighborhood of y_i is established and their corresponding values are not included in the neighborhood of $\mathbf{x}_i$, then it is untrustworthy.

It counts the k amount of neighbors that are not neighbors of the i th point in the other variables. T&C considers the degree of overlapping between the neighborhoods in both ways, providing a metric of similarity of neighborhoods in each direction. False nearest neighbors is a special case of continuity, where the ratio of the values of the variables is examined. In this subsection, $i_y(j)$ denotes the indices of points that are the first k th neighbor of the i th point in the independent variables, whereas $i_x(j)$ stands for the indices of points that are the first k th neighbor of the i th point in the independent variables.

If the rank in the independent variables is denoted by $r_{i,i_x(j)}^x \leq k$, then the $\mathbf{x}_{i_x(j)}$ is the j th neighbor of $\mathbf{x}_i$. If $\mathbf{x}_{i_y(j)}$ cannot be located in the k neighborhood of $\mathbf{x}_i$, but $y_{i_y(j)}$ is in the k neighborhood of y_i , then a set of untrustworthy neighbors can be defined.

$$U_k(y_i, \mathbf{x}_i) = \left\{ i_y(j) \in \{1, \dots, N\} \mid r_{i,i_y(j)}^y \leq k \wedge r_{i,i_y(j)}^x > k \right\} \quad (2.21)$$

where $U_k(\mathbf{x}_i)$ denotes the set where the k number of neighbors of y_i are not considered the k th neighbors of $\mathbf{x}_i$.

The ranks of non-neighbor values can be averaged and scaled to the $[0,1]$ intervals, which results in the trustworthiness measure:

$$T_k = 1 - \frac{2}{Nk(2N - 3k - 1)} \sum_{i=1}^N \sum_{i_y(j) \in U_k(y_i, \mathbf{x}_i)} (r_{i,i_y(j)}^x - k) \quad (2.22)$$

where T_k is the trustworthiness of the model based on k th neighbor local models, N is

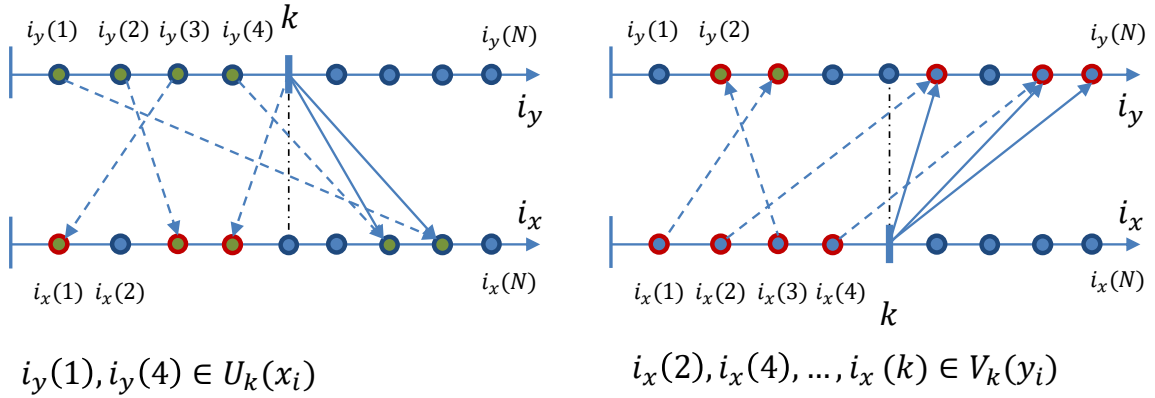


Figure 2.3: **Graphical illustration of trustworthiness and continuity methods.** The left-hand side subfigure depicts the trustworthiness measure. The indices are sorted according to the closest neighbors in both variables, and the corresponding ranks are examined in the independent ones. Continuity is the opposite of trustworthiness. The right-hand side of the figure depicts continuity. Moreover, points with a green filling or red edges are false neighbors, while having both is considered a sign of being a good neighbor. T&C can be calculated as in Eq. 2.22 and 2.24, respectively.

the number of observations, $U_k(\mathbf{x}_i)$ is the set of indices of neighbors in the dependent variable that are not the k th neighbors in the independent variables. Trustworthiness is also scaled to $0 \leq T_k \leq 1$ by $2/Nk(2N - 3k - 1)$ scaling coefficient.

The continuity measures the validity of the projected points, where the k neighbors $\mathbf{x}_{i_x(j)}$ may be located in the k neighborhood of $\mathbf{x}_i$, however $r_{i,i_x(j)}^y$ is out of the predefined range of k :

$$V_k(\mathbf{x}_i, y_i) = \left\{ i_x(j) \in \{1, \dots, N\} \left| r_{i,i_x(j)}^x \leq k \wedge r_{i,i_x(j)}^y > k \right. \right\} \quad (2.23)$$

where $V_k(y_i)$ is the set where the neighbors of $\mathbf{x}_i$ are not neighbors of y_i

The continuity measure is formalized as:

$$C_k = 1 - \frac{2}{Nk(2N - 3k - 1)} \sum_{i=1}^N \sum_{i_x(j) \in V_k(\mathbf{x}_i, y_i)} (r_{i,i_x(j)}^y - k) \quad (2.24)$$

where C_k is the continuity of the model based on k th neighbor local models, $U_k(y_i)$ is the set of neighbors that are not the k th neighbor in the dependent variable. Continuity is scaled to the range of $0 \leq C_k \leq 1$ by applying the $2/Nk(2N - 3k - 1)$ scaling coefficient.

Figure 2.3 depicts the basic operation of trustworthiness and continuity measures. The upper axis denotes the indices of the closest neighbors in the dependent variable. In

contrast, the lower axis does the same, but for the independent variables. Each measure provides a one-directional projection measurement through rank differences evaluated through a k nearest neighbors local model, where only the false neighbors are taken into account. The indices that are denoted by points filled with green are part of the neighborhood in the dependent variables. At the same time, the red edge indicates the membership in the local neighborhood in the independent variables. The ones with both marks are good neighbors, and the ones with only one are considered false neighbors, whose ranks are adjusted to their distance to the neighborhood. On the left-hand side, the trustworthiness measure is illustrated, which sums up the distance of the rank of the false neighbor to the local neighborhood in the independent variables, and *vice versa* with continuity, but to the other direction. It is only reasonable to use the continuity measure for feature selection as it evaluates the goodness of the projection of the first k neighbors in the independent variable to the dependent variable.

2.5 Neighbourhood ranking-based feature selection

The novel method calculates the rank difference of the neighbors in the independent and dependent variables. When a combination of variables is selected, the Euclidean Distance is calculated before being ranked. As an iterative process, the method calculates the ranking of the distances to each point, resulting in an $\mathbf{R}_{N \times N}^x$ matrix. Similarly, the ranking is also established with the corresponding points of the dependent variable, yielding a ranking matrix $\mathbf{R}_{N \times N}^y$. Then the two matrices are subtracted and taken their mean. It is crucial to scale the ranking differences to the zero and one interval, as the summation may lead to incomprehensible numbers. The method resembles the sum of ranking differences. The theoretical maximum of SRD is used as a scaling coefficient. It is only reasonable that the ranking difference of the same vectors yields zero, whereas the maximum depends on the number of observations [34]:

$$\text{SRD}_{\max} = \begin{cases} 2 \sum_{s=1}^{\frac{N}{2}} (2s - 1) = 2 \left(\frac{N}{2}\right)^2 & \text{if } N \text{ is even} \\ 2 \sum_{s=1}^{\frac{N}{2}} 2s = 2 \frac{N}{2} \left(\frac{N}{2} + 1\right) & \text{if } N \text{ is odd} \end{cases} \quad (2.25)$$

where s is a rank. The theoretical maximum obtains its value by subtracting two opposite (in direction) rankings that facilitate N ranks. Generally, only half of the subtraction is required as ranking difference are symmetric, only the half of the ranking distances is calculated. As an example, the ranking distance of $[1, 2, 3, 4]$ and $[4, 3, 2, 1]$ is $[-3, -1, 1, 3]$. This proposes that the ranking difference between two corresponding ranks is $(2s - 1)$, where s is index of the ranks. A similar rule can be defined if the number of observations is odd. Here the ranking difference of $[1, 2, 3, 4, 5]$ and $[5, 4, 3, 2, 1]$ that is $[-4, -2, 0, 2, 4]$

which can be generalized as (2s). The rules only apply for one half of the set, thus it is required to be multiplied by 2. If N is even, then the sum of differences return $2(N/2)^2$ whereas if N is odd, then the ranking difference maximum becomes $2N/2 (N/2 + 1)$. Please see [34] for more information.

To evaluate the relationship between the independent and dependent variables, the neighbor ranks of the i th point in both set of variables is subtracted, which is performed for each distance vector:

$$\rho = 1 - \frac{\sum_{i=1}^N \sum_{m=1}^N (|r_{i,m}^y - r_{i,m}^x|)}{N \text{SRD}_{\max}} \quad (2.26)$$

where ρ denotes the correlation of the independent and dependent variables, $r_{i,m}^x$ stands for the rank of the m th point against the i th datapoint in the independent variables in $\mathbf{R}^x$. $r_{i,m}^y$ denotes the rank of the m th point against the i th datapoint according to the dependent variables. The ρ value can also be scaled to the $[0, 1]$ interval with the maximum SRD coefficient calculated in Eq. 2.25.

The methods can also be used with local neighborhoods subtracted from each other, which may be similarly selected to T&C. The first k ranked point is selected in one set of variables, and the corresponding points in the other is subtracted. With k number of neighbors, the set of indices is established:

$$i_x(j) \in I^x(r_{i,m}^x < k) \quad (2.27)$$

$$i_y(j) \in I^y(r_{i,m}^y < k) \quad (2.28)$$

where $i_x(j)$ denote the indices of the k nearest neighbors in the independent variables, and $m_y(j)$ stand for the indices of the nearest neighbors in the dependent variables.

where $m(j)$ is the index of the j neighbor according to the dependent variable, I^x is the set of k closest neighbors in relation to the independent variables. The sets incorporate all nearest neighbors indices. Thus the local neighborhood-ranking based feature selection can be defined:

$$\rho_k^x = 1 - \frac{2 \sum_{m=1}^N \sum_{i_x(j) \in I^x} (|r_{i_x(j),i}^x - r_{i_y(j),i}^y|)}{Nk(2N - 3k - 1)} \quad (2.29)$$

$$\rho_k^y = 1 - \frac{2 \sum_{i=1}^N \sum_{i_y(j) \in I^y} (|r_{i_y(j),i}^y - r_{i_x(j),i}^x|)}{Nk(2N - 3k - 1)} \quad (2.30)$$

where ρ_k^x denotes the local variation of the neighborhood ranking-based feature selection algorithm in terms of the independent variables, while ρ_k^y stand for the local variation

of the neighborhood ranking-based feature selection algorithm in terms of the dependent variables

The method itself is a generalization of T&C as it provides information on the difference of the neighborhood in each set of variables.

2.6 Discussion on the similarities of the measures

Throughout the paper, the relationship between the methods is stated. This section proves the comments and provides major remarks on the connections.

The most obvious connection is between k-nn and FNN. The error of the nearest neighbor algorithm, where $k = 1$ can be established as:

$$e_i = \|y_i - y_{i_x(1)}\|_2 \quad (2.31)$$

The FNN algorithm is fundamentally based on the nearest neighbors with the exception of only the amount of points, where e_i is more than a threshold value α , matters:

$$e_i > \alpha \| \mathbf{x}_i - \mathbf{x}_{i_x(1)} \|_2 \mid i_x(j) \in S_k(\mathbf{x}_i) \quad (2.32)$$

Therefore the FNN determines whether the error is above the error threshold. The k-NN provides information on model validity by taking the mean squared error value, and the FNN determines the validity of the model similarly.

The FNN is also related to the T&C technique, especially to the continuity method. T&C focuses on the neighbors that are over a threshold that is specified in ordinal numbers. Therefore the continuity measure is the ranking difference variation of the Fk-NN method. Let us define the set of false neighbors:

$$F_k(\mathbf{x}_i, y_i) = \left\{ \frac{\|y_i - y_{i_x(j)}\|_2}{\|\mathbf{x}_i - \mathbf{x}_{i_x(j)}\|_2} > \alpha \mid i_x(j) \in r_{i,i(j)}^x < k \right\} \quad (2.33)$$

The set is established based on the neighborhoods according to the first k ranks of the independent variables. We can also determine the false neighbors if the ranking difference is greater than the neighborhood. In other words, the rank is out of the neighborhood in the dependent variable:

$$r_{i,i_x(j)}^y > k \mid i_x(j) \in \{1, \dots, N\} \quad (2.34)$$

We define the set of indices that are the k th nearest neighbors in the independent variables but are excluded from the neighborhood in the dependent variables:

$$V_k(\mathbf{x}_i, y_i) = \left\{ r_{i, i_x(j)}^x \leq k \wedge r_{i, i_x(j)}^y > k \mid i_x(j) \in \{1, \dots, N\} \right\} \quad (2.35)$$

The ratio of good neighbors can be calculated by one minus the false neighbors:

$$\text{Gk-NN} = 1 - \frac{1}{N} \sum_{i=1}^N \frac{1}{k} |F_k(\mathbf{x}_i)| \quad (2.36)$$

If the set of good neighbors is determined based on ranks, then:

$$\hat{C}_k = 1 - \frac{1}{N} \sum_{i=1}^N \frac{1}{k} |V_k(\mathbf{x}_i, y_i)| \quad (2.37)$$

The content of the set may provide more information on the local neighborhood, therefore the distance of the ranks of the set from the neighborhood are summed up and scaled accordingly:

$$C_k = 1 - \frac{2}{Nk(2N - 3k - 1)} \sum_{i=1}^N \sum_{i_x(j) \in V_k(y_i)} (r_{i, i_x(j)}^y - k) \quad (2.38)$$

Both approaches measure the ratio of good neighbors through false neighbors. While the Gk-NN sums up the length of the set of false neighbors and sums it up for each point, the continuity sums up the ranking distance from the border of the neighborhood or how close is the point to be neighbor in the other variable. This calculates the rank deviation of the false neighbors based on their distance from their neighborhood.

T&C technique is a special case of the novel NRFS case, thus the generalization. The NRFS method considers the ranking difference, but continuity only regards false neighbors:

$$\rho_k^x = 1 - \frac{2 \sum_{m=1}^N \sum_{i_x(j) \in I^x} |r_{i_x(j), i}^y - r_{i_x(j), i}^x|}{Nk(2N - 3k - 1)}, r_{i, i_x(j)}^x \in [1, 2, \dots, k] \quad (2.39)$$

Let us suppose that only the false neighbors are taken into account:

$$i_x(j) \in I^x(r_{i, m}^x \leq k \wedge r_{i, m}^y > k) = i_x(j) \in V_k(y_i) \quad (2.40)$$

Substituting into Eq. 2.39:

$$\hat{\rho}_k^x = 1 - \frac{2 \sum_{m=1}^N \sum_{i_x(j) \in V_k(y_i)} (|r_{i(j), m}^y - r_{i(j), m}^x|)}{Nk(2N - 3k - 1)} \quad (2.41)$$

If the ranking difference is measured from the neighborhood:

$$C_k = 1 - \frac{2 \sum_{m=1}^N \sum_{i_x(j) \in V_k(y_i)} (|r_{i(j), m}^y| - k)}{Nk(2N - 3k - 1)}, r_{i_x(j), m}^x = k \quad (2.42)$$

Thus, the continuity is a special case of the NRFS where the ranking difference from the neighborhood of the false neighbors are calculated.

Moreover, the novel method contains similarities to the distance correlation (DC) technique; thus, first, DC is introduced, and the connection between the NRFS and DC is established.

Distance correlation measures the covariance and the variance between the multivariate variables. The significant difference to classic measures, such as Pearson, is that DC requires the Euclidean distance matrix of the variables to calculate the distance covariance and variance, similarly to what is defined in Eq.2.1. The method can handle more than one independent and dependent variable.

First, the distance correlation takes the double centered distances of the Euclidean distance matrix:

$$a_{i,m} = d_{i,m}^x - \bar{d}_{i,\cdot}^x - \bar{d}_{\cdot,m}^x + \bar{d}^x \quad (2.43)$$

where $a_{i,m}$ denotes the double centered distance of the Euclidean distance of the i th and m th point in the independent variables. $\bar{d}_{i,\cdot}^x$ represents the mean of the i th row of the independent distance matrix, $\bar{d}_{\cdot,m}^x$ stands for the mean of the m th column, $\bar{d}^x$ denotes the grand mean (mean of the means) of the independent distance matrix. Similarly to the distances in the independent variable, it can be done for the dependent as well:

$$b_{i,m} = d_{i,m}^y - \bar{d}_{i,\cdot}^y - \bar{d}_{\cdot,m}^y + \bar{d}^y \quad (2.44)$$

where $b_{i,m}$ stands for the double centered distance of the Euclidean distance in the dependent variable.

The distance correlation is very much the same as the general Pearson correlation with the exception that the (multivariate) Euclidean distance of the input data is examined:

$$dCor = \frac{\sum_{i,m=1}^N a_{i,m} b_{i,m}}{\sqrt{\sum_{i,m=1}^N a_{i,m}^2 \sum_{i,m=1}^N b_{i,m}^2}} \quad (2.45)$$

The NRFS is essentially an instance of the distance correlation. Thus it can be described as a particular case of distance correlation. Let us suppose that the ranks are examined. If the ranking matrix consists of different ranks (assuming that no distance is the same), then Spearman rank correlation based on the generalized correlation can be derived [44]:

$$rdCor = 1 - \frac{6 \sum_{i=1}^N \sum_{m=1}^N r_{i,m}^x - r_{i,m}^y}{N(N^2 - 1)} \quad (2.46)$$

The NRFS only differs from the ranking distance that the absolute value of the differences is taken, thus the scaling coefficient is adjusted:

$$\rho = 1 - \frac{\sum_{i=1}^N \sum_{m=1}^N (|r_{i,m}^x - r_{i,m}^y|)}{NSRD_{\max}} \quad (2.47)$$

where the $SRD_{\max}$ denotes the equations described in Eq. 2.25.

2.7 Application of the proposed metrics

Establishing a prediction model is not always worth it. It is essential to explore the predictability based on the capability of the dataset, as the model may not be able to predict the output when optimized from a dataset. Model-free methods can evaluate whether the dataset provides adequate information and whether the features are relevant to the output. These data-driven solutions may provide the necessary background for establishing models, yet the model-free methods require an optimizer. During simple cases, where the number of variables is significantly small, brute force can be employed to find the correct combination of the crucial features. This is not advised as each feature combination is tested and can be a time-consuming effort if the dataset consists of more than ten features.

A possible optimizer method is the forward selection algorithm which is a straightforward method for feature selection [36]. The method requires two sets, one of which is the set of unused variables, whereas the other contains those who had been selected. The new features is selected depending on which combination of the new and previously selected variables returns the least error. The time complexity of this method is $O(n^2)$, whereas brute force requires $O(\sum_{c=1}^n \binom{n}{c})$.

Performing feature selection with many features may immensely decelerate the search as brute force would require immense time to select the appropriate combination. As a response, heuristic algorithms were implemented to select feature combinations randomly. A well-known example is the genetic algorithm that has widely been used in feature selection [37]. GAs are complex heuristic methods where random individuals try to provide the solution with the least error. The competition goes on for generations (iteration), which may result in selecting the best combination of features. However, the correct combination may not be included in the starting population due to randomization. Thus the time required may drastically be decreased. However, it is important to note that as the GAs are random, the evaluation of their performance should be statistical.

Neural networks are also featured in this work so that the results are validated with a well-known and high-accuracy algorithm. If a valid model can be chosen, one should

not forget that model-driven structure selection is necessary. The neighborhood-based methods require meta parameters to be optimized (k , α), similarly to neural networks (number of hidden layers).

Feature selection plays a vital role in identifying the order of dynamic systems. Here the input is based on an input variable, time-shifted equivalent, and the past outputs. The order of a system defines which time delay is required to describe the system accurately. The inputs can be systematically added to determine the order of the system; brute force can be used.

The next section shows that the methods can describe nonlinear relationships through three simple equations. I also include the Friedman dataset benchmarked with forward selection, GA, and validated by a neural network. Lastly, I determine the order of a simulated polymerization reactor dataset as an example for dynamic system identification.

Chapter 3

Application of the neighborhood ranking-based feature selection

The main goal of this work is to establish a link between the neighborhood-based methods and use them for feature selection. Thus the procedures are performed and benchmarked on different datasets to provide adequate information on their performance. The application potentials are discussed in section 2.7: In the first use-case, I employ the feature selection on a linear, a monotonous, and a nonlinear equation. Then, I benchmark the methods on the Friedman dataset [40] with forward selection, GA, and a neural network. Then I analyze a simulated polymerization reactor to provide an example for dynamic system model order determination [45].

3.1 Linear, monotonous and periodic equations

This subsection discusses didactic examples of feature selection problems. A dataset of four features is generated for each problem with a thousand samples generated uniformly randomly. The following simple equations represent linear, monotonic and nonlinear problems:

$$y = x_1 + x_2 \tag{3.1}$$

$$y = x_1 + \log(x_2^2) \tag{3.2}$$

$$y = \cos(x_1) + \exp(x_2) \tag{3.3}$$

As the number of features is meager, the brute force approach is used with 16 calls for a cost function. I run the k-NN, FNN, continuity, and local variation of NRFS based on the independent variable neighborhood, distance correlation, and rank distance correlation

algorithms on the three datasets. The number of nearest neighbors is selected to be ten for each method. The results are illustrated in Figure 3.1, where the fitness values of the combinations were scaled between zero and one. All approaches accurately solve the feature selection problem for the linear problems. Monotonous problems may be difficult for FNN as the distances change significantly over the independent variable. An another reason may be that the threshold value is difficult to properly determine with sudden changes to the y value. The nonlinear equation is problematic for the ranking and standard distance correlation. The FNN selected the feature combinations where the x_2 member is incorporated. I assume that FNN considers the other two features as noise.

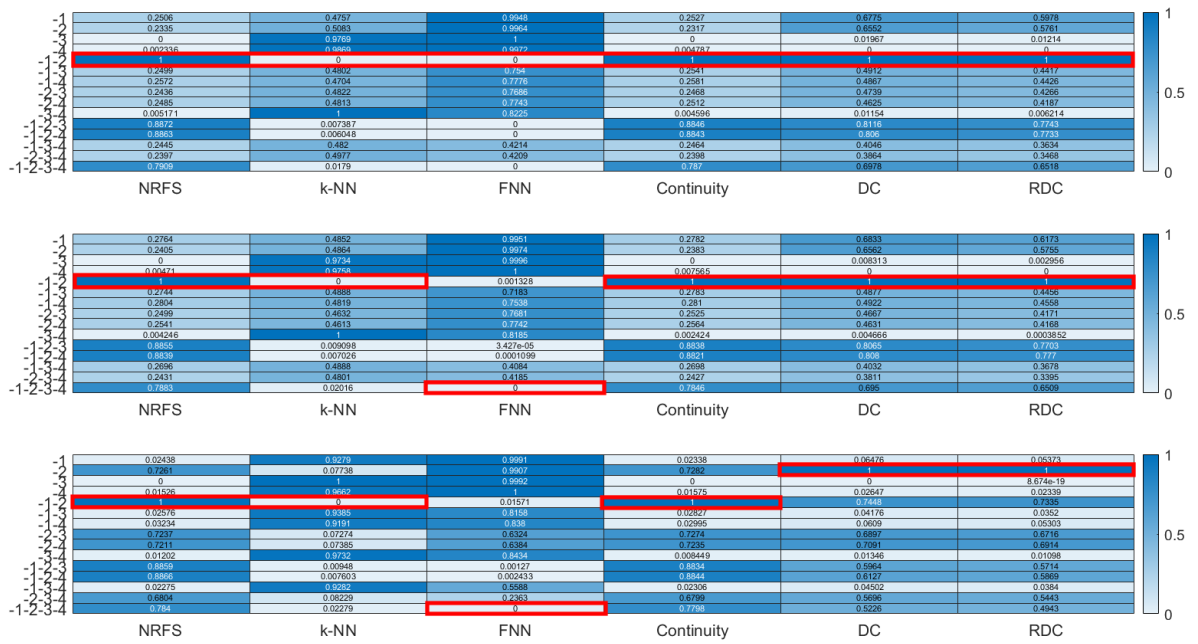


Figure 3.1: **Heatmaps of the brute force feature selection with neighborhood-based methods as cost functions.** The local neighborhood is selected to be 10. The methods can undoubtedly solve the linear and monotonous function, but DC and ranking DC may fail against the nonlinear one. The FNN may have failed due to an incorrect threshold value, which must be optimized for each example. The values are scaled between zero and one so that the colors may match.

3.2 Friedman-1 model

The Friedman dataset is widely used in feature selection as it contains five relevant variables including one with nonlinear correlation and five random, irrelevant features [40]. A thousand random points are generated for each feature uniformly between zero and one.

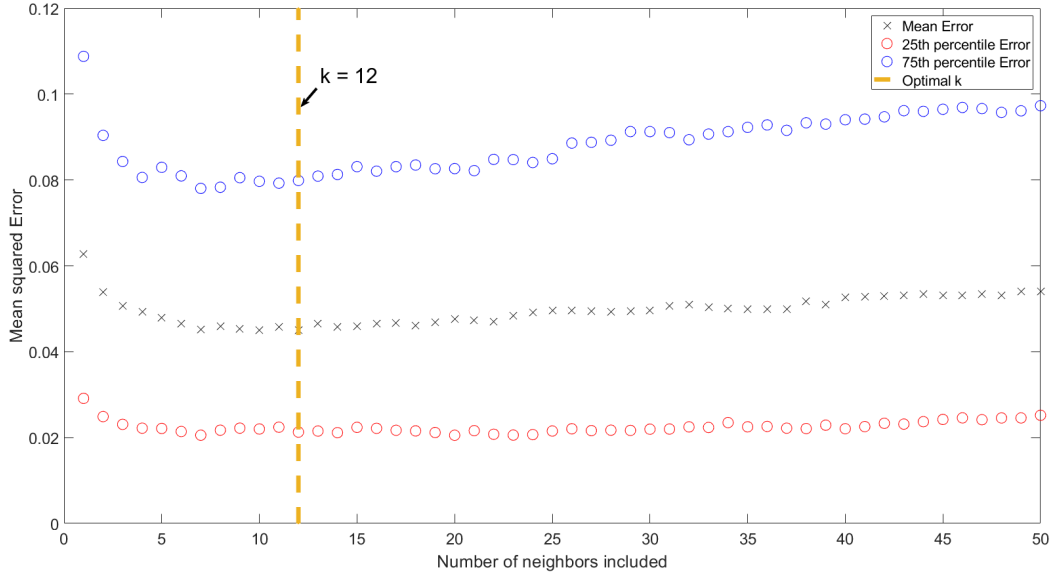


Figure 3.2: **Selection of the optimal amount of neighbors.** The figure shows that 12 is the optimal number as the k-NN has its minimum error at 12.

The dependent variable is calculated the following way:

$$y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \sigma \quad (3.4)$$

Only the first five variables are related. Yet, the optimal number of neighbors must be determined before applying the methods. Therefore I analyzed the behavior of k-NN for one to fifty neighbors, which is illustrated in Figure 3.2. As the database is generated through random means, the number of neighbors may differ with each retry. The random number generator seed was set to the same constant for each experiment. In this example, the 12 nearest neighbors provided the least error. The 75th percentile may be used for the identification of α as the error value is the product of the threshold value and the distance of the independent variable (Eq. 2.32).

I run the neighborhood-based methods as a cost function with forward selection optimizer. Figure 3.3 illustrates the error of the methods and the order in the features were selected:

The neighborhood-based methods each selected the relevant features in the same order. The error values of NRFSX, k-NN, and continuity had their minimum at the fifth inclusion of features that is the right combination of variables. FNN reached its first zero at the fourth inclusion, while the Distance correlations did so with the third. Nonetheless, DC determined the next two features, while rank DC failed with the last. The methods ran for: NRFS - 12.6486; k-NN - 15.3284; FNN - 15.2658; Cont. - 22.8044; DC - 12.3796; RDC 31.9856 secs. The MATLAB implementation of the Rank distance correlation is

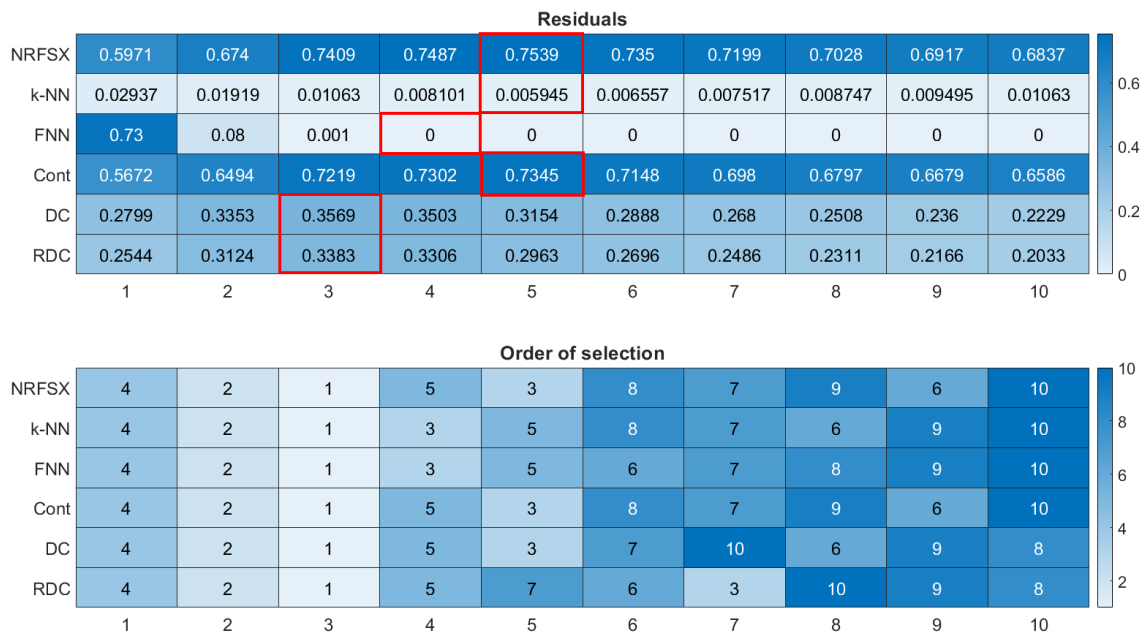


Figure 3.3: **Forward selection with neighborhood-based methods as cost functions.** The upper heatmap provides the fitness values, the bests framed with red rectangles. The lower table show in which order were the features selected.

rather costly. It may fail due to the analysis of the whole of the ranking, which may not be able to represent nonlinear correlation, such shown in subsection 3.1. I assume that the last five random features have no reason in their order of inclusion.

Genetic algorithms are also used as an optimizer, one of the best techniques for feature selection. The method is similar to the forward selection. The optimal number of neighbors is determined before the GA is employed. GAs are heuristic in nature; therefore, there is a chance that the optimal combination of features is not included in the starting population. I iteratively determined the combinations of variables needed to provide the least error and summed them up in Figure 3.4. The starting bitstring population was set to 10, while the algorithm lasted for five generations with a 0.1 chance for mutation. The optimal number of neighbors remained at 12. The tests were performed tenfold for each method.

It is hard to generalize due to the heuristic nature of GAs. In this case, the best method was the continuity method, which only failed to determine one variable once during the ten iterations. The other techniques also provided good results. FNN was seemingly the weakest of each but could decide on the dominant features in more than half of the iterations. Even though DC and RDC provided adequate results, I assume that the GAs can select the features with their help. The time required to run RDC is significant. I also measured the elapsed time for each method and iteration, which can be seen in Table 3.1.

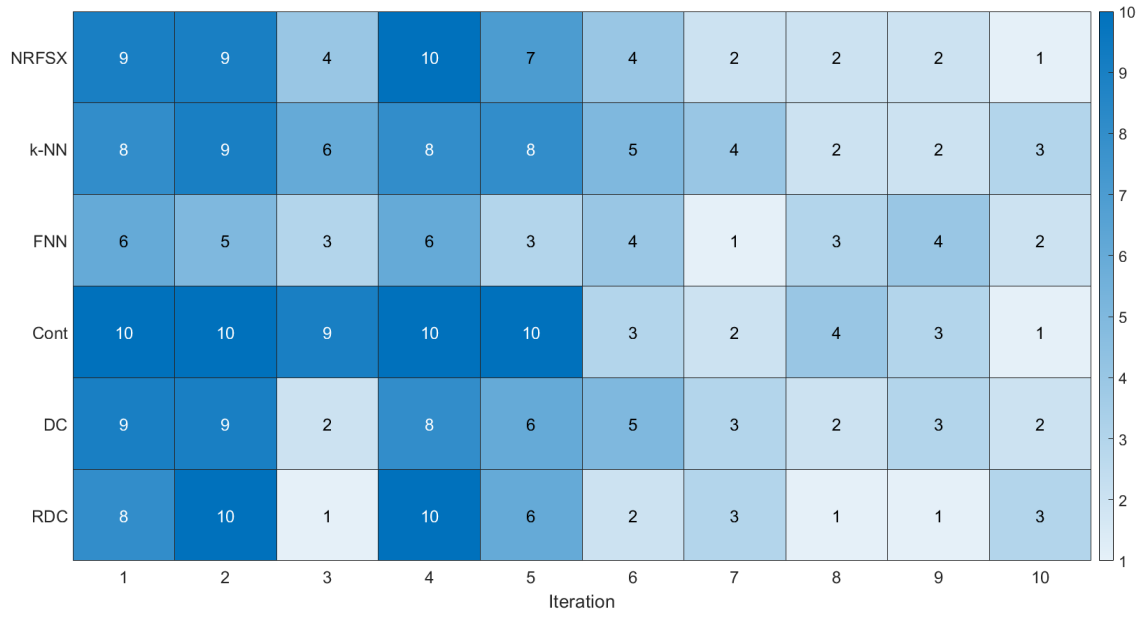


Figure 3.4: **Genetic algorithm with neighborhood based methods as cost functions.** Continuity performed the best, while NRFSX and Distance correlation performed similarly. FNN failed, probably due inefficient optimization of its threshold value.

Table 3.1: **Genetic Algorithm runtime for each method.**

Methods/Iteration	no. 1	no. 2	no. 3	no. 4	no. 5	no. 6	no. 7	no. 8	no. 9	no. 10	Mean
NRFSX	15.1269	14.0988	13.9970	13.8535	14.0849	13.9397	13.8633	14.1935	14.1965	14.1238	14.1478
k-NN	23.3937	22.3667	20.6169	17.9957	17.8209	17.4705	18.5345	18.6658	18.5533	17.4275	19.2845
FNN	16.4284	18.6576	18.1092	19.5296	18.6383	18.3998	18.2965	17.8849	20.5918	18.6404	18.5176
Cont	29.6046	24.6531	24.0867	25.4077	25.9334	25.5706	25.4766	25.6523	25.9177	25.4916	25.7794
DC	16.0677	15.9280	16.6722	16.1874	16.1959	16.1343	16.2133	15.4161	16.2490	15.9270	16.0991
RDC	34.9496	34.6437	34.2235	35.2113	36.0165	34.6389	36.3461	35.3866	33.7893	33.7659	34.8971

Even though continuity provided the best results, its runtime came second to RDC, which failed to recognize two features properly. The two fastest were the Distance correlation and the NRFS, which selected more relevant features with lesser runtime.

I validated the methods against the neural network, with the good features and all features as input data. In Figure 3.5, I illustrate the boxplot of the mean squared error. The neural network performs better if only the selected features are designated as input. If irrelevant features are chosen, however, the error value may increase significantly.

Table 3.2 contains the runtime of the neighborhood-based methods compared to the neural network. In general, the NRFS was the fastest, though Distance correlation was similarly fast and its performance with GAs is similar to NRFS. Yet, it was not able to identify dynamic systems, and its feature selection in nonlinear correlation may not identify the proper combination of variables with forward selection.

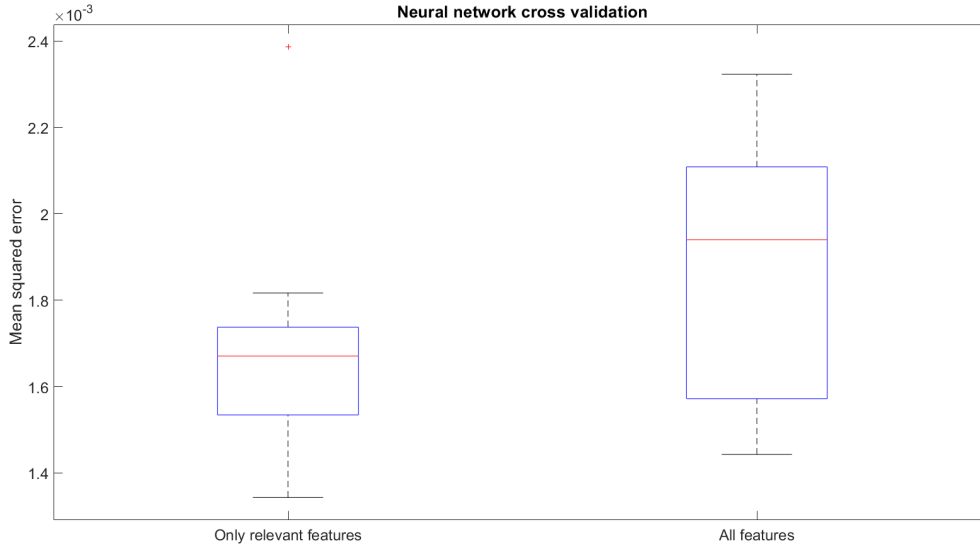


Figure 3.5: **Learning error of the neural network cross validation for the Friedman dataset.** With only the relevant features neural network predicts the output with higher accuracy.

Table 3.2: Runtime performance of a neural network and the neighborhood-based methods. The neural network takes considerably more time than others, while the generalize method provided the best runtime out of the other implementations.

Runtime	Neural network (CV)	NRFSX	k-NN	FNN	Cont.	DC	RDC
1	5.5579	0.3044	0.3158	0.3413	0.4861	0.3126	0.6240
2	2.7891	0.2557	0.3253	0.3100	0.4257	0.2698	0.5802
3	2.6822	0.2600	0.3153	0.3142	0.4209	0.2678	0.5857
Mean	3.6764	<u>0.2733</u>	0.3188	0.3219	0.4443	0.2834	0.5966

3.3 Dynamic modeling - polymerization reactor

Dynamic modelling is common task in process engineering. The identification of dynamic models may help boost the production of a factory, thus was selected as an application example for the neighborhood-based techniques. In process engineering, reactor model identification provides an excellent use-case for order identification. The dynamics inside the reactor can be described with a nonlinear function $f()$:

$$y_i = f(\mathbf{x}_i); i = 1, \dots, N \quad (3.5)$$

where y_i denotes the output, $\mathbf{x}_i$ stands for the input of the model at the i th point.

As the function is nonlinear, the Nonlinear Auto-Regressive models with eXogenous inputs (NARX) is employed for output prediction. This model predicts the output based on the past values of the process input (u_k) and output (y_k), where the model order is defined the required past values for accurate prediction [46]:

$$\mathbf{x}_i = [y_{i-1}, y_{i-2}, \dots, y_{i-n_y}, u_{i-1}, u_{i-2}, \dots, u_{i-n_u}] \quad (3.6)$$

where n_y denotes the order of the output, and n_u stand for the order of the input.

I use the input generated uniformly randomly between 0.005 and 0.015 by a simulation model of a continuous polymerization reactor by [45] that polymerize methyl methacrylate with azobisisobutyronitrile (initiator) and toluene (solvent). A jacketed CSTR houses the reaction that can be analyzed such in Eq. 3.6. For more information on this model please see [47]. Brute force was applied: The first six orders for both input and output was examined with k-NN, FNN, NRFS, continuity, distance and rank distance correlations.

For the FNN, I use ratio tables to determine the model order. Identification works as follows: The first cell with a close-to-zero number closer to the left upper corner determines the order of the model [25]. In contrast, the others are minimized/maximized. The methods, except for DC and RDC have been able to identify the order of the dynamic system, which is illustrated in Figure 3.6. Most were able to identify the system: NRFSX, k-NN, and continuity performed well in finding two-degree input and one-degree output lag. In the FNN, the a small number is delegated to ($n_u = 2, n_y = 1$). DC fails due to the similarity of the variables, which are essentially the same vectors but time-delayed. The input of Rank DC is also a shifted ranking; thus, the highest correlation will be when only the first-order input and output are selected.

Two input delays ($n_u = 2$), one output delay ($n_y = 1$) have also been identified for the polymerization reactor in [25], the methods seem to be able to identify model order.

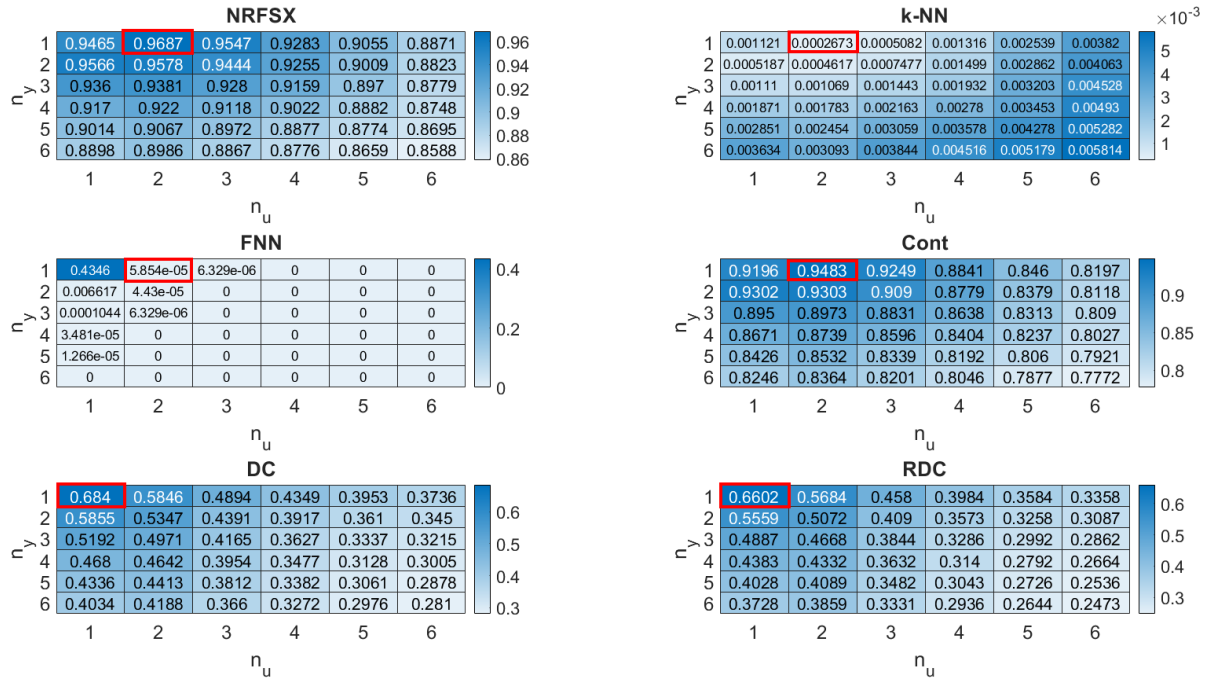


Figure 3.6: **Order identification of the Polymerization reactor.** Each neighborhood-based method can calculate the order of the reactor, except DC and RDC, which may fail due to the redundant input (time-delayed variable). The number of neighbors were selected to be $k = 3$.

Chapter 4

Discussion and conclusion

Feature selection is an essentially important tool for determining relevant variables for modeling. Building a model may not be worth the effort; the data may not be informative enough, or problems regarding identification may mask the relevancy of the input variables. Model-free techniques may prove whether the data provides enough information to optimize the parameters and the necessary features are selected, which may reduce the overall error and boost the performance and speed of the model. Thus feature selection algorithms should be performed without a predefined model. Model-free methods, such as the neighborhood-based approaches, are favorable as they provide information on which features should be selected and whether a dataset holds enough information for optimizing parameters. However, the algorithms have meta-parameters such as the optimal number of neighbors, which is optimized through the mean squared error of the k -NN algorithm.

I generalized the neighborhood-based methods, which resulted in a similar algorithm to trustworthiness and continuity, namely the neighborhood ranking-based feature selection (NRFS). The novel approach has better runtime than the others and is capable of determining nonlinear correlation for feature selection. The k nearest neighborhood evaluation of false nearest neighbors was also introduced. This work proposed non-trivial connections between the methods: FNN is considered the special case of k -NN and trustworthiness and continuity (T&C). At the same time, T&C can be derived from NRFS, which is an instance of (rank) distance correlation. I propose using the neighborhood-based methods for feature selection. The disadvantage of the methods is that they require an optimization algorithm for the selection itself, as they are candidates for cost functions. It is important to note that the runtime of the neighbor-based algorithms is directly influenced by the amount of observed points. The computational time was shown to be significantly less than neural networks. Immensely high observation count may significantly decelerate the feature selection, to which sampling may be a good solution.

Neighbor-based algorithms are also exceptional in filtering the observation. If the

features were selected, points that significantly increase the mean squared error might be deemed an outlier. Outlier detection may also be used iteratively to filter out the invalid observations and remove features that were deemed relevant due to the outliers. As the neighborhood methods are one-directional, meaning that the approaches include one-way projection of one set of variables to the other, the causality may also be established. However due to its complexity, it is designated as a future work of the author. I also generalized the FNN for the k -th nearest neighbors that is not yet implemented.

The methods are applied for feature selection and have been tried on simpler examples of linear monotonic and nonlinear functions. Each solved the linear equation, whereas the monotonous one proved difficult for the FNN. nonlinear equations were only solved by the local variation of NRFS, k -NN, and continuity. I also analyzed the Friedman database. I employed k -NN as a cost to establish the optimal number of neighbors used throughout the subsection. Both forward selection and genetic algorithms optimized the feature selections using the algorithms mentioned above as cost functions. In the case of forward selection, the first five features were correctly determined except for ranking distance correlation. The FNN, And distance correlations gained their minimum before selecting the fifth feature. The NRFS, k -NN and continuity were successful in predicting the correct model.

The proposed algorithms are capable of identifying the order of dynamic systems. As a use case, I examined the input and output of a simulated polymerization reactor. Here the input variable consisted of the time-delayed version (two units) of the independent variable along with the output variable delayed by one unit of time. Distance and ranking distance correlation failed to determine the order of the system, as they cannot differentiate between the same vectors even if it was only shifted by one index. The other methods determined the two order input and one order output of the system. Both the simple and the dynamic system was optimized by brute force.

Even though genetic algorithms require more time to calculate the best combination due to their heuristic nature, their optimization of the feature selection was more successful than that of the forward selection. Continuity performed the best, though the time required was significantly higher than the others. The NRFS and distance correlation performed very similarly regarding time and accuracy.

The neighborhood-based methods proved to be capable of feature selection. Possible applications may include the identification of dynamic systems, selection of appropriate variables for industrial and socio-economic systems. Regarding environmental sciences, feature selection may play an important role in exploring connections between independent variables, such as greenhouse gas emissions, degradation of ecosystems, education and other economic indicators.

Acknowledgments

This research was funded by the National Laboratory for Climate Change (NKFIH-471-3/2021) and the Foundation For the University of Pannonia (Scholarship for the Scientific Student's Association Conference: PA-S2R-00006/2022). Also, I would like to give my heartiest appreciation to my supervisor, János Abonyi who encouraged me to give my best to complete this work and spared incomprehensible amount of time to help my efforts come true.

Bibliography

- [1] Viktor Sebestyén, Tímea Czvetkó, and János Abonyi. The applicability of big data in climate change research: The importance of system of systems thinking. *Frontiers in Environmental Science*, 9, 2021.
- [2] Ryan J. Urbanowicz, Melissa Meeker, William La Cava, Randal S. Olson, and Jason H. Moore. Relief-based feature selection: Introduction and review. *Journal of Biomedical Informatics*, 85:189–203, 2018.
- [3] A. Jović, K. Brkić, and N. Bogunović. A review of feature selection methods with applications. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1200–1205, 2015.
- [4] S. Visalakshi and V. Radha. A literature review of feature selection techniques and applications: Review of feature selection in data mining. In *2014 IEEE International Conference on Computational Intelligence and Computing Research*, pages 1–6, 2014.
- [5] Chao Gao, Hanbo Sun, Tuo Wang, Ming Tang, Nico Bohnen, Martijn Muller, Talia Herman, Nir Giladi, Alexandr Kalinin, Cathie Spino, William Dauer, Jeffrey Hausdorff, and Ivo Dinov. Model-based and model-free machine learning techniques for diagnostic prediction and classification of clinical outcomes in parkinson’s disease. *Scientific Reports*, 8, 05 2018.
- [6] Antonio R. Linero. Bayesian regression trees for high-dimensional prediction and variable selection. *Journal of the American Statistical Association*, 113(522):626–636, 2018.
- [7] Camilo L. M. Morais, Fernanda S. L. Costa, and Kássio M. G. Lima. Variable selection with a support vector machine for discriminating cryptococcus fungal species based on atr-ftir spectroscopy. *Anal. Methods*, 9:2964–2970, 2017.
- [8] Giovanna Castellano and Anna Maria Fanelli. Variable selection using neural-network models. *Neurocomputing*, 31(1):1–13, 2000.

- [9] Mona Azadkia. Optimal choice of k for k -nearest neighbor regression, 2020.
- [10] Matthew B. Kennel, Reggie Brown, and Henry D. I. Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Phys. Rev. A*, 45:3403–3411, Mar 1992.
- [11] V. Onclinx, V. Wertz, and M. Verleysen. Nonlinear data projection on non-euclidean manifolds with controlled trade-off between trustworthiness and continuity. *Neurocomputing*, 72(7):1444–1454, 2009. Advances in Machine Learning and Computational Intelligence.
- [12] Gábor J. Székely, Maria L. Rizzo, and Nail K. Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769 – 2794, 2007.
- [13] Mirko Kück and Michael Freitag. Forecasting of customer demands for production planning by local k -nearest neighbor models. *International Journal of Production Economics*, 231:107837, 2021.
- [14] Chao Hu, Gaurav Jain, Puqiang Zhang, Craig Schmidt, Parthasarathy Gomadam, and Tom Gorka. Data-driven method based on particle swarm optimization and k -nearest neighbor regression for estimating capacity of lithium-ion battery. *Applied Energy*, 129:49–55, 2014.
- [15] Gongde Guo, Daniel Neagu, and Mark T. D. Cronin. Using knn model for automatic feature selection. In Sameer Singh, Maneesha Singh, Chid Apte, and Petra Perner, editors, *Pattern Recognition and Data Mining*, pages 410–419, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
- [16] Donald A. Adjeroh, E. James Harner, and Shengqiao Li. Random knn modeling and variable selection for high dimensional data. 2009.
- [17] Labiba Gillani Fahad, Syed Fahad Tahir, and Muttukrishnan Rajarajan. Feature selection and data balancing for activity recognition in smart homes. In *2015 IEEE International Conference on Communications (ICC)*, pages 512–517, 2015.
- [18] Fouad Khelifi and Jianmin Jiang. k -nn regression to improve statistical feature extraction for texture retrieval. *IEEE Transactions on Image Processing*, 20(1):293–298, 2011.
- [19] Amir Navot, Lavi Shpigelman, Naftali Tishby, and Eilon Vaadia. Nearest neighbor based feature selection for regression and its application to neural activity. In

- Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2006.
- [20] Aiguo Wang, Ning An, Guilin Chen, Lian Li, and Gil Alterovitz. Accelerating wrapper-based feature selection with k-nearest-neighbor. *Knowledge-Based Systems*, 83:81–91, 2015.
- [21] Chan Hee Park and Seoung Bum Kim. Sequential random k-nearest neighbor feature selection for high-dimensional data. *Expert Systems with Applications*, 42(5):2336–2342, 2015.
- [22] P. Bugata and P. Drotár. Weighted nearest neighbors feature selection. *Knowledge-Based Systems*, 163:749–761, 2019.
- [23] R. Puspadini, H. Mawengkang, and S. Efendi. Feature selection on k-nearest neighbor algorithm using similarity measure. pages 226–231, 2020. cited By 1.
- [24] Mahesh Pal, Teja B. Charan, and Akshay Poriya. K-nearest neighbour-based feature selection using hyperspectral data. *Remote Sensing Letters*, 12(2):132–141, 2021.
- [25] Balazs Feil, János Abonyi, and Ferenc Szeifert. Determining the model order of nonlinear input-output systems by fuzzy clustering. 01 2003.
- [26] C. Rhodes and M. Morari. Determining the model order of nonlinear input/output systems directly from data. In *Proceedings of 1995 American Control Conference - ACC'95*, volume 3, pages 2190–2194 vol.3, 1995.
- [27] Rainer Hegger and Holger Kantz. Improved false nearest neighbor method to detect determinism in time series data. *Phys. Rev. E*, 60:4970–4973, Oct 1999.
- [28] Anna Krakovská, Kristína Mezeiová, and Hana Budáčová. Use of false nearest neighbours for selecting variables and embedding parameters for state space reconstruction. *Journal of Complex Systems*, 2015:12, 03 2015.
- [29] Shilun Zuo and Zhiqiang Liao. Bearing fault dominant symptom parameters selection based on canonical discriminant analysis and false nearest neighbor using ga filtering signal. *Mathematical Problems in Engineering*, 2020:1–13, 04 2020.
- [30] Guijun Ma, Yong Zhang, Cheng Cheng, Beitong Zhou, Pengchao Hu, and Ye Yuan. Remaining useful life prediction of lithium-ion batteries based on false nearest neighbors and a hybrid neural network. *Applied Energy*, 253:113626, 2019.

- [31] Qing-Ping Mei, Tai-Fu Li, Li-Zhong Yao, Xiao-Hong Liu, Yuan-Li Hu, and Lei Hu. Characterization of a wavelength selection method using near-infrared spectroscopy and partial least squares with false nearest neighbors and its application in the detection of the chemical oxygen demand of waste liquid. *Spectroscopy Letters*, 52(9):553–562, 2019.
- [32] X.-H. Gu, T.-F. Li, L.-P. Yang, J. Yi, and W. Zhou. Original feature selection based on false nearest neighbor criterion in supervised locality preserving subspace. *Guangxue Jingmi Gongcheng/Optics and Precision Engineering*, 22(7):1921–1928, 2014. cited By 0.
- [33] K. Héberger. Sum of ranking differences compares methods or models fairly. *TrAC - Trends in Analytical Chemistry*, 29(1):101–109, 2010.
- [34] Károly Héberger and Klára Kollár-Hunek. Sum of ranking differences for method discrimination and its validation: Comparison of ranks with random numbers. *Journal of Chemometrics*, 25:151–158, 04 2011.
- [35] Károly Héberger. Method and model comparison by sum of ranking differences in cases of repeated observations (ties). *Chemometrics and Intelligent Laboratory Systems*, 127:139–146, 08 2013.
- [36] K.Z. Mao. Orthogonal forward selection and backward elimination algorithms for feature subset selection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1):629–634, 2004.
- [37] Xiao-Ying Liu, Yong Liang, Sai Wang, Zi-Yi Yang, and Han-Shuo Ye. A hybrid genetic algorithm with wrapper-embedded approaches for feature selection. *IEEE Access*, 6:22863–22874, 2018.
- [38] Hongwei Tan, Guodong Wang, Wendong Wang, and Zili Zhang. Feature selection based on distance correlation: a filter algorithm. *Journal of Applied Statistics*, 49(2):411–426, 2022.
- [39] Aida Brankovic, Marjan Hosseini, and Luigi Piroddi. A distributed feature selection algorithm based on distance correlation with an application to microarrays. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(6):1802–1815, 2019.
- [40] Jerome H. Friedman. Multivariate Adaptive Regression Splines. *The Annals of Statistics*, 19(1):1 – 67, 1991.

- [41] K U Syaliman, E B Nababan, and O S Sitompul. Improving the accuracy of k-nearest neighbor using local mean based and distance weight. *Journal of Physics: Conference Series*, 978:012047, mar 2018.
- [42] Balazs Feil, János Abonyi, and Ferenc Szeifert. Model order selection of nonlinear input-output models - a clustering based approach. *Journal of Process Control*, 14:593–602, 09 2004.
- [43] R Babuška and HB Verbruggen. New approach to constructing fuzzy relational models from data. In *Proceedings Third European Congress on Intelligent Techniques and Soft Computing EUFIT'95*, pages 583–587, 1995.
- [44] Chengwei Xiao, Jiaqi Ye, Rui Máximo Esteves, and Chunming Rong. Using spearman’s correlation coefficients for exploratory data analysis on big dataset. *Concurrency and Computation: Practice and Experience*, 28(14):3866–3878, 2016.
- [45] C. Rhodes and M. Morari. Determining the model order of nonlinear input/output systems. *AIChE Journal*, 44:151–163, 1998.
- [46] Stephen Billings. Nonlinear system identification: Narmax methods in the time, frequency, and spatio-temporal domains. *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*, 08 2013.
- [47] F.J. Doyle, B.A. Ogunnaike, and R. K. Pearson. Nonlinear model-based control using second-order volterra models. *Automatica*, 31:697, 1995.