CORVINUS UNIVERSITY of BUDAPEST

Doctoral School of
Economics, Business
and Informatics

COLLECTION OF THESES

László Kovács

Comparison of Feature Selection Algorithms for Generalized Additive Models

Analysis of a New, Hybrid Metaheuristic

Ph.D. dissertation

Tutors:

Blanka Láng dr.
associate professor

Péter Racskó dr.
scientific advisor

Budapest, 2021

Department of Computer Science

COLLECTION OF THESES

László Kovács

Comparison of Feature Selection Algorithms for Generalized Additive Models

Analysis of a New, Hybrid Metaheuristic

Ph.D. dissertation

Tutors:

Blanka Láng dr.	**Péter Racskó dr.**
associate professor	scientific advisor

Contents

1. Former Research, Justification for the Topic

In supervised machine learning our aim is to predict a well-defined target variable as accurately as possible by utilizing the known values of several feature variables. Nowadays many complex algorithms are available to solve this task. Such as deep learning neural networks, random forests, support vector machines, etc. On the other hand, more and more authors, like Molnar (2020) and Du et al. (2019) draw attention to the fact that those algorithms that provide the most accurate estimates of the target variable are poor at determining marginal effects of the feature variables to the target. However, in certain practical applications, the most important result of supervised learning is not necessarily the accurate estimation of the target, but the discovery of each feature’s marginal effect. For example, a bank has to offer a clear reasoning when declining a credit application. In cases like this, out focus should be on building an explainer model, not a predictive one.

In our current bid data environment, when the number of possible features is large, determining marginal effects can be challenging even for a linear regression model. One tool that can be utilized to make supervised learning models more interpretable is feature selection as proposed by Molnar (2020) and James at al. (2013).

The most important principle of feature selection as defined by Hall (1999) is to identify a feature set where each element correlates well with the target, but the features are uncorrelated with each other. In a linear case this means the avoidance of multicollinearity. This principle is similar to the preference of parsimonious models in Econometrics (Wooldridge, 2016).

In my doctoral dissertation, I examine the behaviour of a new hybrid genetic-improved harmony search algorithm (HGHS algorithm for short) in solving the feature selection task of generalized additive models. Research questions K1-K5 of the dissertation aim to explore the technical solutions of applying the algorithm for generalized additive models, the quality of the models proposed by the algorithm and means to decrease the expected runtime of the algorithm.

1.1. Former Research

The most important predecessors of the research conducted in my dissertation are two of my co-authored publications: Láng – Kovács (2014) and Láng et al. (2017). I participated in these two studies as a BSc Student in Business Information Systems. Both publications are based on my own independent work that I did for my BSc thesis and for a paper I presented at the Scientific Student Associations' Conference of Corvinus University of Budapest in 2014.

In these two papers, my co-authors and I introduce the HGHS algorithm for solving feature selection problems. The algorithm is a best subset feature selector. Its most important novelty to the already existing algorithms proposed by the literature is its new constraints for eliminating feature redundancy in the models. In the linear case, this means selecting uncorrelated features, which is one of the most important assumptions during the parameter estimation of linear models. During the development of the algorithm my contribution was idea for combining the genetic and harmony search algorithms. The idea behind this combination is that during feature selection we need recombination operators with more randomness than those of the genetic algorithm. These recombination operators can be provided by the harmony search algorithm, but we need to preserve the genetic algorithm's parallel generation of new individuals to the new populations. Optimized implementation of the algorithm in C# is a joint work with my co-authors. The first working implementation of HGHS is my own work that is part of my BSc thesis. In the referred two papers, we show that by applying HGHS, we can get feature subsets that are not affected by multicollinearity, which is a great help in the interpretation of marginal effects. Since we have a ceteris paribus assumption when interpreting marginal effects. Furthermore, violating the assumption of uncorrelated features also makes the parameter estimates of linear models less stable (Hastie et al. 2011).

An immediate predecessor of my doctoral dissertation is my independent publication, Kovács (2019) that is based on the research I conducted for my MSc thesis in Actuarial Science. In this paper, I extended the HGHS algorithm to handle Cox's proportional hazard models as well, not just simple linear models during feature selection. With this extended algorithm, I estimate the survival curves of life insurance policies. Similarly to the linear case, the HGHS can propose models with similar prediction accuracy as the other feature selection algorithms proposed for proportional hazard models, but the HGHS models achieve this accuracy with uncorrelated features. Due to this property of the HGHS, the features affecting life insurance policy survival are more easily identified with HGHS than with other feature selection methods proposing models with some redundant features. Most important yield of this research regarding my dissertation is the first version of the R implementation of HGHS. This implementation is the immediate basis for the R codes that I wrote for my dissertation. These R codes are also a product of my own work.

In my doctoral dissertation, I examine how the HGHS algorithm can be extended to handle generalized additive models (GAMs) as well. In GAMs, non-redundancy of features is also an important assumption as feature effects in the model are joined additively. This means that the estimation of the target by GAM is given as a sum of the non-linear functions $\hat{f}_1, \hat{f}_2, \dots, \hat{f}_m$ representing the effect of each feature (Hastie – Tibshirani, 1990). So, the constraing in the HGHS that aims to ensure that only independent features (or at least non-redundant features according to a preferred metric) are selected in the final model is still useful if the aim of the analyst is to use the GAM for identifying the most important features that affect the target in a non-linear way.

1.2. Research Questions

Based ont he former research presented in the Section 1.1., my doctoral dissertation aims to answer research questions K1-K5. These questions investigate the behaviour of HGHS extended to handle GAMs.

K1. Can the extension of HGHS to handle GAMs be realized without modifying its decision variables and its binary individual-representation applied in the linear case?

K2. Can the HGHS find models for real-world datasets with a smaller number of non-redundant features, that have similar prediction accuracy to models proposed by other state-of-the-art feature selection algorithms for GAMs?

K3. Do the random or controlled recombination operators of HGHS need to be preferred for the generating a new population (or memory) when we apply the algorithms for GAMs?

K4. What is the best method to ensure the appropriate quality of the initial population in HGHS? Is it more efficient to run the HGHS from several random initial populations where the population size and the maximum number of generations are small, or is it worth to apply a large population size and maximum number of generations and run the algorithm only once?

K5. What kind of parallelization strategy can better improve the expected runtime of HGHS? Is the simultaneous computation of the models corresponding to each individual preferred? Or parallelized computation of a single model is more efficient? How well can we scale the algorithm with the preferred strategy for parallelization? How does the numerically measured speed-up of HGHS relates to the theoretical maximum speed-up?

2. Applied Methods

I apply the Design Science methodology proposed by Hevner et al. (2004) when answering research questions K1-K5. The research of my dissertation is designed based on the 7 guidelines of Hevner et al. (2004).

Applying the Design Science Methodology means both developing an artifact and answering research questions. In my dissertation, research questions K1-K5. are answered by extending an algorithm of my own design (HGHS), implementing the extensions, and numerically testing the efficiency of the new algorithm. This new algorithm, its implementation and the numerical experiments can be considered as an artifact.

In my dissertation, I have developed a working implementation of the HGHS that can handle the feature selection problem for GAMs on real-world datasets. This means that the HGHS, presented in more detailed in Section 3, satisfies the requirements of the 1st Design Science guideline of Hevner et al. (2004).

In my research, I introduce the theoretical and methodological framework of GAMs based on the latest literature. I emphasize that parameter estimation of GAMs is sensitive to redundancy of features in the model that is called concurvity in the GAM framework. I show that concurvity can make business interpretability of GAMs difficult. By analysing state-of-the-art feature selection algorithms for GAMs, I highlight that the most popular algorithms either do not account for concurvity or only address it in a limited way. So, with the help of literature review, I show that the artifact of my dissertation, the HGHS, provides solution to an important and relevant business problem in line with the 2nd Design Science guideline.

With a detailed literature review, I support my claim that the numerical comparison of the performance of HGHS and the performance of other examined feature selection algorithms for GAMs is done by applying state-of-the-art validation methods and performance metrics. With this, I take into account the 3rd guideline of Design Science.

As a research result, I show that the HGHS algorithm fully takes concurvity into account during feature selection. Furthermore, I apply the HGHS on two practical feature selection problems with different parameters, and I measure its performance in a way that is comparable to that of the other examined feature selection methods. Numerical results confirm that the HGHS can propose concurvity-free GAMs with a prediction accuracy that can approximate that of the other examined algorithms in most of the applied performance metrics. Numerical results also suggest reasonable expected runtime for HGHS when simultaneously computing GAMs corresponding to each individual in the population. With these numerical results, I confirm that the artifact of my research, the HGHS, has verifiable contributions as required by the 4th guideline of Design Science.

Optimizing the performance of HGHS is realized by ceteris paribus sensitivity analysis of the algorithm parameters. Based on the results of parameter optimization, I apply the HGHS on larger datasets in a new way, by running the algorithm from several random initial populations with a small population size to increase efficiency. I also examine the scalability of HGHS by running it on several virtual environments capable for parallel execution with different parameters. The empirical seep-up of HGHS is always compared with the maximum theoretical seed-up by parallelization available on the current environment. Furthermore, I apply strictly unform validation methods when analysing the results of the numerical experiments. With these principles in place, I aim to comply with the 5th Design Science guideline of research rigor and with the 6th guideline of utilizing available means to reach desired ends while satisfying laws in the problem environment.

My doctoral research results have been actively communicated (7th guideline of Design Science) as I presented my research results in several Hungarian and international scientific conferences and published papers in referred scientific journals. My results were presented as part of the “Data Analysis in Practice” lecture series of Corvinus University of Budapest and also presented during the Corvinus Research Week of January 2020. Both forums helped me share my results with interested students at the university and with researchers who can apply the HGHS algorithm in their own research.

To ensure reproducibility of my numerical results, the source code of the HGHS algorithm in R and the R scripts of the numerical performance tests are part of my dissertation. All the R scripts, the training and test datasets in *Rda* format, and the tables containing detailed numerical results in *csv* and *xlsx* formats are available on the https://github.com/KoLa992/Hybrid-algorithm-for-GAMs repository. Every R script was run in R version 3.5.3., on a 64-bit Windows 10 operating system. The hardware configuration used for the numerical experiments (except for the tests of scalability) is a PC with Intel Core i7-8750H 2,20 GHz processor and 8 GB 2666 MHz DDR4 RAM.

3. Results of the Dissertation

In Section 3.1., I introduce the feature selection task to be solved by the HGHS algorithm. In Section 3.2., I highlight the uniqueness of HGHS compared to other state-of-the-art feature selection algorithms in GAM framework. In Section 3.3., I introduce the main concepts of the HGHS algorithm and the parameters that affect its efficiency. Based on the results of this section, I can answer research question K1. In Sections 3.4. and 3.5., I examine the efficiency of HGHS numerically and compare the results with other state-of-the-art feature selection algorithms proposed by the literature. Based on the results of these two chapters, I answer research questions K2, K3 and K4. Section 3.6. deals with the scalability of HGHS and answers research question K5 with more numerical experiments. Finally, I summarise my results in Section 3.7. and evaluate the practical applications of the HGHS algorithm.

3.1. Feature Selection in a GAM Framework

Let $Y = [y_1, y_2, \dots, y_n]^T \in \mathbb{R}^n$ be a vector of an i.i.d. sample drawn from a random variable with a distribution from an exponential family. In a GAM framework, the expected value of Y can be estimated with the help of $X_j = \left[x_{j1}, x_{j2}, \dots, x_{jn}\right]^T$ known features with the model given is equation (1) (Hastie – Tibshirani, 1990).

$$h\big(E(Y)\big) = \varepsilon + \textstyle\sum_{j=1}^{p} f_j\big(X_j\big) \qquad (1)$$

Where $h(\cdot)$ is the link function for the distribution of Y, $\varepsilon = [\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n]^T$ is a vector of model errors and $f_j(\cdot)$ is the transformation function for the jth feature.

In my dissertation, the f_j functions are represented with the help of thin plate splines. The parameters are estimated by penalized iterative weighted least squares (Wood, 2017). The main advantage of this representation to the alternatives proposed by the literature is that it can select the form of f_js automatically with the help of a penalty term that is dependent on the 2nd derivative of f_j. In this representation, we only need to give the maximal number of spline bases as a parameter. This maximal number is denoted as k_j for the jth feature. The optimal choice for k_j is the smallest integer that is large enough that its corresponding f_j thin plate spline function captures most of the variance in X_j. Fortunately, Augustin et al. (2012) proposed a test where the null hypothesis is that $Var\left(f_j\big(X_j\big)\right)$ is not significantly different from $Var\big(X_j\big)$. With the help of this test, selection of k_j can be automated in the HGHS. Details for this method are given in Section 3.3.

Let $X = \{X_1, X_2, \dots, X_m\}$ be a set of all possible features for a GAM. During feature selection, the task is to choose a $\tilde{X} = \{X_1, X_2, \dots, X_p\} \subseteq X$ subset such that the resulting GAM has the best out-of-sample performance. To achieve the best out-of-sample performance, we need to compromise in using in-sample information. If we use too few in-sample information, we will fail to gain a good enough understanding of the relationship between features and the target. However, if we use too much in-sample information, we will overfocus our model and it will have poor out-of-sample performance. To maximize out-of-sample performance for our GAM, it is advised to apply the principles of parsimony during feature selection. According to this principle, we should select the $\tilde{X} \subseteq X$ subset such as that the GAM we obtain has the best possible fit to Y in-sample, with the application of as few features as possible (Wooldridge, 2016).

Out-of-sample performance for our model can be measured in several ways. In my dissertation, I utilize the adjusted McFadden's pseudo-R-squared measure ($\bar{R}^2$). Description of the measure is given in McFadden (1974). So, during feature selection $\tilde{X}$ should be selected such that $\bar{R}^2$ is maximal. On the other hand, evaluating every possible subset of X is a NP-hard problem as we have $2^m - 1$ possible subsets to examine, excluding the empty set (Huo – Ni, 2007).

To avoid redundancy between the features that are selected in $\tilde{X}$, an extra constraint should be applied during feature selection. To introduce this new constraint, a measure for the non-linear extension of multicollinearity called concurvity must be given. Based on Wood (2017), concurvity can be measured with an index on a $[0,1]$ scale in the case of GAMs that apply thin plate splines. If the index is 0 for the jth feature, then the feature's variance cannot be explained by non-linear, additive combinations of the rest of the features in $\tilde{X}$. On the other hand, if the index is 1, then jth feature can be perfectly reproduced by the rest of the features in in $\tilde{X}$. The main concept of the index is the based on the idea that a smooth term, f_j, in the model can be decomposed into a part, g_j, that lies entirely in the space of one or more other terms $f_{c \neq j}$ in the model, and a remainder part that is completely within the term's own space. If g_j makes up a large part of f_j then there is a concurvity problem. So, the index is given as the square of $\frac{\|g_j\|}{\|f_j\|}$, that is the ratio of the squared Euclidean norms of the vectors of f_j and g_j evaluated at the observed covariate values. In my dissertation, I follow the recommendations of Wood (2017): if this concurvity index is above 0.5 for a X_j feature, then it can be considered redundant (has harmful concurvity).

3.2. Feature Selection Algorithms for GAMs

Based on results of my literature review, feature selection algorithms in a GAM framework can be classified into three groups.

The first group follows the stepwise logic: it only adds or removes one feature at an iteration, and it follows this procedure until the aim function for the feature selection cannot be improved further significantly. The specific algorithms in this group show differences in the subtleness of their stepwise steps. The classical Stepwise algorithm treats the features in a binary way: a feature is either included in $\tilde{X}$ or not. See Venables – Ripley (2002) and Hastie et al. (2011) for details. On the other hand, the GAMBoost algorithm utilizes basis spline functions, and in each iteration, it adds "weight" to the feature that causes the largest improvement in the aim function of the feature selection. Once the algorithm has stopped, features with 0 weight (all the basis functions have 0 coefficients) are the ones that are excluded from $\tilde{X}$. So, this algorithm also follows a stepwise logic, since it only modifies the properties of one feature in each iteration, but it does it in a continuous and not a binary way. Details of this algorithm can be found in Binder – Tutz (2008) and Schmid – Hothorn (2008). The concept of GAMBoost is improved in the Modified Backfitting procedure by Belitz – Lang (2008). This paper allows increasing the weights of several features in one iteration, making the algorithm more suitable for parallelization (Umaluf et al., 2015).

The second group is the family of regularization techniques. These algorithms are generalizations of the Lasso by Tibshirani (1996) to non-linear models. The main concept of regularization methods is to execute feature selection during parameter estimation with some special penalty term in the error function that is minimized by choosing the model parameters. These penalty terms result in estimating non-relevant f_js as identically 0. Specific algorithms in this group include: COSSO (Lin – Zhang, 2006), two kinds of penalized thin plate splines (Marra – Wood, 2011), Nonnegative Garrotte (Cantoni et al., 2011).

An important property of both stepwise and regularization algorithms is that they do not attempt to directly to avoid multicollinearity, or in the non-linear case, concurvity. Furthermore, multiple authors have showed that consistency of stepwise and regularization methods is violated if harmful multicollinearity, or in the non-linear case, concurvity is present in the model: Chong – Jun (2005), Zhao – Yu (2006), Signoretto et al. (2008), Jia – Yu (2010). The lack of consistency for GAM feature selection means that features with identically 0 f_js in the population can be selected to $\tilde{X}$, and features with non-zero basis spline coefficients in the populations can be excluded from $\tilde{X}$.

The third group of feature selection algorithm contains algorithms that executes feature selection based on mutual information of the target and the features. Algorithms in this group are special in the sense that they do not assume GAM, or any other model framework for feature selection. These methods apply an $I(x, y)$ index to measure the mutual information of random variables x and y, and they maximize a custom feature selection objective function with some standard optimization algorithm. The custom objective functions are constructed in a way to ensure selecting X_js to $\tilde{X}$ where $I(Y, X_j)$ is high, but $I(X_k, X_j)$ is low for all $k \neq j$. With the inclusion of this second aspect, these objective functions try to control for redundancy (multicollinearity or concurvity) between the selected features. However, application of $I(x, y)$ mutual information measures only examine redundancy of features between variable pairs. These algorithms cannot handle cases were a feature can be constructed as a multivariate function of other features included in the model. So, concurvity can still be a problem for GAMs proposed by algorithms that use $I(x, y)$ measures. Most popular algorithms in this group are the mRMR (Song et al., 2012) and the HSIC-Lasso (Climente-González, et al., 2019).

The application metaheuristic algorithms to solve feature selection tasks is quite frequent in recent literature. Some examples: Wang et al. (2015), Krömer – Platoš (2016), Mafarja – Mirjalili (2017), Sayed et al. (2019) és Abdel-Basset et al. (2020). A common characteristic of these papers that they do not consider GLM or GAM as a framework during feature selection. They work in the framework of other supervised learning models like k-NN, support vector machine, neural network, etc. Partly for this reason, these algorithms only interpret feature selection for classification tasks as the most popular among supervised learning tasks. These works only focus on improving prediction accuracy via feature selection, so interpreting marginal effect of features is disregarded. So, they do not add any constraint for redundancy (multicollinearity or concurvity) to their feature selection tasks.

To the best of my knowledge, the HGHS algorithm is the only metaheuristic feature selection algorithm that applies constraints to ensure a not redundant feature set in its proposed final model.

3.3. The HGHS algorithm in GAM framework

In my previous works I proposed a hybrid genetic-harmony search algorithm (HGHS algorithm) to solve feature selection in linear models. The algorithm applies the VIF measure of features to eliminate multicollinearity completely from its proposed model. So, the algorithm handles multivariate redundancy between features as well, and is not limited to tackle redundancy only between variable pairs as the mRMR or the HSIC-Lasso. In this section, I introduce the extension of the algorithm to the GAM framework by applying thin plate splines to represent the f_j transformation functions of the features. In my dissertation the aim function of HGHS is McFadden's $\bar{R}^2$. The concepts of the genetic algorithm can be found in Goldberg (1989) and a detailed description of the harmony search algorithm is given in Mahdavi et al. (2007). Flow chart of HGHS is given in Figure 7 of my dissertation.

The HGHS algorithm represents possible solutions of the feature selection task as a bit sequence with a length of m. So, the only decision variable is the inclusion or exclusion of a specific feature. In the HGHS, the parallelizable population (or memory in the terms of the harmony search algorithm) handling of the genetic algorithm is preserved, but the cross-over recombination operator of the genetic algorithm is replaced to the more random operators from the harmony search algorithm. This replacement is necessary as the cross-over operator of the genetic algorithm is only effective for problems where the quality of individuals is worth improving in smaller parts and creating new solutions by crossing-over these parts from different previous solutions. However, as datasets usually contain feature variables in a random order, there are no patterns in specific parts of the bit sequences representing a solution.

The probability of generating a new individual from the better-than-average individuals of the previous memory ($HMCR$, harmony memory consideration rate) is increasing during iterations, while the probability of mutation (modification of a chosen element, denoted *bw*) is decreasing. With this fine tuning, completely random individual generation of new individuals is dominant in the early iterations of the algorithm. As we get closer and closer to the optimal solution during the iterations, smaller and smaller parts of the search space need to be mapped during the generation of new individuals, so inheriting information form better-than-average individuals of the previous memory can become more dominant. An early stopping criterion is also applied in the HGHS: if in some of the last iterations, the objective function value of the best individual of the population remains unchanged, the algorithm stops. The number of iterations without change in the objective function can be given as a parameter.

The HGHS can only be applied in a GAM framework if the solutions can still be represented as bit sequences. If order and breakpoints of the spline function fitted on a feature need to be determined as well, then a more complex representation of individuals is required. Fortunately, application of thin plate splines introduced in Section 3.1. ensures the preservation of binary representation of individuals, so search space of the feature selection task does not increase in size.

In the algorithms, I apply the following technique. If an X_j feature is included in the GAM, then by default I set $k_j = 10$ as recommended by Wood (2017). If the value set of X_j is smaller than 10, then k_j equals the number of unique values in X_j. If p-value of the test proposed by Augustin et al. (2012) is less than $\alpha = 0{,}01$, then I increase k_j by 5 until I have chosen a large enough k_j for the feature.

To control for concurvity, only those solutions can be allowed to be transferred to the new memory, where the concurvity index defined in Section 3.1. is below 0.5 for all features in the model. If the ith individual in the population satisfies this constraint, then binary variable C_i takes the value of 1, and 0 otherwise. Furthermore, the algorithm should prefer those models that contains variables where we can reject the null hypothesis of $f_j \equiv 0$. To test this null hypothesis, a χ^2-test proposed by Marra – Wood (2011) can be applied. If on a user-specified significance level, we can reject the null hypothesis of $f_j \equiv 0$ for all features in the ith individual in the population, then binary variable S_i takes the value of 1, and 0 otherwise.

Incorporating these constraints is done through the selection of better-than-average individuals. The average objective function of the current population (or memory) is determined by the weighted average formula of (2), and only those i individuals can be labelled as better-than-average, where $C_i = S_i = 1$.

$$\bar{R}_M^2 = \frac{\sum_{i=1}^{N} \bar{R}_i^2 \cdot C_i \cdot S_i}{\sum_{i=1}^{N} C_i \cdot S_i} \qquad (2)$$

In (2) N is the size of the population/memory, $\bar{R}_i^2$ is the objective function of the ith individual. If $\sum_{i=1}^{N} C_i \cdot S_i = 0$, then $\bar{R}_M^2$ is simply the arithmetic mean of the $\bar{R}_i^2$ values, and every individual can be considered as a better-than-average individual.

Due to these constraints of the HGHS, it is possible that there is no individual in the initial random memory that can be selected for the new memory and satisfies the $C_i = S_i = 1$ constraints. So, it is also easily possible that it takes several iterations until the algorithm finds solutions that satisfy all the constraints. This suggest that the runtime of the algorithm is highly dependent on the quality of its initial population/memory. To handle this, the algorithm could be run several times with several initial random memories with a smaller memory size and maximal iteration number to conserve the runtime of one trial as suggested by research question 4. In this case, the individual with best objective function value out of the final models of each trial can be considered as optimal. Alternatively, research question K4 proposes running the algorithm only once, but with a high memory size to tackle the poor quality of the initial memory.

Regarding research question K1, we can conclude that the application of thin plate splines presented in this section allows a positive answer to the question. Since, with the proposed procedure to set k_j, we can select order and breakpoints of spline functions automatically. This results in preserving the decision variables of the linear HGHS and change in the binary representation of individuals is not necessary.

3.4. Numerical Results of HGHS for a Small Task

Numerical comparison of HGHS with other feature selection algorithms working in a GAM framework is executed on two real-world datasets. Source of the first dataset is , and it contains 9 variables for 1030 concrete grinders. Our first examined real world dataset is proposed by Yeh (1998) and contains 9 variables of 1030 concrete girders. The task is to estimate the comprehensive strength of concrete material as a non-linear function of age and ingredients. As $m = 8$, the feature selection task is small, which means all of the possible subsets can be generated and the global optima is easily selected. Description of the variables is also given in Yeh (1998). The aim of applying the feature selection algorithms on this dataset is to examine how frequently can each algorithm identify the global optima. The cleaned-up dataset is split to training and test sets in a 7:3 ratio.

Results of the examined feature selection algorithms on this Concrete Comprehensive Strength Dataset are summarised in Tables 2 and 3 of the dissertation. Due to the stochastic nature of these algorithms, every procedure is run 30 times and the results of the best model out of 30 are reported.

Based on these results, I can conclude regarding research question K2 that prediction accuracy on the test set of the model proposed by HGHS is not significantly different from that of the other models proposed by the rest of the algorithms examined in my dissertation. Next to HGHS, the mRMR algorithm is the only one that is capable of proposing models that are free from concurvity in every selected feature (Table 2). However, prediction accuracy of mRMR is less than that of the HGHS. These results are robust regarding all the 9 performance metrics that are applied. Compared to the only other algorithm that proposes a concurvity-free model, the mRMR, the GAM proposed by HGHS has better results in all the 9 performance metrics (Table 3).

The small size of the Concrete Comprehensive Strength Dataset makes optimizing parameters of HGHS possible. Table 4 of the dissertation shows the results of the ceteris paribus optimization of the HGHS parameters. These results suggest that HGHS prefers the random generation of new solutions, but inheritance from the previous population should not be neglected: the $HMCR$ should be increased from 5% to 35% during the iterations. Decreasing the mutation bw probability from 90% to 10% during the HA iterations also supports the idea. Based on Table 4, the answer to research question K3 is that in case of GAM framework, completely random generation of new solutions should be preferred in the HGHS, but controlled solution generation based on inheritance from the previous population/memory should not be completely neglected.

Another important conclusion is that when the HGHS is run with its optimal settings given in Table 4, then in more than third of trials (12/30), the algorithm can find the global optima or the second-best solution before the 5th iteration. So, only about $\frac{5*20}{2^8} = 0{,}4 = 40\%$ of the search space is examined by the algorithm when one of the best two solutions is found. Based on this observation, we can conclude in case of research question K4 that the poor quality of initial memory/population should be addressed by running the algorithm from several initial random populations rather than running the algorithm once with large population/memory size.

Detailed results of the ceteris paribus search of the optimal parametrization are reported in the appendices of the dissertation.

3.5. Numerical Results of HGHS for a Large Task

In the second dataset used for numerical experiments, the task is to estimate to estimate for clients in a Taiwanese bank if they are to report default on their credit card loans in one month from now. This dataset consists of 30000 observations and 26 possible features after applying a dummy coding for categorical variables. Source of the dataset is Yeh – Lien (2009) and description of the variables is also given in this paper. The feature selection task in this case is not solvable via examining all possible feature subsets and selecting the best one. Best models proposed by each algorithm are only comparable to each other, there is no reference point. The cleaned-up dataset is split to training and test sets in a 7:3 ratio. I apply the results of the GLM and LightGBM models of Yang – Zhang (2018) as a benchmark. Further benchmarks are a

Decision Tree algorithm and a Random Forest combined with Recursive Feature Elimination algorithm.

Results of the benchmark and GAM feature selection algorithms on the Credit Card Default Dataset are shown in Tables 6 and 7 of the dissertation. Due to the stochastic nature of these algorithms, every procedure is run 20 times and the results of the best model out of 20 are reported. Based on the conclusions drawn from the Concrete Comprehensive Strength Dataset, the HGHS is run from 5 initial random populations with small population size (60) and number of maximal generations (6). Other parameter settings are the optimal values given in Table 4. Detailed explanation of the parametrization is given in Section 10.4. of the dissertation. Numerical results on the Credit Card Default Dataset confirm the answer given to research question K2, stating that prediction performance on the test set for the models proposed by HGHS is not significantly different from that of the models proposed by the other examined feature selection algorithms. Furthermore, based on results in Table 6, it can also be concluded that the AUC of the GAM proposed by HGHS is higher than that of the GLM in Yang – Zhang (2018) and is not significantly smaller than the AUC of the LightGBM model which is the model with the best AUC according to the authors. The HGHS is also the only one from the algorithms that are working in a GAM framework that can propose a feature subset completely free from concurvity on the Credit Card Default Dataset. This latter result is not matched either by the mRMR or by the HSIC-Lasso, even though these algorithms control for concurvity as this control is only for redundancy between variable pairs.

The Credit Card Default Dataset shows more diverse results when the set of applied performance metrics is extended than the Concrete Comprehensive Strength Dataset examined in Section 3.4. Here, if we examine, the two models that have a feature set without concurvity (the HGHS and the Decision Tree), we can find that in prediction accuracy on the test set, the HGHS has better performance in "recall-like" metrics, while the Decision Tree has stronger "precision-like" metrics. In analyses predicting credit risk, the "recall-like" metrics are more relevant for decision makers according to Moula (2017). The HGHS has better performance regarding these metrics, so it can be suggested that non-redundant features influencing credit risk should be identified based on the results of a HGHS model rather than a decision tree model.

Expected runtime of the HGHS is acceptable compared to the rest of the algorithms (the expected runtime is similar to that of the Random Forest combined with RFE and significantly faster than the expected runtime of the GAMBoost) based on Table 6. Based on these results, I can also confirm my answer to research question K4: it is worth running the HGHS from several initial random memories/populations with smaller memory size and maximal number of generations to handle the poor quality of the initial memory/population of the algorithm that arises in GAM frameworks.

3.6. Examining the Scalability of HGHS

Vertical scalability of the HGHS algorithm is examined as a function of processor core used to parallelly compute the GAMs corresponding to individuals in a given population (or memory). The numerical experiments for scalability are executed on the Credit Card Default Dataset as Table 6 of the dissertation clearly shows that expected runtime is critical here for the HGHS, due the larger size of this dataset.

Hardware configuration used for the numerical experiments of Sections 3.4. and 3.5. is given in Section 1. Hardware resources utilized for numerical examination of scalability of HGHS are provided by Microsoft Azure Data Science Virtual Machine (DSVM). On the DVSM platform, virtual hardware configurations with 4, 8, 12 and 16 processor cores are available with a university account (Etaati, 2019).

Before numerically examining scalability of HGHS, it is essential to determine the maximum amount of speed-up achievable by parallelization given the number of available processor cores as a reference point. Amdahl's law can be applied to determine this upper limit (Bryant et al., 2016). Amdahl's law is a formula that can be used to determine the upper limit of the speed-up gained by parallelizing a task or algorithm. Exact description of this law can be found in Amdahl (1967).

During the numerical experiments executed in Azure DSVM, the HGHS algorithm is still run 20 times and expected runtime and the relative standard deviation of runtimes are examined as a function processor cores, and finally the amount of speed-up compared to an architecture using only one core is given. These numerical results of speed-up are compared to their theoretical maximum value calculated according to Amdahl's law. These results are contained in Table 9 and Figure 20 of the dissertation. The HGHS achieves the maximal theoretical speed-up according to Amdahl's law in case of 4 cores, but it fails to meet these maximum speed-ups in case of more cores. The explanation behind this phenomenon is the optimal parametrization of HGHS that prefers the random elements of the algorithm that causes idle time in some iterations. This idle time also causes high relative standard deviation of runtimes on the hardware configuration with 16 cores.

To understand the behaviour of the idle time experienced in the parallelization of HGHS on more than 4 processor cores, one should take a look at the histogram describing the distribution of the runtimes required to compute a GAM in the Credit Card Default Dataset based on 100 simulated feature subsets. This histogram is given in Figure 21 of the dissertation. Based on this histogram, distribution of the runtimes has a significant right skew, with one huge outlier. So, there's an empirical probability of 1% that we get a feature subset in a population/memory, where computing the corresponding GAM takes an extremely long time. This causes problems during parallelization as the other processor cores "need to wait" for the core that computes this GAM with extremely long runtime to finish before the selection of better-than-average individuals start in the population. This wait is needed so the average objective function can be calculated by (2) for the current population/memory. However, (2) can only be computed once all the individuals have a computed corresponding GAM.

Due to waiting for the individual with the highest runtime, there will be an idle time in every iteration of the HGHS when only one processor cores are working, and the rest have no computation tasks. If there is no individual in the population with an extremely outlier runtime, then the idle times can be quite short, so the runtime of whole the algorithm can approximate the maximal theoretical speed-up of Amdahl's law. On the other hand, if many populations contain individuals with outlier runtimes, then the idle times caused by these outliers cumulate, and runtime of the whole algorithm will miss the theoretically maximal speed-up. As optimal parametrization of HGHS is dominated by the random elements of the algorithm, the presence of individuals with outlier runtimes can have a great variance. The effect of this variance is most noticeable in the high relative standard deviation of the runtimes in the case of 16 cores. The loss in runtime due to this idle time is growing with the number of applied processor cores since adding extra cores only increases the number of unutilized cores.

3.7. Summary of Answers to the Research Questions of the Dissertation

Based on the results of the literature reviews, algorithm design and numerical efficiency tests presented in the previous subsections of Section 3, research question K1-K5 of my dissertation can be answered.

K1. The first research question dealt with the extension of HGHS to a GAM framework in terms of decision variables and the representation of individuals. With the help of the thin plate splines introduced in Section 3.1., I showed in Section 3.3. that the selection of order and breakpoints of spline functions can be automated in GAMs. With this result, the decision variables and binary representation of individuals in the linear case of HGHS can be preserved for GAMs as well.

K2. The second research question addressed the quality of models proposed by HGHS. Based on the results of the numerical experiments in Sections 3.4. and 3.5., it can be concluded that prediction accuracy on the test sets of the models proposed by the HGHS is not significantly different from that of the models proposed by the other feature selection algorithms examined in my dissertation. Furthermore, the HGHS is the only one of the examined algorithms that can propose models with feature sets that are completely free of concurvity on both examined real-world datasets. This result cannot be matched by the other two algorithms that control for concurvity, the mRMR and the HSIC-Lasso, since they only address the redundancy between variable pairs. In case of the Concrete Comprehensive Strength Dataset, the mRMR algorithm proposed a concurvity-free feature subset (see Table 2 of the dissertation), but the prediction accuracy of the proposed model is smaller than that of the model proposed by the HGHS. In case of the Concrete Comprehensive Strength Dataset, the results are robust regarding all the 9 performance metrics that are applied (see Table 2 of the dissertation). In case of the Credit Card Default Dataset, the results are more diverse. In this dataset, if we examine, the two models that have a feature set without concurvity (the HGHS and the Decision Tree), we can find that in prediction accuracy on the test set, the HGHS has better performance in “recall-like” metrics, while the Decision Tree has stronger “precision-like” metrics. In analyses predicting credit risk, the “recall-like” metrics are more relevant for decision makers according to Moula (2017) and the HGHS has better performance regarding these metrics (see Table 7 of the dissertation). So, it can be suggested that non-redundant features influencing credit risk should be identified based on the results of a HGHS model rather than a decision tree model.

K3. Research question three tackles the properties of the recombination operators of the HGHS. Based on the results in Section 3.4., I can conclude that when applying the HGHS in a GAM framework, it is preferred to generate completely random new individuals during the iterations, but inheritance from the previous population should not be neglected. In fact, it is also preferred to continuously increase the probability of inheriting an individual from the better-than average individuals of the previous memory/population.

K4. Research question four examines the optimal memory (or population) size and the stopping criterion of the HGHS algorithm. Results of the ceteris paribus sensitivity analysis presented in Section 3.4. highlight that in case of the optimal parameter set of HGHS, the algorithm can find the global optima or the second-best solution before mapping 40% of the search space in more than third of trials. Based on this observation, we can conclude that the poor quality of initial memory/population should be addressed by running the algorithm from several initial random populations rather than running the algorithm once with large population/memory size.

K5. The last research question addresses the parallelization strategies of the HGHS algorithm. Based on the results of Sections 3.5. and 3.6., it is preferred to simultaneously compute the models corresponding to each individual in HGHS. With the help of parallelization, the expected runtime of the HGHS is not significantly higher than that of the Random Forest combined with RFE algorithm on the larger Credit Card Default Dataset. Furthermore, expected runtime of the HGHS is significantly shorter than that of the GAMBoost (see Table 6 of the dissertation). Scalability of the HGHS is numerically examined on virtual architectures with 4, 8, 12 and 16 processor cores. Based on the empirical results, the HGHS achieves the maximal theoretical speed-up according to Amdahl’s law in case of 4 cores, but it fails to meet these maximum speed-ups in case of more cores. The explanation behind this phenomenon is the optimal parametrization of HGHS that prefers the random elements of the algorithm that causes idle time in some iterations. This idle time also causes high relative standard deviation of runtimes on the hardware configuration with 16 cores (see Table 9 and Figure 20 of the dissertation).

Based on the literature reviews and discussion, the HGHS algorithm proposed in my dissertation is the only metaheuristic solution to the feature selection task in a GAM framework that applies constraints to ensure a redundancy-free feature set in its proposed final model in a multivariate way, not just controlling for redundancy between variable pairs.

Based on results of the dissertation, the HGHS is a preferrable feature selection method to the other algorithms examined in the dissertation, if our aim is to build a sparse model and we have no serious time constraints.

4. Main References

Abdel-Basset, M., El-Shahat, D., El-henawy, I., de Albuquerque, V. H. C., & Mirjalili, S. (2020). A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. Expert Systems with Applications, 139, 112824.

Amdahl, G. M. (1967, April). Validity of the single processor approach to achieving large scale computing capabilities. In Proceedings of the April 18-20, 1967, spring joint computer conference (pp. 483-485).

Augustin, N. H., Sauleau, E. A., & Wood, S. N. (2012). On quantile quantile plots for generalized linear models. Computational Statistics & Data Analysis, 56(8), 2404-2409.

Belitz, C., & Lang, S. (2008). Simultaneous selection of variables and smoothing parameters in structured additive regression models. Computational Statistics & Data Analysis, 53(1), 61-81.

Binder, H., & Tutz, G. (2008). A comparison of methods for the fitting of generalized additive models. Statistics and Computing, 18(1), 87-99.

Bryant, R. E., David Richard, O. H., & David Richard, O. H. (2016). Computer systems: a programmer's perspective. Third Edition. Upper Saddle River: Prentice Hall.

Cantoni, E., Flemming, J. M., & Ronchetti, E. (2011). Variable selection in additive models by non-negative garrote. Statistical modelling, 11(3), 237-252.

Chong, I. G., & Jun, C. H. (2005). Performance of some variable selection methods when multicollinearity is present. Chemometrics and intelligent laboratory systems, 78(1-2), 103-112.

Climente-González, H., Azencott, C. A., Kaski, S., & Yamada, M. (2019). Block HSIC Lasso: model-free biomarker detection for ultra-high dimensional data. Bioinformatics, 35(14), i427-i435.

Du, M., Liu, N., & Hu, X. (2019). Techniques for interpretable machine learning. Communications of the ACM, 63(1), 68-77.

Etaati, L. (2019). Data Science Virtual Machine and AI Frameworks. In Machine Learning with Microsoft Technologies (pp. 273-285). Apress, Berkeley, CA.

Goldberg, D. E., (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Boston, Kluwer Academic Publishers.

Hall, M. A. (1999). Correlation-based feature selection for machine learning. Doctoral Dissertation. University of Waikato.

Hastie, T. J., & Tibshirani, R. J. (1990) Generalized Additive Models. London: Chapman and Hall.

Hastie, T. J., Tibshirani, R., & Friedman, J. (2011). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). NYC, NY: Springer.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. Management Information Systems Quarterly, 28(1), 6.

Huo, X., & Ni, X. (2007). When do stepwise algorithms meet subset selection criteria?. The Annals of Statistics, 870-887.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning: with applications in R. New York :Springer.

Jia, J., & Yu, B. (2010). On Model Selection Consistency of the Elastic Net. Statistica Sinica, 20, 595-611.

Krömer, P., & Platoš, J. (2016, July). Genetic algorithm for entropy-based feature subset selection. In 2016 IEEE Congress on Evolutionary Computation (CEC) (pp. 4486-4493). IEEE.

Lin, Y., & Zhang, H. H. (2006). Component selection and smoothing in multivariate nonparametric regression. The Annals of Statistics, 34(5), 2272-2297.

Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. Applied Mathematics and Computation, 188, 1567-1579.

Mafarja, M. M., & Mirjalili, S. (2017). Hybrid whale optimization algorithm with simulated annealing for feature selection. Neurocomputing, 260, 302-312.

Marra, G., & Wood, S. N. (2011). Practical variable selection for generalized additive models. Computational Statistics & Data Analysis, 55(7), 2372-2387.

McFadden, D. (1974). Conditional logit analysis of qualitative choice behaviour, in: P. Zarembka (ed.), Frontiers in Econometrics, Academic Press, New York, 105-142.

Molnar, C. (2020). Interpretable machine learning. Leanpub.

Moula, F. E., Guotai, C., & Abedin, M. Z. (2017). Credit default prediction modeling: an application of support vector machine. Risk Management, 19(2), 158-187.

Sayed, G. I., Tharwat, A., & Hassanien, A. E. (2019). Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection. Applied Intelligence, 49(1), 188-205.

Schmid, M., & Hothorn, T. (2008). Boosting additive models using component-wise P-splines. Computational Statistics & Data Analysis, 53(2), 298-311.

Signoretto, M., Pelckmans, K., & Suykens, J. A. (2008). Functional ANOVA Models: Convex-concave approach and concurvity analysis (No. 08-203). Internal Report.

Song, L., Smola, A., Gretton, A., Bedo, J., & Borgwardt, K. (2012). Feature selection via dependence maximization. Journal of Machine Learning Research, 13(5), 1393-1434.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267-288.

Umlauf, N., Adler, D., Kneib, T., Lang, S., & Zeileis, A. (2015). Structured Additive Regression Models: An R Interface to BayesX. Journal of Statistical Software, 63(21), 1-46.

Venables, W. N. & Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth Edition. New York :Springer.

Wang, Y., Liu, Y., Feng, L., & Zhu, X. (2015). Novel feature selection method based on harmony search for email classification. Knowledge-Based Systems, 73, 311-323.

Wood, S. N. (2017) Generalized Additive Models: An Introduction with R (2nd edition). Chapman and Hall/CRC.

Wooldridge, J. M. (2016). Introductory econometrics: A modern approach. Nelson Education.

Yang, S., & Zhang, H. (2018). Comparison of several data mining methods in credit card default prediction. Intelligent Information Management, 10(05), 115.

Yeh, I. C. (1998). Modeling of strength of high-performance concrete using artificial neural networks. Cement and Concrete research, 28(12), 1797-1808.

Yeh, I. C., & Lien, C. H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, 36(2), 2473-2480.

Zhao, P., & Yu, B. (2006). On model selection consistency of Lasso. The Journal of Machine Learning Research, 7, 2541-2563.

5. Personal Publications Regarding the Topic

Referred Scientific Journals

Láng, B. & Kovács, L. (2014). Linear Regression Model Selection using Improved Harmony Search Algorithm. *SEFBIS Journal*, 9(1), pp. 15-22.

Láng, B., Kovács, L. & Mohácsi, L. (2017). Linear Regression Model Selection using a Hybrid Genetic - Improved Harmony Search Parallelized Algorithm. *SEFBIS Journal*, 11(1), pp. 2-9.

Kovács, L. (2019). Applications of Metaheuristics in Insurance. *Society and Economy*, 41(3), pp. 371-395.

Kovács, L. (2021). Változószelekció általánosított additív modellben metaheurisztika segítségével. *SZIGMA Matematikai-közgazdasági folyóirat*. (Accepted, publication in progress)

Refereed Conference Papers

Kovács, L. (2021). Performance Testing of Feature Selection Algorithms for Generalized Additive Models. Proceedings of the 16th International Symposium on Operational Research: SOR '21. (Accepted, publication in progress)