DOCTORAL THESIS

FOR THE DEGREE OF DR. RER. NAT.

RELEVANCE LEARNING FOR REDUNDANT FEATURES

LUKAS PFANNSCHMIDT

Bielefeld University,
Faculty of Technology,
Machine Learning Group

19.11.2021

Dissertation submitted to the Faculty of Technology of Bielefeld University to obtain the academic degree: Doctor rerum naturalium (Dr. rer. nat.).

The contents of this work were presented to the PhD committee on the 23. August 2021. The committee was composed of Prof. Dr. Alexander Schönhuth, Prof. Dr. Barbara Hammer, Prof. Dr. Benoît Frénay and Dr. Basil Ell.

The *Smart Thesis* template used in this document was written by Jan Philip Göpfert and Andreas Stöckel and is inspired by the *Classic Thesis* template developed by André Miede.

The source code of all described software tools is available at:
`https://github.com/lpfann`

Printed on permanent, non-aging paper according to DIN ISO 9706.
Gedruckt auf alterungsbeständigem Papier DIN ISO 9706.

ABSTRACT

Feature selection is a widely used strategy in machine learning for the reduction of feature sets to their relevant essence to improve predictions and performance. It is also employed for knowledge discovery in applied disciplines such as biology and medicine to find potentially causal factors. But machine learning models often do not represent a unique solution to a given problem, especially in high dimensional settings where redundant factors are likely and spurious correlations exist.

Basing decisions about causal elements on feature selection is therefore inaccurate or wrong when not considering the presence of redundant but also relevant features. Most existing selection algorithms are specifically removing redundancies and not suitable for the task of all-relevant feature selection, or they require careful parametrization and are hard to interpret, which makes them difficult to use.

This thesis is focused on feature selection methods for the analytical use case to facilitate understanding of potential causal factors, for linear and non-linear problems. We propose several new algorithms and methods for all-relevant feature selection to improve knowledge discovery, enabled by statistical methods to improve the accuracy of existing solutions and allow the differentiation between different types of relevance. Furthermore, we offer a new heuristic to automatically group related features together, and we analyse the definition of relevance in the context of privileged information, where data is only available in training.

We also introduce software implementations, which were specifically designed to be modular, efficient and able to parallelize for applications in high dimensional problems. The methods and implementations were evaluated on a wide range of synthetic and real datasets to show their performance in comparison with existing algorithms.

ACKNOWLEDGEMENTS

This thesis not only represents the culmination of a lot of academic work but also the time spent together with awesome people. In no particular order, I want to thank all the people who accompanied, supported and motivated me on my way.

The several groups I was part of, which were there for academic discussions, but most were cool colleagues and friends. The members of DiDy with regular tournaments in the 'Printer Room'. The people from the Datamining group at SFU and the amazing sea to summit hike. The Machine Learning group with the best Christmas parties in the CITEC (and in general). I thank all the co-authors and contributors working together with me. Most important – or with strong relevance – for this thesis' completion was Barbara Hammer, who kept me going forward and always spread optimism.

While doing a PhD is time-consuming, there is still time spent outside the university. For these times, I want to thank all my friends which helped me take my mind of nagging problems and my family for supporting me ever since.

Most grateful I am for Hannah, who always lifted my spirits, stuck with me even when continents apart and who is the best partner I could hope for.

CONTENTS

1 INTRODUCTION

1.1 MOTIVATION

At the advent of a new decade comes the time to reflect what happened in the last and to imagine what lies ahead in the dawning one. In the last decade, we could observe the maturing of digital technologies and computers, earlier perceived as clunky desktop tools in offices now growing into small, versatile companions in everybody's pockets. Together with the compact form factor also came the inclusion of digital sensors, monitoring a multitude of different modalities. An example being the acceleration sensor in mobile phones often used to infer steps taken by the owner every day and therefore predicting activity and health. The number of computers in all possible forms will increase to the point where it is not even visible from the outside to recognize one, and with it, the number of sensors, collected information and possibilities.

With the growing amount of information also come new challenges for computer science, trying to make sense of raw and mostly noisy data. These challenges could already be observed in Bioinformatics, the research field combining computer science and biology, due to increasing availability of highly sensitive biotechnologies, as well as the increasing digitalization of biomedical diagnostics. There, one could observe a trend towards big and complex but also much more capable machine learning models. These models try to learn an unknown relation between input data and a known outcome and are used successfully in medical diagnoses such as cancer prediction [Kon01; BZ08; CW06] or Covid-19 screening [LHC20]. The input data consists of multiple variables or features, and their true meaning is often unknown, but sometimes these features directly correspond to established markers, known as biomarkers. Knowing the true semantic meaning of a feature is important in research, a recent important example being the analysis of pandemic dynamics, where certain COVID-19 risk groups could be identified through such markers [Rod+20].

Even when specific biomarkers or the semantic meaning of features are unknown, many learning models can perform on raw data without preselection of variables. While still achieving a high prediction accuracy, they are less suited for data exploration and understanding of the underlying relationships. For the latter, insight into the model behaviour and its relevant driving factors is necessary [Vel+12] and selecting and identifying features constitute the first steps to unravel the underlying relationships by allowing interpretability. Many predictive models are not easily interpretable even in the case of low dimensional data, or they suffer from a low ability to generalize in high dimensional settings such that the information about features is not transferable. A paradigm for interpretable modelling, which alleviates those shortcomings for the analytical use case, is necessary.

Feature Selection (FS) represents a prominent paradigm which enables the inference of sparse and interpretable prediction models [GE03]. Sparse meaning that not all features are considered and interpretability being a fuzzy measure [BF16] often defined regarding a human overseeing a model and understanding the relationship. Interpretability is then often achieved by just having a smaller set of relevant features to consider for a human observer. Most of the existing methods enforce sparsity through the removal of totally irrelevant features as well as redundant features, which do not contain information improving the prediction of the model.

But, the removal of redundancies is contrary to the goal stated earlier and not an improvement in the case of high dimensional problems, because many machine learning solutions use optimizations with non-unique solutions. Thus, in the presence of redundant but relevant features, the selected feature set often depends on arbitrary initialization or algorithmic design choices. Hence, possibly relevant but redundant features can easily be overlooked, even though redundant features are getting more common with a growing number of sensors. The all-relevant feature selection problem (ARFS) deals with the challenge to determine all features, which are potentially relevant for a given task, introduced by *Kohavi et al.* [KJ97] in the 90s. The all-relevant feature set represents a good foundation for knowledge discovery because it explicitly includes all possibly relevant features, such as used in alternative hypotheses. Identifying the subset with all-relevant features is generally computationally intractable but approximations exist.

This thesis is concerned with the evaluation, application and extension of those ARFS heuristics for the analytical use case in a wide range of data modalities and problem types such as linear classification and ordinal regression, non-linear classification and regression, or privileged information and classical learning settings. While taking advantage of existing approaches we also introduce several methodic extensions to improve their results such as a statistical feature selection threshold and automatic grouping of related features. We present new software implementations with visualization of important elements and runtime improvements by parallelization.

1.2 RESEARCH QUESTIONS

In this thesis, we are going to answer several research questions:

> **RQ 1**: How to uncover all relevant features in a machine learning setting, where a degree of redundancy in the feature space is present, with *high precision* and *efficient* runtime?

This setting is the theoretical problem of ARFS with a focus on usable methodology in several scenarios. We cover the answers to this question for linear classification in Chapter 3, for ordinal regression in

Chapter 4, and in Chapter 5 for non-linear classification and regression problems. For this question, we predominantly seek an algorithmic output of binary feature relevance for each input feature leading to the determination of the all-relevant subset.

> **RQ 2**: Can we distinguish weakly from strong relevant features, and can we assess the relation of weakly relevant ones?

This is answered in Chapters 3 to 5 by exposing the class of feature relevance for each input feature. As such, we extend the binary output of relevance as in RQ 1 to include weak relevance. In the following, we denote this ternary measure as the feature class. Furthermore, we propose a clustering method for related features in Chapter 3 resulting in feature groups.

> **RQ 3**: Can the relevance of privileged features in a Privileged Information (PI) setting be computed similarly to regular features?

In addition to the classical machine learning setting, where the model's set of input features at the time of training is identical to the time of prediction, we also regard the scenario in which privileged information is used exclusively at the training stage. A definition of relevance, methodology and evaluation for this is given in Chapter 4 in the context of ordinal regression.

> **RQ 4**: How does feature selection in the context of redundancies perform on real data from the biomedical domain?

This question represents the initial idea for this thesis, which seeks applicability of useful theoretical models in the biomedical domain. It is answered in many evaluations using real data by comparing models popular in this domain and by proposing new methods and practical implementations such as in Chapter 3.

Because we consider various settings and problems in this thesis, we are going to use a table at the beginning of the relevant sections to improve understanding. Table 1.1 shows the possible values and relevant chapters. Not shown is Chapter 2, which contains basic definitions and related methods for the ARFS problem. The problem aspect considers binary classification, the extension of it to several classes on an ordinal scale (ordinal regression) and prediction of continuous values in regression. The model row differentiates between two model capabilities: those who are limited to linear relationships and those who can also represent non-linear dependencies. The type of machine learning refers to the classical and privileged setting described in RQ 3. For the output, we consider the binary relevance, the ternary feature class including the weak relevance, and the clustering of these into feature groups.

Table 1.1: Overview of aspects considered in this thesis.

All-Relevant Feature Selection

Chapter	3	4	5
Problem	classification	ordinal regression	classification, regression
Model	linear	linear	non-linear
Type	classical	classical, privileged	classical
Data	synthetic, real	synthetic, real	synthetic, real
Output	relevance, feature class, feature groups	relevance, feature class	relevance, feature class

Within the frame of the thesis, the following publications could be presented to an international audience:

- Christina Göpfert, Lukas Pfannschmidt, and Barbara Hammer. "Feature Relevance Bounds for Linear Classification". In: *Proceedings of the ESANN, 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. Ed. by Michele Verleysen. Bruges: Ciaco - i6doc.com, 2017, pp. 187–192
- Christina Göpfert, Lukas Pfannschmidt, Jan Philip Göpfert, and Barbara Hammer. "Interpretation of Linear Classifiers by Means of Feature Relevance Bounds". In: *Neurocomputing* 298 (July 12, 2018), pp. 69–79. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2017.11.074
- Lukas Pfannschmidt, Christina Göpfert, Ursula Neumann, Dominik Heider, and Barbara Hammer. "FRI – Feature Relevance Intervals for Interpretable and Interactive Data Exploration". In: *2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. July 2019. DOI: 10.1109/CIBCB.2019.8791489. arXiv: 1903.00719
- Lukas Pfannschmidt, Jonathan Jakob, Michael Biehl, Peter Tino, and Barbara Hammer. "Feature Relevance Bounds for Ordinal Regression". In: *ESANN 2019*. ESANN 2019. Bruges: i6doc, Feb. 20, 2019, ES2019–162. ISBN: 978-2-87587-065-0. URL: https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2019-162.pdf
- Lukas Pfannschmidt, Jonathan Jakob, Fabian Hinder, Michael Biehl, Peter Tino, and Barbara Hammer. "Feature Relevance Determination for Ordinal Regression in the Context of Feature Redundancies and Privileged Information". In: *Neurocomputing* (Apr. 9, 2020). ISSN: 0925-2312. DOI: 10.1016/j.neucom.2019.12.133. arXiv: 1912.04832
- Lukas Pfannschmidt and Barbara Hammer. "Sequential Feature Classification in the Context of Redundancies". In: Apr. 15, 2020. arXiv: 2004.00658. URL: http://arxiv.org/abs/2004.00658

2 FOUNDATIONS ON FEATURE SELECTION

2.1 FEATURE RELEVANCE

Is a feature relevant?

This question alone, without further specification, can not be answered. Despite the fact of its crucial role in many sciences, the implicit assumptions needed to answer it are often not specified and can lead to wrong conclusions. In the following, we give a short overview of these, sometimes competing, assumptions.

Let X be data set $X := \{\boldsymbol{x}_i \in \mathbb{R}^d; i = 1, \dots, n\}$ with n samples and with $\mathcal{D} := \{\ell \in \mathbb{Z}; \ell = 1, \dots, d\}$ as the set of all features such that cardinality $|\mathcal{D}| = d$. The target variable $\boldsymbol{y} \in \mathbb{R}^n$ is distributed according to some potentially unknown function dependent on X such that $g(X) = \boldsymbol{y}$.

There are multiple ways to define feature relevance and compute it. In general, relevance can be represented as a binary value (relevant, irrelevant) and also as a quantitative value which denotes importance in relation to other features. We consider feature relevance for:

1. a single function,
2. a set of functions or
3. all possible functions.

In case 1 the relevance of a feature is given by its presence or usage in an estimating function (or model) f. The usage can be inferred by observing the function parameters, such as the coefficients c in a simple linear model:

$$f(\boldsymbol{x}_i) = c_1 \cdot x_{i,1} + c_2 \cdot x_{i,2} + c_3 \cdot x_{i,3}$$

A feature would then be considered irrelevant if the coefficient is zero. This also allows a relative measure of relevance when observing the coefficients themselves, where bigger coefficients could be considered as more relevant, given a proper normalization in the preprocessing. If the function is not so simple, a direct attribution can be challenging.

As a proxy to direct attribution, we can instead observe the loss function. The loss function quantifies the deviation from the true function when approximating it with function f. Because g is not available to us, we instead measure the absolute deviation from the samples of g given by $\boldsymbol{y}$. Consider an exemplary loss for a regression problem as

$$\mathcal{L}(X, \boldsymbol{y}, f) := \sum_{\forall i} |f(x_i) - y_i|$$

Now, to measure a feature's usage in a model we can modify its contained information and check this loss. We define $\text{perm}(X_\ell)$ as the random permutation of values in X_ℓ and $X \diamond \ell$ as the dataset where X_ℓ was replaced by its random permutation. Then we consider ℓ as not present or removed. If we observe $\mathcal{L}(X, \boldsymbol{y}, f) = \mathcal{L}(X \diamond \ell, \boldsymbol{y}, f)$, i. e. the loss without ℓ present is identical, ℓ can be considered as irrelevant. If we observe increased loss $\mathcal{L}(X, \boldsymbol{y}, f) < \mathcal{L}(X \diamond \ell, \boldsymbol{y}, f)$, we can deduce ℓ was used by the function and consider it as relevant and further, assign it a positive relevance measure depending on the absolute loss deviation. This view is analogous to the definition of relevance stated in [BL97] or in probabilistic terms in [Nil+07].

In machine learning we do not know the function f beforehand, instead, we use optimization to find a function given an objective to minimize a loss function. The optimal function in the set of all available functions $\mathcal{F}$ is then

$$f^\star := \arg\min_{f \in \mathcal{F}} \mathcal{L}(X, f),$$

in the case of a *unique* solution. If we also consider other functions with a similar or identical loss we get the set

$$\mathcal{F}^\star := \{h \in \mathcal{F} \mid \mathcal{L}(X, h) \approx \mathcal{L}(X, f^\star)\}.$$

As we now consider sets of functions, the question of relevance shifts to case 2.

We can abstract from a specific subset of functions and consider all possible functions as in information theory [CT91]. Information theory purely considers the information content of variables, based on their statistical distributions, without reasoning about functions themselves. One of the tools in information theory is mutual information (MI) [Sha48]. It's defined for two variables A and B as

$$\mathrm{MI}(A; B) = D_{\mathrm{KL}}(P_{(A,B)} \| P_A \otimes P_B),$$

where D_{KL} denotes the Kullback–Leibler divergence using the joint distribution $P_{(A,B)}$ and the marginal distributions P_A and P_B. We could compute this for a feature ℓ and target y and if $\mathrm{MI}(X_\ell; y) > 0$, we could consider the feature as relevant [Bat94]. In theory, this approach allows the best general reasoning about relevance, but computing and approximating MI is challenging especially in high dimensional settings and applying it can lead to problems such as in the presence of label noise [FDV14], in some regression settings with specific conditional estimation error [FDV13a] or when using it as a selection criterion in classification [FDV13b].

Because of the specific needs in biomedicine, we attempt to balance the need for predictive accuracy, interpretable and truthful relevancies, and computational performance. Thus, for the following thesis, we limit ourselves to the definition of relevance for a set of functions. In our case, we specifically analyse and evaluate relevancies for two popular classes of functions, namely linear SVM-like models, and Random Forests (RFs).

Redundancies In the literature of applied fields often the scope of relevance is not clearly stated. Such as when only the relevance of a unique solution given by the training data is considered [ZSR02], even though this solution can vary greatly when new data is encountered, especially in the case of collinearity [Dor+13]. This can lead to wrong conclusions about feature relevancies when correlated or identical features are present, which we refer to as redundant features. Redundant features are pairs or groups of features with a partial or complete overlap of information, i. e. a pair of features ℓ and k is (in part) redundant when $\mathrm{MI}(X_\ell; X_k) > 0$.

Because we often do not know the semantic meaning or true function of input features, reasoning about cause and effect as described by Pearl in [Pea09] should be done carefully when employing machine learning, especially when considering spurious correlation. We consider spuriously correlated features as having overlapping information even though no causal relationship exists. Spurious correlation between two or more features can occur just by chance, which gets increasingly likely when considering high dimensional data sets [FZ16], or with limited data with an unknown latent variable affecting both features.

2.2 FEATURE SELECTION

Unlike relevance determination, which assigns real values to the features, feature selection aims for a discrete subset of features which suffice to provide the relevant information. It can be viewed as an extension of relevance determination with a decision threshold, which determines a feature's inclusion in such a subset based on its relevance. Here, the additional challenge of finding a robust threshold arises, which should discriminate between irrelevant noise and relevant features. Furthermore, feature selection differs in the choice of including or removing redundant features which results in two different types of feature selection subsets.

Before we define the problems related to the finding of those sets, we define several classes of feature relevance similar to [KJ97] in terms of their use in an optimal model. Again, we consider the set of all (near)-optimal functions $\mathcal{F}^\star := \{h \in \mathcal{F} \mid \mathcal{L}(X, h) \approx \mathcal{L}(X, f^\star)\}$. Essential features, which have no mutual information with others, are denoted as strong relevant features with the set of all strongly relevant features

$$\mathcal{S} := \{\ell \in \mathcal{D} \mid \forall f \in \mathcal{F}^\star : \mathcal{L}(X, y, f^\star) < \mathcal{L}(X \diamond \ell, y, f)\},$$

where *all* optimal functions show increased loss without a feature in $\mathcal{S}$.

All features who have information overlap or redundancies with at least one other feature are called weakly relevant features. Let us consider a subset $\mathcal{D}_* \subset \mathcal{D}$ and define

$$\mathcal{W} := \{\ell \in \mathcal{D} \mid \exists f \in \mathcal{F}^\star : \mathcal{L}(X \diamond \mathcal{D}_*, y, f) < \mathcal{L}(X \diamond \ell, y, f) \approx \mathcal{L}(X, y, f^\star)\}$$

as the set of weakly relevant features, where the loss of a single ℓ does not decrease model performance, but a subset $\mathcal{D}_*$ exists where this is the case. Here, $\mathcal{D}_*$ represents the case where all features with mutual information with ℓ and ℓ itself are permuted. Not all features in $\mathcal{W}$ together need to have common mutual information, e. g. multiple redundant feature pairs could exist, with no mutual information between different pairs. The union of $\mathcal{S}$ and $\mathcal{W}$ is called the set of all-relevant features $\mathcal{A} := \mathcal{S} \cup \mathcal{W}$. Lastly, there are irrelevant features ($\mathcal{I}$) which are neither strong nor weakly relevant.

In general, the task of FS is defined as finding a subset of all features in a way that the loss of information is minimal and the reduction in set size is maximal. It's utilized to reduce model complexity to improve generalization ability and computational efficiency. Further, it can reduce costs, e. g. by minimizing necessary clinical sampling for predictive purposes or in general by reducing the necessary sensor resolution. It is also especially important in applications where interpretability and accountability are essential by highlighting relevant elements in potentially highly complex data.

Problem definitions Given the different scopes of feature relevance from Section 2.1 we also encounter different problems when wanting to select features according to their relevance. Kohavi and John [KJ97] first identified these different types in terms of an optimal Bayes classifier.

In general, feature selection is the task of finding a function

$$f_s(X, y) = \mathcal{D}_s$$

which takes input data and outputs a set of relevant features $\mathcal{D}_s$. The output set is defined by

$$\hat{\mathcal{D}}_s := \{\ell \in \mathcal{D} \mid \mathcal{D}_s \text{ fulfils feature selection criterion}\}$$

where the criterion differs between the selection problems.

The most popular type of feature selection is concerned with finding the minimal-optimal feature set for a set of all functions $\mathcal{F}$ and its optimal function $f^\star$ and represents case (2) from Section 2.1.

Definition 2.2.1 (MFS)**.** Minimal-optimal feature selection seeks a feature subset $\mathcal{M}$ out of all features $\mathcal{D}$ with the smallest possible size such that predicting y with optimal function $f^\star$ is still possible, i. e. no crucial information is lost and $\mathcal{L}(X^{\mathcal{D}}, \boldsymbol{y}, f^\star) = \mathcal{L}(X^{\mathcal{M}}, \boldsymbol{y}, f^\star)$.

Problem 2.2.1 is often performed by removing irrelevant features and any kind of redundant information such as duplicates. If the optimal function $f^\star$ is unique, the resulting feature set also represents true relevance if membership of the set is regarded as a binary relevance measure. In many cases, especially in high dimensional problems with many redundant features, $f^\star$ is not unique and thus the solution gives no true relevance.

Definition 2.2.2 (ARFS)**.** All-relevant feature selection seeks a feature subset $\mathcal{A}$ out of $\mathcal{D}$ including all relevant features with information about y and smallest possible size.

This problem attempts to preserve all relevant features, including redundancies, and therefore correctly represents the true relevance for all possible functions in $\mathcal{D}$. Solving this problem is much harder than finding $\mathcal{M}$ though and in general, requires an exponential amount of time to perform an exhaustive subset search although polynomial-time algorithms exist for constrained sub-problems [Nil+07]. While the goal of all-relevant feature selection is finding all features belonging to $\mathcal{A}$, it's not identifying the detailed composition

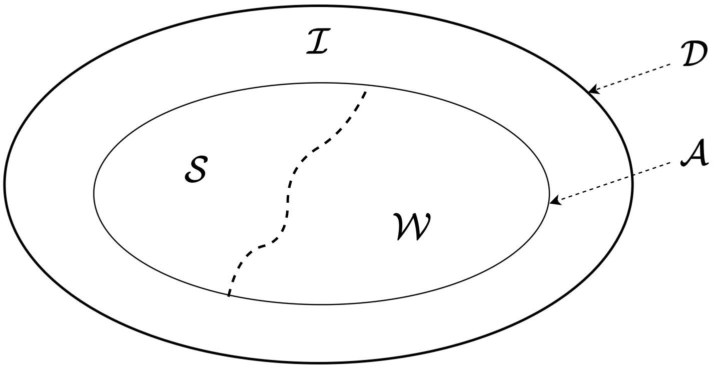

Figure 2.1: Representation of feature (sub-) sets considered in this thesis. The set of strongly relevant features ($\mathcal{S}$), the set of weakly relevant features ($\mathcal{W}$), irrelevant features ($\mathcal{I}$), all relevant features ($\mathcal{A}$) and all input features ($\mathcal{D}$). Not pictured is the minimal-optimal set ($\mathcal{M}$).

of $\mathcal{S}$ and $\mathcal{W}$ as pictured in Figure 2.1. In other words, in the general problem, the feature selection algorithm does not discriminate between strong and weak relevance.

In this thesis, we are presenting several all-relevant feature selection algorithms, which limit the domain to a specific class of model to overcome the computational limitations. Furthermore, we also present methods to perform the decomposition into $\mathcal{S}$ and $\mathcal{W}$.

2.2.1 *Approaches*

To solve the problems described in Section 2.2 there exists a wide range of methods [GE03; KJ97]. The majority of existing approaches are solving the minimal-optimal feature selection problem (MFS) whereby only a handful is solving ARFS.

In general, feature selection methods can be summarized into three categories:

Filter Filter methods are based on the information content of features in relation to the target, disregarding a specific model. They are based on the relevance type 3 from Section 2.1. They can be very efficient and fast and thus particularly suited as a screening technology for high dimensional data [YL03]. One common filtering measure is the Pearson correlation which is extremely performant but can not capture all possible non-linear dependencies between feature and target. As mentioned earlier, the general measure of mutual information theoretically does not have this limitation and heuristics such as *MIFS* have been already proposed nearly 30 years ago [Bat94]. Estimation of MI is not trivial can lead to incorrect results, which was shown for MFS on classification data [FDV13b], where the subset with maximal mutual information did not produce the optimal model. An advantage of mutual information is the handling of multiple features at once for higher-order dependencies. This allows the correct representation of relationships where multiple features are required, such as in the classical XOR problem.

Wrapper Wrapper approaches perform decisions on the feature set while evaluating a model on a predefined measure [KJ97]. An example with loss function as the measure is

$$\mathcal{D}_s := \operatorname*{arg\,min}_{\mathcal{D}_s^* \subseteq \mathcal{D}} \mathcal{L}(X, \mathcal{D}_s^*, f)$$

where only the features in the candidate set $\mathcal{D}_s^*$ would be used in X and the set with minimal loss would be selected. Note that exhaustively evaluating all possible subsets $\mathcal{D}_s^* \subseteq \mathcal{D}$, with $2^{|\mathcal{D}|}$ possible candidates, is computationally intractable for common data set sizes. Several heuristics exist to improve upon this, such as greedy search methods which employ the strategy of adding or removing features to or from a candidate set such as in Recursive Feature Elimination (RFE) [GMS17]. Starting from the complete set, RFE recursively chooses features with the smallest change in $\mathcal{L}$ until a termination condition is fulfilled, such as a desired set size.

Embedded Embedded approaches use the model parameters itself to find relevant features. They are integrated into the optimization of the model itself which can be more efficient. A popular example is the Lasso [Tib96] which can be used in linear models such as the Support Vector Machine (SVM). An exemplary and simplified optimization term is

$$(\tilde{\boldsymbol{w}}, \tilde{\chi}) \in \operatorname*{arg\,min}_{\boldsymbol{w}, \chi} \|\boldsymbol{w}\|_1 + \sum_{i=1}^{n} \chi_i$$

where the sum $\sum_{i=1}^{n} \chi_i$ represents the deviation from y. Minimizing the L_1-norm of parameter vector $\boldsymbol{w}$ leads small coefficients to converge to zero. The non-zero coefficients can then be selected as the set

$$\mathcal{D}_s := \{\ell \in \mathcal{D} \mid |\tilde{\boldsymbol{w}}_\ell| > 0\}.$$

In practice, the convergence is not perfect and thresholds other than zero are usually used.

Since embedded FS methods do not rely on iterative feature selection or weighting, embedded approaches have the benefit that they can effectively take into account interdependencies of groups of features if the model has those capabilities. While complex models and optimization schemes allow for wide applications of machine learning, the growing numbers of parameters make embedded approaches challenging. The relation between input features and model parameters is not always as trivial as the Lasso example suggests.

2.3 FEATURE SELECTION METHODS FOR POSSIBLY REDUNDANT FEATURES

In this thesis, we are focusing on solutions for the ARFS problem stated in Definition 2.2.2 and related challenges such as the distinction between strong and weak relevance. Finding an optimal solution to the ARFS problem is computationally intractable [Kum14] but several approximations and similar techniques exist.

In 2005 the ElasticNet (EN) [ZH05] improved upon the instability of sparse models [LeC+95] by using a combination of multiple different regularization terms which lead to better conservation of weakly relevant features in the model weights, but the original method is lacking a selection threshold such that only relevancy values are available. Because EN is simply a combination of regularization terms it can easily be used in a wide range of models for classification and regression.

The statistically equivalent signature (SES) [Lag+16] approach proposes a technology which groups mutually equivalent features into groups out of which minimal feature subsets can be constructed.

Another proposal called *stability selection* uses resampling for more robust selection especially in high dimensional problems [MB10; SS13].

In 2017 Neumann et al. presented the Ensemble Feature Selection (EFS) method which combined multiple existing FS methods to remove their individual biases and produced aggregated feature relevancies [NGH17; Neu+16] but no proper selection threshold and no distinction between strong and weak relevance.

Also noteworthy are methods from game theory such as the clustering of features using a Nash Stable Partition [GSS11].

In the scope of this thesis, we are using and extending two particular approaches for solving the ARFS problem which are described in Sections 2.3.1 and 2.3.2.

2.3.1 *Boruta*

Boruta is a heuristic wrapper approach using an Random Forest (RF) model internally [KR11; KR10]. The RF is an ensemble of tree models, each constructed using different features and samples. The relevance of input features can be measured by counting the number of feature inclusions in each subtree because the tree models themselves are grown by minimizing a loss function. The authors note the advantages of this ensemble model, where features are included randomly in the subtrees and the relevance of features is measured independently such that redundant features are not overshadowed by others and all redundant features exhibit a measurable relevance signal. Because irrelevant features can also be included by chance, Boruta utilizes statistical testing to discern between those and truly relevant ones. To create the statistic they extend the dataset randomly generated contrast variables as a reference, also known as shadow features. These shadow features are used to create a statistical testing threshold, which guides the discrimination between signal and noise features.

Originally, Boruta was presented and tested using an RF classifier but it can also work with an RF regressor. Alternatives to the Boruta method are discussed and evaluated in [DSS19], whereby Boruta was identified as best performing technology among the tested ones if used for different dimensionalities of the data.

2.3.2 *Feature Relevance Bounds for Linear Models*

Recently, Göpfert et al. discussed a novel approach to efficiently compute the relevance of features in the presence of feature redundancies. They investigated this for linear classification models, a particularly relevant setting in biomedicine [Göp+18; GPH17]. Their approach efficiently assigns relevance bounds to features, rather than simple coefficient values. These bounds mirror the range of possible weight coefficients of a feature when considering *all* possible models, hence offering detailed and complete information also in the case of feature redundancies. The relevance bounds, therefore, enable classification of features into relevance classes for solving the ARFS. The work done in [Göp+18; GPH17] represents the theoretical foundation of the feature relevance bounds. In this section, we are outlining these ideas.

Let us consider the task of binary classification. For a classification problem, we observe data

$$X = \Big\{(x_1, y_1), \ldots, (x_n, y_n) \ \in \mathbb{R}^d \times \{-1, 1\}\Big\} \tag{2.1}$$

with n samples and d real-valued features which have been tied to a target or response y by an unknown function. We assume that all d features have been standardized at mean zero and standard deviation 1.

It is common practice to evaluate the relevance of a feature for a given classification through the weights assigned to the feature by a linear classifier such as an SVM [CL08]. Sparsity in the feature set, which improves the interpretability of relevancies, can be emphasized by resorting to models such as Lasso or sparse SVM models [Tib96; Yao+17]. Yet, provided the solution is not unique as is often the case especially in high dimensional data, the resulting feature relevance is to some extent arbitrary, i. e. possibly rendering the model interpretation invalid.

Here Göpfert et al. [Göp+18] propose an alternative: instead of taking the weights of a single model as an approximation of feature relevancies they take multiple models into account, i. e. a class of models. A model class is characterized by its similarity in the quality of the solution, i. e. similar generalization ability as characterized by the size of the weight vector of the SVM [ABR00] and similar training loss, which measures wrongly classified samples. Through the use of a class of models, we can approximate the global solution of the ARFS problem as shown in [Göp+18] by introducing relevance bounds. They replace the original weight value as seen in a single model and introduce maximal and minimal possible weights, i. e. bounds of possible weightings per feature for all models with a similar characterization as given by the baseline solution.

Baseline Solution

Assume we are interested in linear classifiers of the form

$$y \mapsto \operatorname{sgn}(\boldsymbol{w}^\top \boldsymbol{x} - b)$$

where $\boldsymbol{w}$ is the normal vector of the separating hyperplane, b denotes the bias and sgn refers to the sign function.

The baseline solution is then given by an L_1-regularized soft-margin SVM:

$$\begin{aligned} \big(\tilde{\boldsymbol{w}}, \tilde{b}, \tilde{\chi}\big) &\in \underset{w, b, \chi}{\arg\min} \|\boldsymbol{w}\|_1 + C \cdot \sum_{i=1}^{n} \chi_i \\ \text{s.t. for all i} \qquad & y_i(\boldsymbol{w}^\top \boldsymbol{x}_i - b) \geq 1 - \chi_i \\ & \chi_i \geq 0 \end{aligned}$$

Through optimization, we acquire a model fully defined by the normal vector of a hyperplane $\boldsymbol{w}$ and its offset b from the origin. Prediction of samples is based on the signed distance from the plane. As usual for a soft margin model, χ_i are slack variables to guarantee the feasibility of the optimization problem in case of unavoidable classification errors, e. g. noise or error in the data. C is a regularization parameter which depends on the dataset's distribution. We choose the parameter guided by 3-fold stratified cross-validation and the F_1 weighted by each class support to account for possible class imbalances.

From the model with the best C, we obtain constraints for controlling the generalization error of equivalent models: the upper limit on the L_1 norm of the weight vector

$$\mu := \|\tilde{\boldsymbol{w}}\|_1$$

and error term

$$\rho := \sum_{i=1}^{n} \tilde{\chi}_i \, .$$

These values determine the class of equivalent classifiers, which consist of all SVM solutions $(\boldsymbol{w}', b', \chi')$ such that $\|\boldsymbol{w}'\|_1 \leq \mu$ and $\sum_i \chi_i' \leq \rho$. All these alternatives are considered equivalent since they show the same performance for the given classification task as the baseline solution. Hence, all weight vectors associated with an equivalent solution are relevant to determine the relevance of a feature to the given classification problem.

Minimum and Maximum Bounds for Linear Classifiers

Using these constraints we now define feature relevance bounds for every feature ℓ independently, i. e. we determine the interval of weight vectors resulting if we take into account the weights of all possible equivalent linear classifiers. More specifically, we want to compute extremal weight values for each feature given a similar error to the baseline.

For the lower bound, i. e. the lowest possible value of feature ℓ, we define the optimization problem

$$\begin{aligned} &\text{minRel}(X,\ell) : \min_{\boldsymbol{w},b,\chi} |\boldsymbol{w}_\ell| \\ &\quad \text{s.t. for all i} \\ &\qquad\qquad y_i\left(\boldsymbol{w}^\top x_i - b\right) \geq 1 - \chi_i, \quad \chi_i \geq 0 \\ &\qquad \text{and} \\ &\qquad\qquad \sum_{i=1}^{n} \chi_i \leq \rho \\ &\qquad\qquad \|\boldsymbol{w}\|_1 \leq (1+\delta)\cdot\mu \, . \end{aligned} \tag{2.2}$$

And the upper bound for ℓ is defined as

$$\begin{aligned} &\text{maxRel}(X,\ell) : \max_{\boldsymbol{w},b,\chi} |\boldsymbol{w}_\ell| \\ &\quad \text{s.t. for all i} \\ &\qquad\qquad y_i\left(\boldsymbol{w}^\top x_i - b\right) \geq 1 - \chi_i, \quad \chi_i \geq 0 \\ &\qquad \text{and} \\ &\qquad\qquad \sum_{i=1}^{n} \chi_i \leq \rho \\ &\qquad\qquad \|\boldsymbol{w}\|_1 \leq (1+\delta)\cdot\mu \, . \end{aligned} \tag{2.3}$$

The optimization problems can be rewritten as linear optimization problems and solved in polynomial time [Göp+18] using appropriate solvers. To account for their numerical inaccuracies the relaxation factor $\delta = 0.001$ allows minor deviations from μ.

2.4 SUMMARY

Several feature selection methods exist, which supposedly can handle redundant features through various means, such as a group preserving regularization. But many do not provide an actual method to select features according to the criterion stated in the ARFS problem.

The relevance bound method described in Section 2.3.2 is powerful and can solve the ARFS, but in practice, several problems such as numerical instabilities can occur which make the results noisy. Because of these instabilities, we present a statistical framework in Chapter 3 to improve feature selection results by providing a robust feature classification threshold and demonstrate its applicability in the context of biomedical data analysis. Furthermore, the original approach was limited to classification, which we extend to ordinal regression and privileged information features in Chapter 4. We also introduce an automatic grouping of weakly relevant features, which helps in highlighting possibly interesting relations.

Because the relevance bounds can only be efficiently solved in the case of linear models we also evaluate and extend the Boruta method to handle nonlinear problems in an efficient way. While Boruta can perform all-relevant feature selection, it does not discriminate between strong and weak relevance because it lacks actual relevance values. In Chapter 5 we evaluate a new approach to extend Boruta with such a discrimination approach.

3 APPLICATIONS OF FEATURE RELEVANCE BOUNDS

Parts of this chapter are based on:

Lukas Pfannschmidt, Christina Göpfert, Ursula Neumann, Dominik Heider, and Barbara Hammer. "FRI – Feature Relevance Intervals for Interpretable and Interactive Data Exploration". In: *2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. July 2019. DOI: 10.1109/CIBCB.2019.8791489. arXiv: 1903.00719

3.1 BACKGROUND

Context	
Problem	*classification*
Model	*linear*
Type	*classical*

In this chapter, we present an accessible and extended implementation of the feature relevance bounds method described in Section 2.3.2 Here, we only consider the classical setting of machine learning, without privileged information. We are also only considering relevance bounds for linear classification models. Even though linear models can not be applied to every problem, they are widely used in medicine [Hua+eb].

Together with the implementation in Python[1] we also worked on extending the methodology to improve feature selection accuracy by introducing a statistical feature selection threshold.

Furthermore, we use the relevance bounds in visualizing the model together with the relevance classes which enables interpretability which is important in biomedical research.

Specifically for the use case of biomarker discovery, we also propose a model refinement and design method with automation to provide features with similar functionality. Knowing about a group of features which could fulfil the same role in a model can be very important in the design of diagnostic tests where the source of data can differ by the cost or invasiveness of acquisition. Explicit redundancies of feature relevance then enable a practitioner to avoid features if they can be substituted by others. Additionally, feature groups could induce knowledge about novel biological relationships. This is especially useful in gene co-expression or metabolomics experiments where groups of functional units are common [vDam+18].

Overall, these additions can highlight elements which are crucial for the problem at hand while also providing alternatives which have the same information.

In summary, in this chapter, we seek answers to research questions 1, 2 and 4 from Section 1.2.

Figure 3.1 displays the structure of our proposed software pipeline called Feature Relevance Intervals method (FRI). We have already covered the original theoretical proposal in Section 2.3.2. In Section 3.2 we give details of our implementation. We describe how we classify each feature into three relevance groups and how to reduce false positives using a probe-based threshold estimation in Section 3.2.1. Then we show how we can constrain the use of features to certain relevance values (Section 3.2.2) to facilitate interactive data exploration and model design. In Section 3.2.3 we show how to automate this model design step to produce groups of related features. All these aspects are then evaluated in Section 3.3 quantitatively using simulated (Section 3.3.1) and biomedical data (Section 3.3.1). To check the correctness of the related feature groups, we perform a classic clustering analysis with

[1] Source code and Python package available at github.com/lpfann/fri

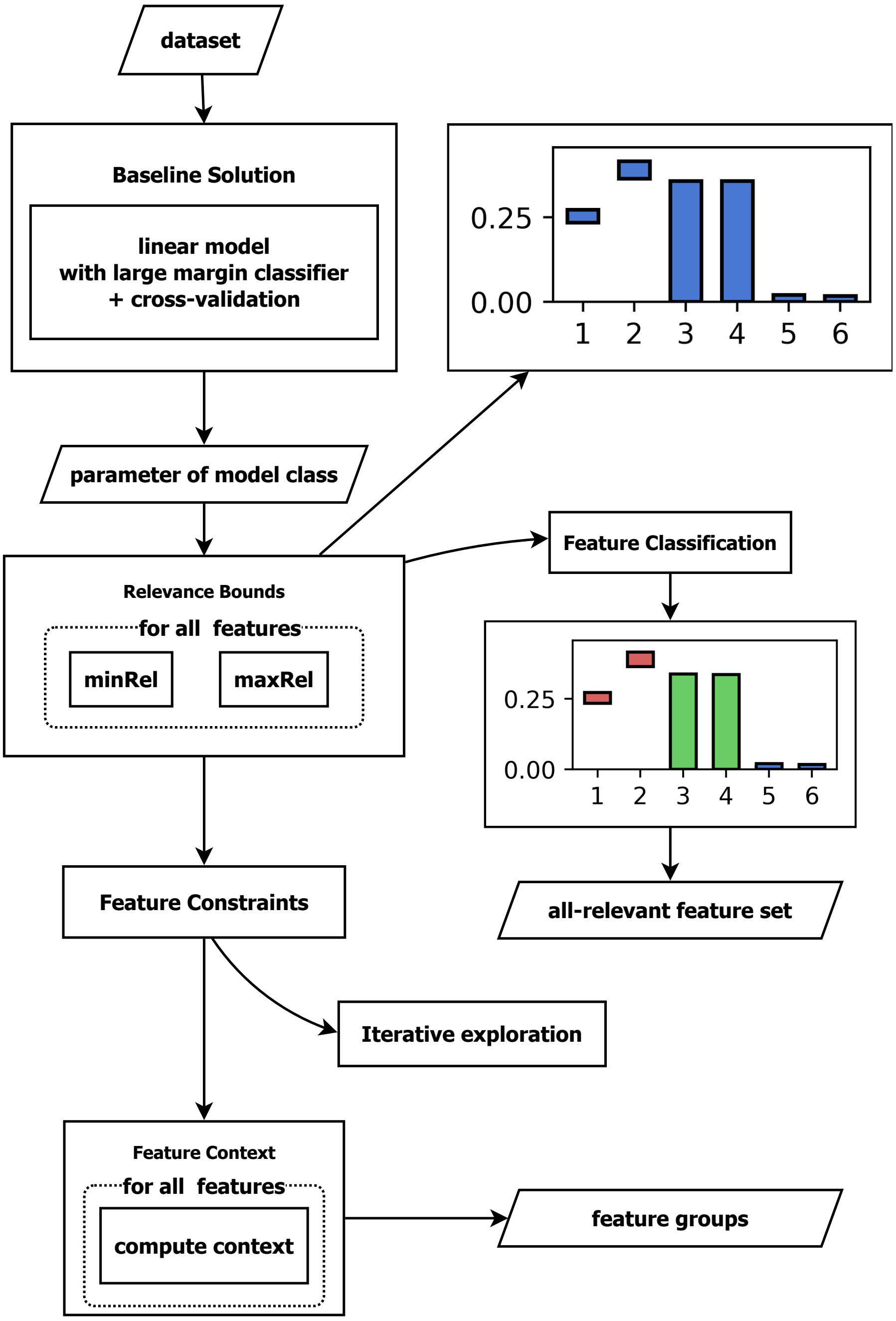

Figure 3.1: Overview of proposed pipeline implemented in the FRI tool. Rectangles represent methods and slanted parallelograms represent input and outputs. Also visible are the relevance interval visualizations.

known ground truth data in Section 3.3.4. Finally, we test the runtime of our implementation in Section 3.3.5.

3.2 METHODOLOGY

The algorithm described in Section 2.3.2 is very useful in theory but applying it in practice can be challenging. In the following sections, we are presenting improvements and extensions to the basic relevance bounds method which are realized in our *Python* implementation. The focus of these extensions is easy usability, good interpretability and efficient runtime such that non-experts can utilize it.

3.2.1 *Feature Classification*

As a reminder, we are considering the task of binary classification. For a classification problem, we observe data

$$X = \Big\{ (x_1, y_1), \dots, (x_n, y_n) \ \in \mathbb{R}^d \times \{-1, 1\} \Big\} \tag{3.1}$$

with n samples and d real-valued features which have been tied to a target or response y by an unknown function. We assume that all d features have been standardized at mean zero and standard deviation 1.

In Section 2.3.2 we already described the lower and upper relevance bounds for each feature. For the lower bound, ℓ, we defined

$$\begin{aligned} &\text{minRel}(X, \ell) : \min_{w, b, \chi} |\boldsymbol{w}_\ell| \\ &\quad \text{s.t. for all i} \\ &\qquad\qquad y_i \left(\boldsymbol{w}^\top x_i - b \right) \geq 1 - \chi_i, \quad \chi_i \geq 0 \\ &\qquad \text{and} \\ &\qquad\qquad \sum_{i=1}^{n} \chi_i \leq \rho \\ &\qquad\qquad \|\boldsymbol{w}\|_1 \leq (1 + \delta) \cdot \mu \, . \end{aligned} \tag{3.2}$$

And the upper bound

$$\begin{aligned} &\text{maxRel}(X, \ell) : \max_{w, b, \chi} |\boldsymbol{w}_\ell| \\ &\quad \text{s.t. for all i} \\ &\qquad\qquad y_i \left(\boldsymbol{w}^\top x_i - b \right) \geq 1 - \chi_i, \quad \chi_i \geq 0 \\ &\qquad \text{and} \\ &\qquad\qquad \sum_{i=1}^{n} \chi_i \leq \rho \\ &\qquad\qquad \|\boldsymbol{w}\|_1 \leq (1 + \delta) \cdot \mu \, . \end{aligned} \tag{3.3}$$

Computing these bounds for all features results in matrix

$$\text{RI}(X) := \begin{pmatrix} \text{minRel}(X, 0) & \text{maxRel}(X, 0) \\ \text{minRel}(X, 1) & \text{maxRel}(X, 1) \\ \vdots & \vdots \\ \text{minRel}(X, d) & \text{maxRel}(X, d) \end{pmatrix} \tag{3.4}$$

with all lower bounds in the first column and all upper bounds in the second. Each row contains the lower and upper relevance bound of every feature which can also be interpreted as an interval of possible feature contributions. These intervals indicate the degree up to which a feature can or must be used in the classification, and they can be visualized for interpretative purposes as seen in Figure 3.2.

In [Göp+18] the authors proposed the following feature classification rules:

Strongly relevant A feature is strongly relevant when its lower relevance bound is bigger than zero. The model class defined by its prediction accuracy is dependent on information from it.

Weakly relevant When two or more features are correlated they can replace each other functionally in the model. These features are characterized by a lower bound equal to zero and an upper bound bigger than zero.

Irrelevant By definition irrelevant features should have no measured relevance at all. Their upper (and lower) bound should therefore be zero.

While the relevance bounds should give a truthful indication of feature relevance, in practice the discrimination between relevant and irrelevant features is challenging: variations of the underlying distributions of the features imply that thresholds for feature relevance can vary for different features. The use of slack variables in the overall model and thus the relevance bounds allow variation in the contribution of features which improves finding stable solutions but also adds noise. This is exacerbated by the behaviour of linear programming solvers, which often have exhibit loss of precision. As an example specifically for relevance bounds: even if feature ℓ is independent, we often observe $\mathrm{maxRel}(\ell) > 0$ and $0 < \mathrm{minRel}(\ell) < 10^{-5}$. This shows that a static, data-independent threshold can not discriminate between noise and relevant features.

Statistical Feature Classification

In this section, we propose to use a threshold based on statistical estimation of variance which allows better discrimination between noise and signal. Instead of selecting a fixed threshold per algorithm, our proposal computes a threshold depending on the training data and parametrization. A similar resampling based approach was used to estimate a stopping threshold for a forward feature selection approach in [Fra+07].

We expect for a given model class defined by $\mathcal{F}_\delta(X)$ the same amount of slackness in the relevancies for irrelevant variables. This slackness is introduced by the parameters of the algorithm (δ, C) and the LP-solver's internal algorithm. Furthermore, we assume that the slackness is consistent overall features because the parametrization is not changing and that it follows a Gaussian distribution with small deviations from the mean as a result from random fluctuations in the data. We assume that this slackness is present for all feature relevancies, but its distribution can only be estimated by observing independent variables.

Because independent variables are by definition irrelevant for the target the only observable relevance in their case should result from slackness. Therefore, we can use independent variables to estimate the distribution of the slackness. To create independent variables we utilize randomly permuted input features from X. We define $\mathrm{perm}(X_\ell)$ as the random permutation of values in X_ℓ and

$$X \diamond \ell_k := \begin{cases} X, & \text{if } k \neq \ell \\ \mathrm{perm}(X_\ell), & \text{otherwise} \end{cases}$$

as the dataset where only ℓ was replaced by its random permutation.

Then we define random variable

$$\pi(\mathrm{maxRel}) := \mathrm{maxRel}(X \diamond \ell, \ell) \quad \ell \sim \mathcal{U}(1, d)$$

where ℓ is an i.i.d sample from the discrete uniform distribution $\mathcal{U}$ over all feature indices in $\mathcal{D}$. The random variable represents the minimal or maximal feature relevance of the permuted randomly chosen feature, thus, we assume that the slackness of minimal and maximal relevance is different and requires two independent distributions. A sample population of such a distribution is defined as

$$\widehat{\pi}(\text{maxRel})_{(\alpha)} := \Big(\pi(\text{maxRel})_i\Big)_{i\in\{1,\ldots,\alpha\}}$$

and

$$\widehat{\pi}(\text{minRel})_{(\alpha)} := \Big(\pi(\text{minRel})_i\Big)_{i\in\{1,\ldots,\alpha\}}$$

with α samples.

Now, we require a statistical framework to test, if the relevance of an actual unperturbed input feature is feasibly distributed according to one of the unknown slackness distributions. Geisser proposed predictive confidence intervals for such a purpose in [Gei93, Chapter 2]. After sampling α times, the predictive interval gives bounds for the most likely outcome of the sample $\pi_{\alpha+1}$. The probability p that the next sample $\pi_{\alpha+1}$ lies in these bounds is given by

$$Pr\Bigg(\overline{\widehat{\pi}}_{(\alpha)} - \mathcal{T}_{\alpha-1}(p)\cdot\sigma\sqrt{1+(\frac{1}{\alpha})} \le \pi_{\alpha+1} \le \overline{\widehat{\pi}}_{(\alpha)} + \mathcal{T}_{\alpha-1}(p)\cdot\sigma\sqrt{1+(\frac{1}{\alpha})}\Bigg) = p$$

where $\overline{\widehat{\pi}}_{(\alpha)}$ denotes the sample mean and σ its standard deviation and $\mathcal{T}$ represents Student's t-distribution with $\alpha-1$ degrees of freedom and denotes p its chosen percentile.

The prediction interval is defined as

$$\Pi(\text{minRel},\alpha) := \overline{\widehat{\pi}(\text{minRel})}_{(\alpha)} \pm \mathcal{T}_{\alpha-1}(p)\cdot\sigma\big(\widehat{\pi}(\text{minRel})_{(\alpha)}\big)\sqrt{1+(1/\alpha)}$$

and analogously for maxRel.

Note the ± which yields two values resulting in the interval which acts as our new feature selection threshold. Instead of testing new samples from permutation features, we instead use actual real features and check if their relevance is inside this interval. The size of Π depends on parameter p and we propose value $p = 0.999$ for a low false-positive rate and $\alpha \ge 50$ which yields sufficiently robust thresholds for a common feature set sizes in our experiments without adding too many computations to the complexity, which we analyse in Section 4.2.2.

To classify feature ℓ as irrelevant we check if its relevance bounds are inside the bounds of our prediction intervals. The following logical test uses the prediction interval to produce the three relevance classes:

Strong relevance:

$$\text{maxRel}(\ell) \notin \Pi(\text{maxRel}) \;\wedge\; \text{minRel}(\ell) \notin \Pi(\text{minRel})$$

Weak relevance:

$$\text{maxRel}(\ell) \notin \Pi(\text{maxRel}) \;\wedge\; \text{minRel}(\ell) \in \Pi(\text{minRel})$$

Irrelevance:

$$\text{maxRel}(\ell) \in \Pi(\text{maxRel}) \;\wedge\; \text{minRel}(\ell) \in \Pi(\text{minRel})$$

Here, parameters X and α where omitted for readability.

Our method provides the set $\mathcal{A} \in \{0,1,2\}^d$, which numerically encodes strongly (2), weakly (1) and irrelevant (0) features.

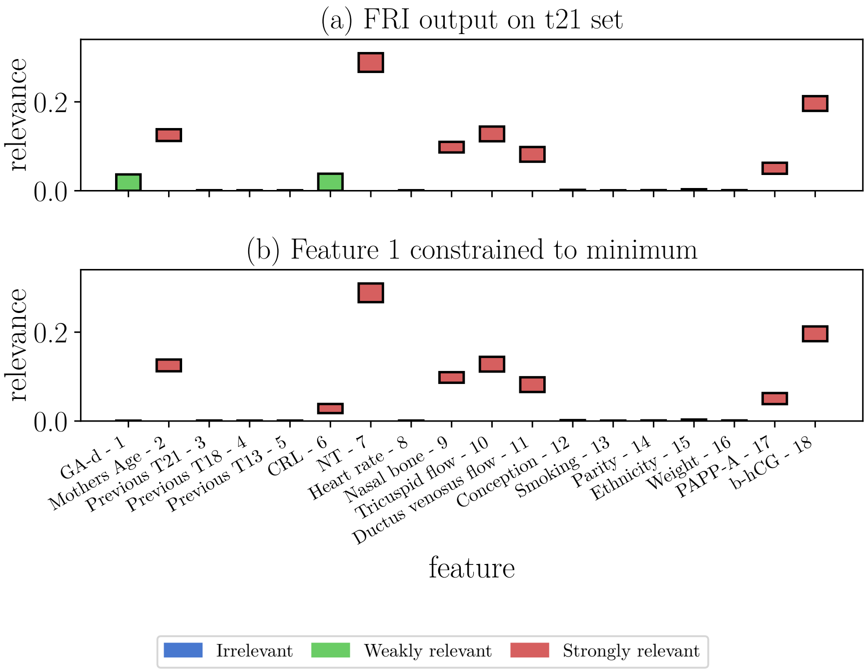

Figure 3.2: Program output using the *t21* dataset visualizing relevance bounds for all features as coloured boxes. Colours correspond to relevance classes assigned by FRI. (a) Shows program output without any constraints introduced by the user. (b) Shows output with feature 1 GA-d ("Gestation age in days") set to its minimum value.

3.2.2 *Feature Constraints*

The mathematical formalization of relevance intervals introduced in Section 2.3.2 opens up the opportunity to integrate prior knowledge about feature relevancies and to iteratively explore solutions by integrating additional constraints for specific features. By solving the problem using Linear Programs (LPs), the addition of constraints is easy. One way to leverage this is the possibility of adding relevance constraints to the optimization.

Given the set of all features $\mathcal{D}$ and a feature ℓ we define a set of additional constraint ranges K. A constrained feature ℓ is defined by

$$K_\ell := (K_{\ell,\min}, K_{\ell,\max}) \tag{3.5}$$

such that

$$K := \{K_\ell \mid \ell \in \mathcal{D}\} \tag{3.6}$$

is the set of all constrained features. Note that $K_\ell \geq 0$ because relevancies are by definition positive. Each constraint pair K_ℓ sets new bounds in the optimization for the usage in the model. In the case of $K_{\ell,\min} = K_{\ell,\max}$ we consider the model's usage of feature ℓ as fixed to a static value. Although the values in K can be chosen arbitrarily under the given restrictions, in practice one should stick within the relevance bounds in RI. Otherwise, most models would be infeasible to solve under the restrictions of similar model parameters introduced in Section 2.3.2.

To compute relevance bounds including individual feature constraints, we have to extend the set of existing constraints in the optimization from

Section 2.3.2. The minimum relevance bound with *constraints* is defined as

$$
\begin{aligned}
\text{minRelC}(X, \ell, K) : \min_{w,b,\chi} & |w_\ell| \\
\text{s.t. for all i} & \\
& y_i \left(w^\top x_i - b \right) \geq 1 - \chi_i, \quad \chi_i \geq 0 \\
\text{and} & \\
& \sum_{i=1}^{n} \chi_i \leq \rho \\
& \|w\|_1 \leq (1+\delta) \cdot \mu \\
& K_{k,\min} \geq |w_k| \geq K_{k,\max} \quad \forall k \in K \,.
\end{aligned}
\tag{3.7}
$$

New is the last constraint bounding $|w_k|$ in between the given K_k. The maxRelC is defined analogously with a maximization objective. To rewrite the new absolute term $|w_\ell|$ as a convex problem, we utilize the baseline solution $\tilde{w}$, which allows us to use the sign of the coefficient $\tilde{w}_\ell$ turning the non-convex absolute term into a simple convex one.

By changing the amount of contribution allowed for one feature, we can observe varying relevance bounds for others and infer potential dependencies between them as in Figure 3.2 (b). In our tool, we provide the means to easily define ranges or values for all features. These preset values can freely be chosen but the following calculation according to the feature relevance bound algorithm is constrained by the initial model values. That can lead to infeasible solutions. To circumvent this we provide a method to only change one or few features while the rest of features is left variable.

3.2.3 *Grouping*

While the method in Section 3.2.2 can facilitate manual model design, we also looked into making this process automatic. The overall goal is to find groups of features that have a similar function in the model, which should be visible by correlated feature relevancies. Feature grouping in general aims to find similar features to facilitate dimensionality reduction or as a part of feature selection. This can be done purely unsupervised based on interdependence measures or supervised in conjunction with a learning model. Examples of this are feature clustering approaches [Büh+13] or linear learning models with integrated grouping terms [Kam+16].

We propose to use the changes in relevance bounds observed in different contexts of constrained subproblems. We use these changes as a proxy for functional similarity, and we measure these changes systematically and interpret them as pairwise similarities. This allows inferring a connected tree representation using hierarchical clustering which can be used as a visual interpretation aid. Furthermore, we can approximate a clustering of the features where each cluster contains similar features.

Feature Context First, to compute relevance changes systematically we fix feature relevancies to their extremal values $\text{RI}_\ell^{min/max}$ and observe how these choices affect other features. The following situations can occur:

1. **Feature information dependency**: The information within feature ℓ (at least partially) depends on another feature k if the latter is required to use the information of the former. This can be observed by setting K_k to its minimum value and observing whether this causes a decrease of the maximum relevance of ℓ. Alternatively, feature dependency can

also manifest itself in the fact that setting K_k to its maximum value increases the minimum value of a feature k, i. e. k can only optimally be used if also ℓ is present

2. **Feature information redundancy**: The information of feature k can be (at least partially) substituted by feature ℓ. This can be observed in two cases: when setting K_k to a minimum value the minimum relevance of feature ℓ is increased. Conversely, when setting K_k to a maximum value, the maximum relevance of feature ℓ is decreased. This can be observed in highly correlated feature pairs.

Functional similar features behave similarly in similar contexts. For every feature ℓ, we can measure the impact of setting K_k to the minimal and maximal possible relevance bounds for all other features. In other words, we express each feature's functional behaviour by observing it in two contexts:

- When set to the *minimal* relevance bound, i. e. $K_\ell^{\min} := \left(\mathrm{RI}_\ell^{\min}, \mathrm{RI}_\ell^{\min}\right)$
- When set to the *maximal* relevance bound, i. e. $K_\ell^{\max} := (\mathrm{RI}_\ell^{\max}, \mathrm{RI}_\ell^{\max})$

This yields a vectorial description for all features which we can use to group features through a clustering algorithm. Note that K only contains one feature ℓ for this use case such that

$$K = \{K_\ell\}$$

and $|K| = 1$.

We represent each feature in its functional context to other features as defined by the feature relevance bounds. In the following, we are always applying the algorithm on X such that $\mathrm{minRel}(X, \ell)$ is shortened to $\mathrm{minRel}(\ell)$. Now, similar to the array

$$\mathrm{RI} := (\mathrm{minRel}(i), \mathrm{maxRel}(i))_{i=0}^{d}$$

we define relevance intervals with a *single* feature (k) constrained as

$$\mathrm{RIC}(k, \min) := \begin{pmatrix} \mathrm{minRelC}(0, K_\ell^{\min}) & \mathrm{maxRelC}(0, K_\ell^{\min}) \\ \mathrm{minRelC}(1, K_\ell^{\min}) & \mathrm{maxRelC}(1, K_\ell^{\min}) \\ \vdots & \vdots \\ \mathrm{minRelC}(d, K_\ell^{\min}) & \mathrm{maxRelC}(d, K_\ell^{\min}) \end{pmatrix} \tag{3.8}$$

where $K_\ell^{\min}$ is $\mathrm{minRel}(k)$. We apply the algorithm from Section 2.3.2 again with one feature constrained to its minimum relevance bounds. Analogous is the definition for the maximum:

$$\mathrm{RIC}(k, \max) := \left(\mathrm{minRelC}(i, K_\ell^{\max}) \;\; \mathrm{maxRelC}(i, K_\ell^{\max})\right)_{i=0}^{d} \tag{3.9}$$

Because we are not interested in the absolute values but in the relative changes, we take the difference to the initial unconstrained feature relevance RI. Additionally, we combine both arrays (3.8) and (3.9) as

$$\mathrm{context}(k) := (\mathrm{RI} - \mathrm{RIC}(k, \min), \mathrm{RI} - \mathrm{RIC}(k, \max)) \in \mathbb{R}^{2d} \tag{3.10}$$

where "$-$" is used as the element-wise difference such that

$$\mathrm{RI} - \mathrm{RIC}(k, min) = \begin{pmatrix} \mathrm{minRel}(0) & - & \mathrm{minRelC}(0, K_\ell^{\min}) \\ \mathrm{maxRel}(0) & - & \mathrm{maxRelC}(0, K_\ell^{\min}) \\ & \vdots & \\ \mathrm{minRel}(d) & - & \mathrm{minRelC}(d, K_\ell^{\min}) \\ \mathrm{maxRel}(d) & - & \mathrm{maxRelC}(d, K_\ell^{\min}) \end{pmatrix}. \tag{3.11}$$

Hence, we measure the change in size and position of all other relevance intervals when k is set to a fixed value. In this case, the fixed values are both the upper and lower relevance bound of k. This definition captures the functional role of feature k for the classification prescription since it accumulates the information in how far feature k is redundant / dependent to other features. Note that $\text{context}(k)_k$ is set to the neutral element 0 because $\text{RIC}(k, \min)_k$ and $\text{RIC}(k, \max)_k$ are static by definition.

Similarity measure For a clustering method, one has to decide on a similarity measure which follows the characteristics of the given data. Give two features ℓ and k, we propose to take the Euclidean distance difference in between two functional context vectors as dissimilarity measure, whereby we modify the contexts by omitting values ℓ and k of the vectors:

$$\delta(\ell, k) := \sqrt{\sum_{\substack{i=0\\ i\neq\ell\\ i\neq k}}^{d} \left(\text{context}(\ell)_i - \text{context}(k)_i\right)^2}$$

In this distance function, we exclude the contribution of ℓ and k and only observe the relation to all other features. This removes a direct pairwise influence like strong direct correlation would produce and focuses on the more general functional dependencies. It also enforces symmetry of the dissimilarity measure which would be violated when one feature would be dependent on the other. In general, however, this choice is no longer necessarily a metric since it might violate the triangle inequality.

Visual Clustering With this measure we can now group features utilizing any suitable clustering technology which relies on pairwise dissimilarities. Here, we opt for a parameterless technology which is offered by classical hierarchical clustering methods in the form of agglomerative clustering. In agglomerative clustering, the features get grouped bottom-up starting from the most similar pair.

To link groups together one has to decide on a linkage function, which defines the similarity for sets. In our case we are using the single linkage which is defined for two sets of features $\mathcal{A}$ and $\mathcal{B}$ as

$$\text{link}(\mathcal{A}, \mathcal{B}) := \min_{a\in\mathcal{A},\ b\in\mathcal{B}} \delta(a, b).$$

This linkage takes the minimal existing distance between all possible pairs of elements. The choice for this linkage is promoting the imputation of function by using the transitive similarity of single features. For example in the biosciences with regulatory elements as features, if one feature of a group is very similar to another feature in another group, they can often be considered part of a common pathway.

The clustering now iteratively aggregates features into growing groups as long as more than one group exists. If all features are in one group the algorithm stops and this final group can be interpreted as a tree. Starting from the root of the tree, i. e. the final group, the edges represent the linkage distance. The distances are balanced such that all inner sibling nodes have the same distance to their direct parent.

We propose to use this tree representation as a visual aid together with a relevance bound visualization to highlight similar features. In Figure 3.3 this visual combination can be seen where the root is at the top.

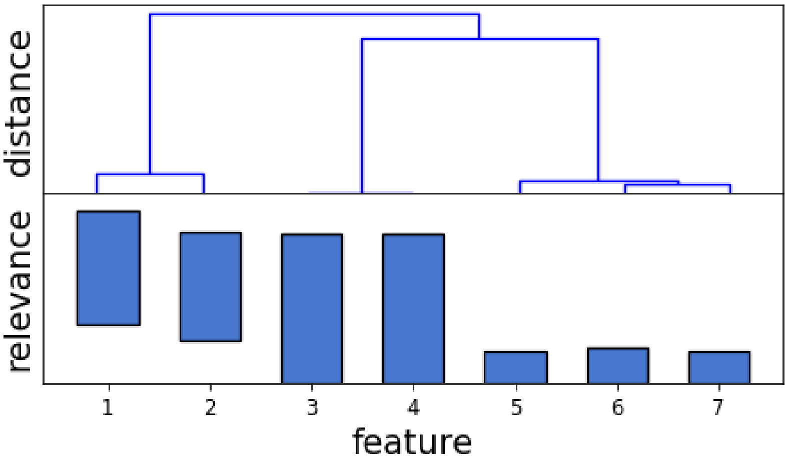

Figure 3.3: Combined figure provided by the methods plotting function. Bottom subplot shows the calculated relevance bounds represented as bars. Top subplot shows the corresponding tree clustering. The vertical length of the lines represents the linkage distance used in the agglomerative clustering.

Flat Clustering To turn a tree structure into distinct groups, i. e. disconnected components, one can cut the edges at an equal length coming from the root, here named link_{cut}. While the true parameter link_{cut} cannot be known in practice, one can decide based on the given application on how to set it. For this, we recommend using the visualization as shown in Figure 3.3 and deciding on a link_{cut} interactively by integrating possibly existing knowledge as guidance. Furthermore, if one expects a certain number of groups k, the parameter can also be set accordingly to cut at the right length to produce that amount of clusters.

Even if the number of clusters k is not known beforehand, several approximations exist to estimate it even though the quality of the resulting clustering is questionable. In our implementation, we include a cutting heuristic which takes the maximum linkage distance in the tree and cuts all edges at that distance. We use that heuristic to perform a quantitative evaluation in Section 3.3.4.

3.3 EVALUATION

To evaluate the different aspects shown in Section 3.2 we perform several evaluations in the following sections. In Section 3.3.1 we describe several synthetic and real datasets which were used in those evaluations. Subsection 3.3.2 focuses on the performance when regarding classical feature selection, Section 3.3.3 demonstrates the interactive use case involving our feature constraints from Section 3.2.2 and Section 3.3.4 highlights the automatic variant. Finally, in Section 3.3.5 we analyse the runtime.

3.3.1 *Data*

For our evaluations, we utilize two types of data: (1) simulated data with known properties and ground truth and (2) real data mainly coming from the biomedical sciences.

Table 3.1: Characteristics of simulated datasets. Each set consists of 30 features with 500 samples.

	Number of features		
data	Strongly relevant	Weakly relevant	Irrelevant
Sim1	4	4	22
Sim2	12	8	10
Sim3	4	0	26
Sim4	18	0	12
Sim5	0	20	10

Simulation Data

All our simulation sets are sampled from a binary classification problem. To generate a multidimensional classification problem, we use a randomly generated prototype vector which defines a hyperplane. The defining features of this plane are strongly relevant. Now points are sampled in this feature space and the class is determined by the side of the hyperplane the points lie on. Weakly relevant features are constructed by replacing a feature of the original feature space with its linear combination. The elements of this combination are highly correlated and produce a set of redundant features. By removing the original feature and replacing it with those elements we achieve weak relevance by definition. Irrelevant features are sampled from a standard normal distribution.

Sim1 and *Sim3* have a sparse relevant feature space while *Sim2* and *Sim4* are dense. Additionally, in *Sim1* and *Sim2* weakly relevant features are present, while they are missing completely in *Sim3* and *Sim4*. *Sim5* had all strongly relevant features removed.

Biomedical Data

The biomedical datasets are gathered from multiple studies and differ in size and type:

t21 This set stems from a series of prenatal examinations of pregnant women with the goal of early diagnosis of chromosomal abnormalities, such as trisomy 21. The study covers sociodemographic, ultrasonographic and serum parameters which result in 18 usable features. The original set contains over 50.000 samples but only a low percentage ($\approx$ 0.8%) of abnormal samples. It was collected by the Fetal Medicine Centre at King's College Hospital and University College London Hospital in London [Nic+05].

flip This set is used for the prediction of fibrosis. The diagnosis of fibrosis is represented as a score which is based on sociodemographic and serum parameters. The set consists of samples of 118 patients and 19 features and was provided by the Department of Gastroenterology, Hepatology and Infectiology of the University Magdeburg [Sow+13].

spectf The *spectf* dataset consists of 44 features describing cardiac Single Proton Emission Computed Tomography (SPECT) images. Each of the 267 patients' images were diagnosed as either normal or abnormal.

wbc The *wbc* dataset contains 32 markers for cell image-based breast cancer diagnostics from 569 patients.

colposcopy A set with 69 extracted structural features from videos acquired during colposcopies [FCF17]. Classification of practitioners clinical judgment using the *Schiller* modality.

Sets *spectf*, *wbc* and *colposcopy* were acquired through the UCI Machine Learning Repository [DK17]. The biomedical datasets are preprocessed before analysis. Samples with over 90% missing values are removed. Sets are split into stratified training and testing subsets. If samples still contain missing feature values, we replace them with the feature's training set mean in both subsets. Similarly, the z-score transformation is based on the training set and applied to both. In case the original set is imbalanced, we use the Synthetic Minority Over-sampling Technique (SMOTE) [Cha+02] in combination with the Nearest Neighbour cleaning rule [Wil72; Lau01] as described in [BPM04]. In one case (*t21*) with an extremely large majority class, we only perform downsampling.

3.3.2 *Selection Accuracy*

The most important aspect of our proposed method is the performance of the feature selection. Performance in this context is the precision and recall of selected relevant features in the set of all input features $\mathcal{D}$. We assess the performance with known ground truth using quantitative measures, and we analyse it qualitatively on real sets.

Benchmark methods To evaluate the method in context we run this analysis with several methods:

- Boruta [KR11]
- Ensemble Feature Selection (EFS) [NGH17]
- ElasticNet (EN) using an equal contribution of L_1 and L_2 regularization [ZH05]. Lasso performed very similar to EN such that we only included the latter.
- Stability Selection (SS) [MB10; SS13]
- Feature Relevance Intervals method (FRI) is the implementation of the methods proposed in Section 3.2

For all methods, the proposed default parameters are used. Hyperparameters are selected according to a cross-validation scheme. For EN, we choose the feature set depending on the coefficients c_i of the model where $c_i > 10^{-5}$ counts as selected.

Supervised

In this section, we focus on the aspect of the all-relevant feature selection problem (ARFS) and compare the match of the selected feature set and the known ground truth of all relevant features. In detail, we measure several key quantities:

True Positives (TPs): The number of correctly identified relevant features.

False Positives (FPs): The number of irrelevant features identified as relevant.

True Negatives (TNs): The number of correctly identified irrelevant features.

False Negatives (FNs): The number of relevant features *not* identified as irrelevant.

Table 3.2: Average training set accuracy on simulation data. In the case of Boruta the internal RF score was reported. For EFS accuracy is not defined.

	accuracy				
data	Boruta	EFS	EN	FRI	Stability Selection (SS)
Sim1	0.99	-	1.00	0.92	1.00
Sim2	0.97	-	1.00	0.96	1.00
Sim3	0.99	-	1.00	0.96	1.00
Sim4	0.97	-	1.00	0.93	1.00
Sim5	1.00	-	1.00	0.91	1.00

These absolute quantities can be combined in relative measures: precision and recall. The recall is defined as the ratio between TP and (TP+FN), i. e.

$$recall := \frac{\mathtt{TP}}{\mathtt{TP}+\mathtt{FN}}.$$

It denotes how many of the relevant features were selected which is crucial when looking for the all relevant feature set.

Precision is defined by

$$precision := \frac{\mathtt{TP}}{\mathtt{TP}+\mathtt{FP}}$$

and describes what rate of false positives are part of the feature set.

Because of the typical trade-off between precision and recall one can also use the F_1measure which is the harmonic mean of the two former:

$$F_1 := 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

For the supervised analysis, we utilize the simulation datasets from Section 3.3.1. Due to known ground truth, we can explicitly evaluate the validity of selected features. All simulation sets consist of 30 features and 500 samples. They differ in the density of the relevant feature space which is defined by the amount of strongly, weakly and irrelevant variables which are listed in Table 3.1. According to these parameters, 50 sets were generated per configuration and the following evaluation refers to the averaged scores which can be seen in Table 3.3.

Before we evaluate the selection measures, we confirm that all models had a proper fit. Listed in Table 3.2 are the training accuracies. Instead of comparing feature quantities, these are the training accuracy on the training samples, i. e. the typical model prediction accuracy. One can see in the table that most classification models had accuracy values over 90% which signifies a sufficient fit of the data. EFS is an ensemble of a variety of statistical models and methods and has no score defined.

To evaluate the feature selection performance we mainly observe the F_1 score in Table 3.3. Here our proposed method FRI takes the lead overall with a nearly perfect score in all simulation sets. Depending on the presence of weakly relevant features, the other methods show loss of recall which leads to a reduced F_1 score. This is especially evident for *Sim2* and *Sim4* in the case of SS and EFS. The worst recall is achieved by SS for *Sim5* where it did not select any of the weakly relevant variables. SS still achieves slightly better scores in *Sim3* where no weakly relevant features are present.

Table 3.3: Feature selection score on simulated datasets. Values are showing the performance of each method to classify the relevance of input features.

score	data	Boruta	EFS	EN	FRI	SS
F_1	Sim1	**0.98**	0.96	0.62	**0.98**	0.77
	Sim2	0.82	0.76	0.84	**0.98**	0.75
	Sim3	0.91	0.71	0.44	0.99	**1.00**
	Sim4	0.82	0.84	0.82	**0.99**	0.91
	Sim5	0.98	0.94	0.80	**0.99**	0.27
precision	Sim1	0.99	0.93	0.46	0.98	1.00
	Sim2	1.00	1.00	0.74	1.00	1.00
	Sim3	0.87	0.57	0.28	0.98	1.00
	Sim4	1.00	1.00	0.69	0.99	1.00
	Sim5	1.00	1.00	0.67	1.00	1.00
recall	Sim1	1.00	1.00	1.00	0.99	0.62
	Sim2	0.72	0.62	0.98	0.97	0.60
	Sim3	0.98	0.98	1.00	1.00	1.00
	Sim4	0.70	0.73	1.00	0.98	0.83
	Sim5	0.95	0.90	1.00	0.99	0.16

Table 3.4: Average training set accuracy on real data. In the case of Boruta the internal RF score was reported. For EFS accuracy is not defined.

	accuracy				
data	Boruta	EFS	EN	FRI	SS
colp.	1.00	-	0.99	0.97	0.99
flip	1.00	-	0.90	0.82	0.90
spectf	1.00	-	0.99	0.92	0.98
t21	1.00	-	0.98	0.93	0.98
wbc	1.00	-	1.00	0.98	1.00

Unsupervised

To assess the quality of the feature selection on real datasets, we have to rely on the problem performance itself since no ground truth feature relevance is available.

First, we observe the model accuracy on the training set samples. Most models show very high accuracies. One exception is the case FRI for the flip dataset which is only at 82% which could be accounted to the model's simplicity in comparison with the alternatives.

Now we again focus on the resulting feature sets. We expect an FS method to pick features which contain information and a loss of features with crucial information is signified in a decrease of performance. Notable exceptions are redundant features, which can decrease performance when their presence increases model complexity and lead to bad generalization. Instead of looking at each model's internal accuracy score, we evaluate the selected feature sets by their discriminative power, whereby the latter is uniformly evaluated by a logistic regression model, which is a very popular model for predictive purposes in medical applications [BWG01] and very fast to evaluate. The model is trained using only the predicted feature set.

Table 3.5: ROC-AUC values of a logistic regression model using features selected by listed models. The values are averaged over 50 bootstraps.

	ROC-AUC				
data	Boruta	EFS	EN	FRI	SS
colposcopy	0.568	0.586	0.640	**0.661**	0.625
flip	0.804	0.652	**0.815**	0.743	0.705
spectf	0.871	0.874	0.867	0.880	**0.888**
t21	0.971	0.977	0.971	0.975	**0.978**
wbc	0.997	0.998	0.998	0.998	0.999

Finally, for this selected model the receiver operation characteristics (ROC) on the holdout validation set is recorded. ROC denotes the trade-off between recall and the false positive rate over all possible threshold parameter choices. For the comparison, we look at its area under the curve (AUC) such that an area of 1 is the maximum possible area and also the best possible score, as it signifies perfect recall with no false positives. We perform the test on 50 bootstrap replicates with sample size $0.7 \cdot n$ where samples are chosen from the original set with replacement. The averaged results are given in Table 3.5. Here the AUC on the five datasets shows no clear overall superior method which is in line with the common expectation that the minimal optimal set is the objective of most methods and sufficient for prediction. On the *spectf*, *t21* and *wbc* datasets most methods produce very similarly performing feature sets. In the case of *colposcopy*, the feature set selected by FRI achieves the best performance. SS produces sightly better sets in two cases. The EN performs solidly in all cases based on its very conservative selection method where informative features are not removed often.

Evaluating our goal, the selection of redundant features, is not discernible when analysing prediction performance. In the search for a complete feature set, we need to take the selected set size into account. Table 3.6 lists the average feature set sizes over all experiments. Because FRI provides additional information by not only conserving all weakly relevant features but also by denoting the feature class (strong/weak relevance) we can explicitly list those as well. As mentioned in the last paragraph, we can easily see that EN is very conservative in its selection. It produces by far the biggest feature sets with many false positives in the case of the *Sim* sets but also most likely in the real datasets. Similarly, Boruta achieves better precision in the simulated data but shows seemingly inflated set sizes. SS on the other hand exhibits very good precision overall. Interestingly, the size of the sets chosen by SS is very similar to FRI_s, the set of strongly relevant features chosen by FRI. This indicates that FRI can find strongly relevant features with high precision, but also highlights the additional information provided by the weakly relevant features contained in FRI_w.

3.3.3 *Interactive Use*

By having additional information available in the potential set of weakly relevant features FRI_w we can gain insights into the structure of the data. We can improve the design of models and diagnostic tests in biomedical applications. Our framework given in Section 3.2.2 allows introducing constraints into the model. This makes it possible to limit the contribution of certain features to specific intervals or a fixed value. These limits can come from

Table 3.6: Average selected feature set size. Additionally, for FRI the size of the strongly($_s$) and weakly ($_w$) relevant feature set is available.

data	feature set size					composition	
	Boruta	EFS	EN	SS	FRI	FRI$_s$	FRI$_w$
Sim1	8.1	8.7	17.8	5.0	8.1	5.1	3.0
Sim2	14.3	12.3	26.6	12.1	19.4	12.4	7.0
Sim3	4.6	7.2	14.8	4.0	4.1	4.0	0.1
Sim4	12.6	13.2	26.2	15.0	17.9	17.9	0.0
Sim5	19.1	17.9	29.7	3.2	19.9	0.0	19.9
colp.	35.1	25.4	46.5	41.5	20.3	5.9	14.4
flip	18.8	8.1	16.9	9.1	8.9	8.8	0.1
spectf	44.0	20.3	43.1	5.9	19.9	5.9	14.0
t21	15.5	7.9	14.2	9.6	9.6	6.6	3.0
wbc	29.9	12.5	26.9	4.7	15.6	4.0	11.6

prior knowledge of the practitioner and represent design goals or existing hypotheses. Depending on the chosen values the model and the resulting relevance bounds change and can be visualized again which lends itself to an iterative and interactive process. In the following, we are going to evaluate that use case on simulated data and the *t21* data set.

The simulated set was generated according to Section 3.3.1. It consists of 8 features, 4 of which are strongly relevant, 3 of which are weakly relevant and one noise feature. Figure 3.4 (a) shows the output of FRI without any constraints. The four strongly relevant features (1-4) are visible as four small rectangles with lower relevance bounds (the bottom part of the rectangle) bigger than zero. The model parameters allow some variation in their contribution to the model. Three weakly relevant features (5-7) are visible as three taller rectangles with equal height because they are perfectly correlated in the normalized space. They can replace each other in the model. This is apparent when we set one of them (e. g. feature 5) to the minimum and maximum relevance bound, i. e. we calculate

$$\text{minRelC}(X, \ell, K)$$

and

$$\text{maxRelC}(X, \ell, K)$$

for all $\ell \neq 5$ and $K = \{K_5\}$. K_5 is then either RI_5^{min} or RI_5^{max}, i. e. fixed to a static value.

In Figure 3.4 (b) feature 5 is set to the minimum bound and the relevance bounds of other features are identical. That is because the other two features are still a correlated pair which allows the same degree of variability in contribution. When feature 5 is set to its maximum relevance bound in (c) we see that feature 6 and 7 no longer have a contribution. Additionally, all other relevance bounds are reduced to single values because the model in this state does not allow any more variability.

It is now interesting to apply this procedure on real data with functional associations between features. In Figure 3.2 (a) the normal FRI output of the *t21* set is presented. As a reminder, this set consists of samples acquired in prenatal examinations of mothers and their unborn children. Features in the study included socioeconomic factors as well as ultrasound imaging metrics. Notably in the output of FRI are two weakly relevant features 1 and 6. Feature 1 represents the *gestational age of the fetus in days ('GA-d')* and feature 6 the *crown-rump length ('CRL')* of the fetus, which is the length as indicated on an ultrasound machine. By intuition, we expect an association between the two measures. If we set one of the two features to its minimum relevance bound (Figure 3.2 (b)), we see that feature 6 becomes strongly relevant in the model. This highlights the association between the two which is very useful in cases where it is not clear a priori. Furthermore, we can use this as a design tool to easily select 'better' features. If we find functional alternatives, we can exclude more expensive features in future experiments or tests.

3.3.4 *Feature Groups*

While an interactive workflow is desirable in many situations, in others the automatic variant is suited better. In this section, we look at the output from the automatic grouping from Section 3.2.3. First, we do a quantitative and supervised evaluation using generated data. After that, we use some data sets from Section 3.3.1 to do a qualitative analysis using already known associations from literature as a reference.

Supervised

To evaluate the quality of the grouping we performed a supervised clustering experiment. Here we exploit the fact, that we have ground-truth knowledge about the groups in simulated data. As such this evaluation is suitable to show if the proposed grouping is sensitive to the functions of features.

We used the commonly used clustering metric V-measure [RH07] to check for the clustering quality. It is the harmonic mean of *homogeneity* and *completeness*. *Homogeneity* measures if one cluster contains only points from the same class. *Completeness* is symmetrical to *homogeneity* and measures if *all* points of the class are in the same cluster. With the harmonic mean of both values, we get an aggregated measure for the overall truthfulness of the clustering. If a method is truthful, it should put all the features of a given group into the same clusters which would result in a value of 1. A nonsense clustering would result in a value of 0.

The V-measure is defined with a given ground truth labelling of clusters. Because clear labelling is seldom found in nature, we use the data generation method from Section 3.3.1. To imitate groups of functional similar features we create identical feature pairs. One pair is considered a related group or cluster.

We generate sets with $n = 100$ and given the following characteristics:

- Set 1: 20 overall features, 5 pairs of identical relevant features, 10 random noise features
- Set 2: 18 overall features, 3 *unique* relevant features, 5 pairs, 5 noise features
- Set 3: 10 overall features, only 5 identical pairs, *no* noise features

Unique features are considered to be in a cluster with size 1. Irrelevant noise features could also be considered unique and handled as such but in this evaluation, we set them to be in a single outlier group. Directly related features are considered to be in one cluster.

We compare several clustering approaches:

- HDBSCAN [MHA17; MH17]: this density-based clustering has several advantages for a comparison. It does not need a parameter k beforehand, instead, it expects a minimum cluster size, in our case set to 2.

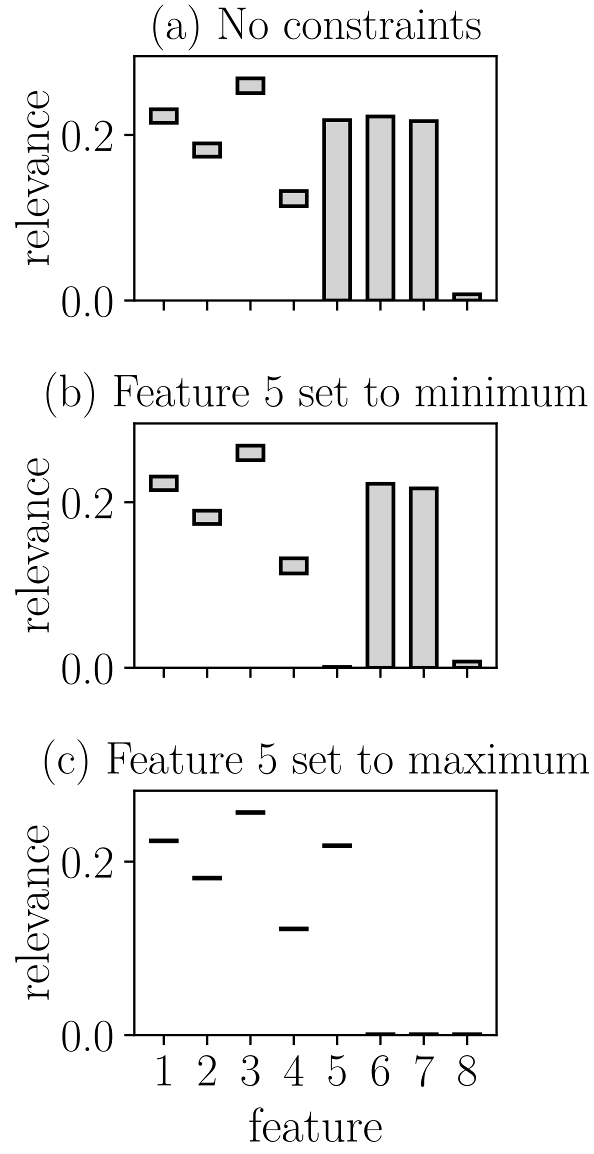

Figure 3.4: Three subplots showing feature relevance bounds in different constraint situations according to Section 3.2.2. Classification data were simulated and consisted of 4 strongly relevant features (1–4), 3 weakly relevant (5–7) and one noise feature (8). Subplot (a) had no feature constraints. Subplot (b) shows the output when feature 5 is constrained to its minimum relevance value and (c) to its maximum value.

Table 3.7: Comparison of flat clustering results from FRI, correlation-based distances and HDBSCAN approach. V-measure was computed over 5 independent runs with provided ground truth. Runtime is given in milliseconds.

		V-measure	runtime (ms)
model	**data**		
corr-clust	**Set 1**	0.72	1.11
	Set 2	0.89	0.93
	Set 3	1.00	0.90
FRI	**Set 1**	1.00	38464
	Set 2	0.90	30186
	Set 3	1.00	14520
HDBSCAN	**Set 1**	0.79	1.17
	Set 2	0.81	1.02
	Set 3	1.00	0.98

- FRI: the method implemented as described in Section 3.2.3 using variation in relevance bounds as a similarity proxy and with the cutting heuristic
- corr-clust: agglomerative clustering and cutting as described in Section 3.2.3 but using Pearson correlation instead of relevance variations

The HDBSCAN method was run on data X; corr-clust used the pairwise correlation of all features.

The experiment was performed 10 times and the average V-measure values are given in Table 3.7. Additionally, the runtime in milliseconds is given.

When considering the V-measure in the first set with many noise features FRI scores perfect and much better than the alternatives. In Set 2 the results are close but FRI has the best score with 0.9. Only in the third set, all methods achieve perfect scores. It is noticeable, that both correlation clustering and HDBSCAN can not handle noise features as well as FRI. Still, the results of correlation clustering are not far off from FRI. When also considering the runtime, which is many magnitudes higher for FRI, the alternatives are much faster. It is arguable if the additional value of FRI, in this case, is worth the extreme differences in computational requirements.

Unsupervised

To replicate a typical use case for our method we compare the combined results in our given visualization to references in literature.

When looking at Figure 3.5, we can observe some closely related feature pairs. One of those pairs is "HDL" and "Bloodsugar" which are two quantitative markers for the corresponding levels in the blood. A functional relation between the two is already known [Leh+13]. The pair exhibits a slight correlation between them (-0.11). Another pairing with very high correlation (0.99) exists between "HbA1C" and "K".

The results in Figure 3.2 on page 22 for the *t21* set show the case where multiple features, which are most likely irrelevant, achieve a high functional similarity. This grouping of likely noise features could be used in advantage by truncating the tree in the future. Other than this, the results also show

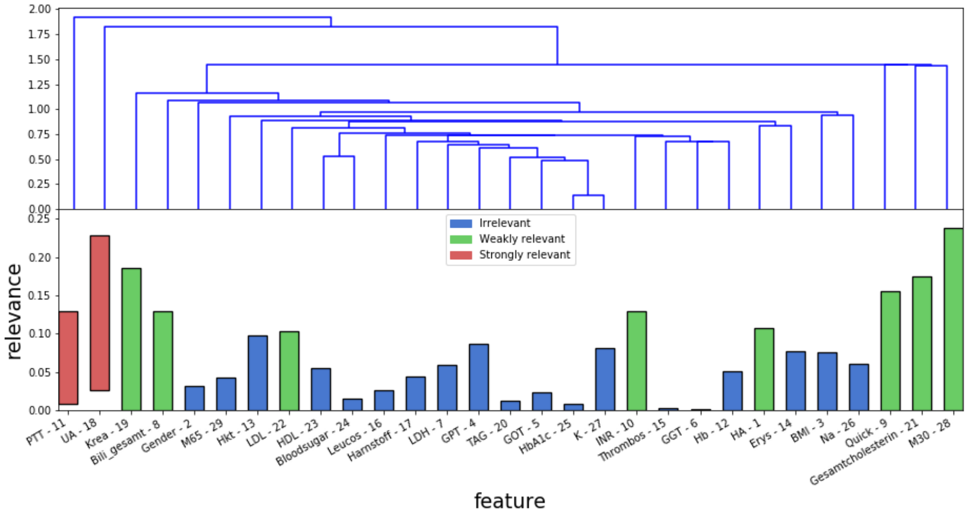

Figure 3.5: Visual output of FRI on the *fibrosis* dataset. Shown on top is the tree representation of the relevance variance as described in Section 3.2.3 and at the bottom the bars representing relevance bounds with colours indicating inferred relevance class as shown in Section 3.2.1.

multiple feature pairs. One pair with a small distance is “GA-d” which stands for the gestation age in days and “CRL” which denotes the rump length of the fetus and which are highly correlated (0.99).

3.3.5 *Runtime*

The computational runtime of a method is not only an indicator for its feasibility on bigger datasets but also especially important in the interactive use case where an analyst is actively involved in model refinement. Because relevance bounds can be solved independently in parallel, we provide the means to speed up computation by utilizing all available CPU cores on the machine. Additionally, we also tested running our program in conjunction with the distributed computation framework *Dask* which allows scaling up to any amount of separate computing nodes in a high-performance cluster such as *Grid Engine* or even in cloud backends.

In this evaluation, we focus on the runtime when considering the feature selection only. Earlier we showed the runtime when considering the feature grouping in Section 3.3.4. In Figure 3.6 we display aggregated mean runtime of the methods used in our evaluations on a single CPU thread. Because only FRI and one other method provided a parallel implementation (SS) the computations were limited to one thread, so an advantage of the parallel processing is not taken into account. EN performed best followed by SS. Both show steady runtimes over all types of data. *Boruta's* runtime is very dependent on the density of the feature space and shows some variance in the case of *t21*. The runtime of FRI is similar to Boruta in most cases but takes a hit in smaller datasets because of the constant factor of sampling permutated features for feature classification. EFS shows the slowest performance in most cases, which stems from its use of multiple complex underlying models at the same time.

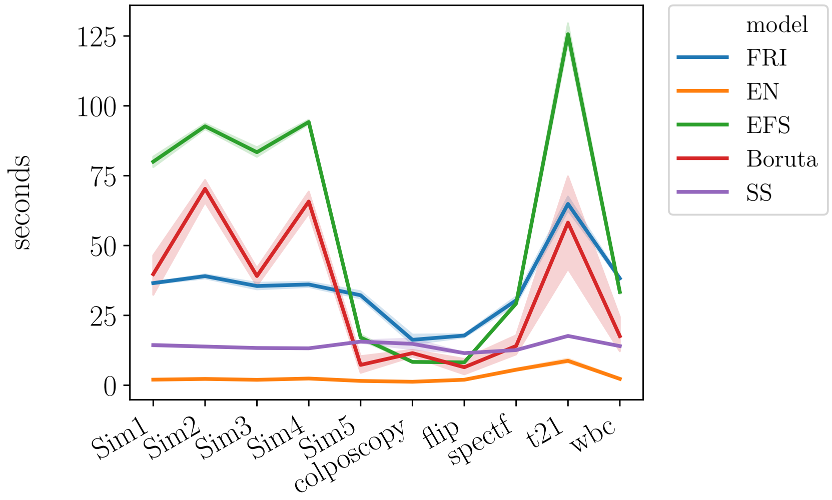

Figure 3.6: Average runtime over all bootstraps with confidence intervals.

3.4 CONCLUSION

In this chapter, we have presented the software library FRI to produce all relevant feature sets for general feature selection as well as perform interactive data exploration. We described how we implemented the algorithm from [Göp+18] and extended the method to allow a practitioner to include new constraints and experiment. We also proposed a threshold estimation method to reduce false positives which are common in all-relevant selection tasks.

In comparison with other methods, we showed that FRI can detect all relevant features in synthetic datasets while minimizing noise through its threshold estimation. On real datasets, we showcased good selection performance and additional information provided by the weakly relevant feature set. Our underlying method ensures to conserve all relevant variables while still maintaining interpretability. This is facilitated by the three relevance classes our method produces as well as the relevance bar representation which should enable better understanding for biological and medical experts in the future. In addition to facilitating understanding, we also provide a way to incorporate prior knowledge to manipulate the model and highlight related features using a tree representation which should help in the design of new experiments and biomarkers for prediction models. Furthermore, we also experimented with automatically grouping related features into clusters and evaluated this on theoretical data.

4 ORDINAL REGRESSION AND THE RELEVANCE OF PRIVILEGED INFORMATION

Parts of this chapter are based on:

- Lukas Pfannschmidt, Jonathan Jakob, Michael Biehl, Peter Tino, and Barbara Hammer. "Feature Relevance Bounds for Ordinal Regression". In: *ESANN 2019*. ESANN 2019. Bruges: i6doc, Feb. 20, 2019, ES2019–162. ISBN: 978-2-87587-065-0. URL: `https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2019-162.pdf`
- Lukas Pfannschmidt, Jonathan Jakob, Fabian Hinder, Michael Biehl, Peter Tino, and Barbara Hammer. "Feature Relevance Determination for Ordinal Regression in the Context of Feature Redundancies and Privileged Information". In: *Neurocomputing* (Apr. 9, 2020). ISSN: 0925-2312. DOI: `10.1016/j.neucom.2019.12.133`. arXiv: `1912.04832`

In the following chapter, we will introduce feature relevance learning in the context of redundant features for ordinal regression (Section 4.1) and privileged information (Section 4.2) based on the theoretical feature relevance bounds [Göp+18] described in Section 2.1 including the improvements demonstrated in Chapter 3.

4.1 LARGE MARGIN ORDINAL REGRESSION

4.1.1 *Background*

Context	
Problem	*ordinal regression*
Model	*linear*
Type	*classical*

While many applied problems are binary classification problems, as discussed in Chapter 3, many applications require an extension. Ordinal regression refers to the task to assign data to a finite number of classes or bins, which are ordered qualitatively along a preference scale. Ordinal data often occur in sociodemographic, financial or medical contexts where it is difficult to give absolute quantitative measurements but easily possible to compare samples and assign those to different bins, which are qualitatively ordered, such as the severity of a disease or the risk of a financial transaction. Another popular example of ranking on ordinal scales takes place in customer feedback or product ranking by humans [HK16]. Here, the quality is often represented by a five-star rating scale, where five stars correspond to the best rating and one star to the worst. Indeed, many human ratings are represented on an ordinal scale rather than absolute values.

The Ordinal Regression Problem (ORP) is the task to embed given data in the real numbers such that they are ordered according to their label, i. e. the target bin. An error is encountered whenever an ordering of two data points assigned to different bins is violated. Although the problem can be treated as a regular regression or classification problem, dedicated techniques are often preferred since they can account for the fact that the distance between ordinal classes in the data is unknown and not necessarily evenly distributed. Examples of ordinal regression include treatments such as the multi-class classification problem [FH01] and extensions of standard models such as the support vector machine (SVM) or learning vector quantization (LVQ) to ordinal regression tasks [SL02; CK07; FT12; TT17]. Recent work proposed an incremental and sparse Bayesian approach with favourable scaling properties [LdR18]. Often, ordinal regression is treated as a pairwise ranking problem [CMR19]. Further, there does exist recent theoretical work which establishes consistency of some surrogate losses for ordinal regression, which have better numeric properties [PBG17].

While several methods exist to solve the ordinal prediction problem itself, only a few consider the specific task of feature selection. The approach in [Gen+07] uses a minimal redundancy formulation based on a feature importance score to find the subset of relevant features. The work in [BES10] focuses on multiple filter methods which are adapted to ranking data. These models deliver sparse ordinal regression models which enable some insight into the underlying classification prescription. Yet, their result is arbitrary in the case of correlated or redundant features: if there does not exist a unique minimum relevant feature set, it often depends on arbitrary initialization or algorithmic design choices, which feature from a set of redundant features is chosen. Hence, weakly relevant features as described in Section 2.2 can easily be overlooked.

In this chapter, we will rely on the SVM-like treatments of the ORP due to the mathematical elegance and flexibility of this formulation [SL02; CK07; FT12]. We adapt the feature relevance bounds formulation first proposed for classification in [Göp+18] for this highly relevant setting as a solution to the all-relevant feature selection problem (ARFS) and demonstrate the benefit of this approach in comparison to alternative popular feature selection models such as Lasso or EN.

Besides formal mathematical modelling using linear optimization tasks, we will also demonstrate the suitability of the model to investigate the role of critical features for an ORP. As an example, the integration of criteria such as age, gender, or ethnicity might improve the prediction accuracy of a given model as measured by an appropriate cost function—yet, it might be debatable if these features can have any relevance for the given task as regards a causal relationship on the one hand; on the other hand, it might be unethical or impossible to gather such features for a prediction model in its daily use. Examples for a questionable impact of such characteristics on a formal model have recently been debated under the umbrella of model fairness [Kea17]. We will discuss how feature relevance profiles, in particular the identification of weakly relevant features, enable further insight into such settings, by explicitly quantifying the possible impact of such features.

In Section 4.1.2 we recapture two large margin ordinal regression formalizations, which differ in the type of constraints they enforce on ordinal classes, namely *implicit* and *explicit* constraints. We propose a new extension to determine feature relevance bounds, which can be transferred to several linear optimization problems and enables accurate all-relevant feature selection using the methodology proposed in Chapter 3. In Section 4.1.3 we perform several benchmarks to highlight the feature selection accuracy using quantitative measures on synthetic and real data. In summary, we seek answers to research questions 1, 2 and 4 from Section 1.2 in the context of ordinal regression, and additionally, in Section 4.2 we answer research question 3 by introducing privileged information.

4.1.2 *Methodology*

We consider the following ordinal regression learning task: We assume class labels $L = \{1, 2, \ldots, l\}$, which are ordered; w.l.o.g. we represent those as natural numbers. We assume training data are given, $X = \{x_i^j \in \mathbb{R}^n \mid i = 1, \ldots, m_j, j \in L\}$ where data point x_i^j is assigned the class label $j \in L$, i. e. x_i^j is contained in bin number j. The full data set has size $m := m_1 + \cdots + m_l$.

Here the index j refers to the ordinal target variable the data point x_i^j belongs to. The ORP can be phrased as the search for a mapping $f : \mathbb{R}^n \to \mathbb{R}$, which preserves the ordering of bins as indicated by the label information. That means the inequality $f(\boldsymbol{x}_{i_1}^{j_1}) < f(\boldsymbol{x}_{i_2}^{j_2})$ should hold for all pairs of class labels $j_1 < j_2$ and data indices i_1 and i_2 in these bins.

In the following, we will restrict to the case of a linear function, i. e. $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$ with parameter $\boldsymbol{w} \in \mathbb{R}^n$. In particular, in the case of high dimensional data, such a linear prescription is often sufficient to model the underlying regularity. Further, it enables a particularly strong link of feature relevancies and the underlying model, as already elaborated in popular sparse models such as Lasso [Tib96]. There do exist different possibilities to model the ORP learning problem. Here, we will introduce two existing optimization problems, which rely on large margins, and which treat the inequality constraints in two different ways.

Explicit Order Constraints One way to model ordinal regression is by an embedding of data in the real numbers via f, whereby the bins are separated by adaptive thresholds b_j, which are learned accordingly. A popular formulation which is inspired by support vector machines imposes a margin around all thresholds b_j for this embedding [CK07]:

$$\min_{\boldsymbol{w},\boldsymbol{b},\chi,\xi} \quad \frac{1}{2}\|\boldsymbol{w}\|_1 + C\sum_{i,j}\left(\chi_i^j + \xi_i^j\right) \tag{4.1}$$

s.t. for all i,j

$$\begin{aligned} &\boldsymbol{w}^\top \boldsymbol{x}_i^j - b_j \leq -1 + \chi_i^j \\ &\boldsymbol{w}^\top \boldsymbol{x}_i^{j+1} - b_j \geq +1 - \xi_i^{j+1} \\ &b_j \leq b_{j+1} \\ &\chi_i^j \geq 0, \xi_i^j \geq 0 \end{aligned} \tag{4.2}$$

where χ_i^j and ξ_i^j are slack variables. The thresholds b_j for $j = 1, \ldots, l-1$ determine the boundaries which separate the classes, b_j referring to the boundary in between bin j and bin $j+1$. The hyper-parameter $C > 0$ controls the trade-off between the margin and the number of errors and it can be chosen through cross-validation. We adapt the problem from [CK07], which uses L_2 regularization, and use L_1 regularization in (4.1), aiming for sparse solutions. In this definition, the linear ordering of classes is enforced *explicitly* through constraint $b_j \leq b_{j+1}$. When we refer to 4.2 in the future, we specifically refer to the constraints of the problem.

Implicit Order Constraints Another definition highlighted in [CK05] enforces the ordering implicitly, by requiring that all data of bin 1 to j are embedded below the threshold b_j, all data from bins $j+1$ to l are above the threshold. This leads to the implicitly constrained problem:

$$\min_{\boldsymbol{w},\boldsymbol{b},\chi,\xi} \frac{1}{2}\|\boldsymbol{w}\|_1 + C\sum_{j=1}^{l-1}\left(\sum_{k=1}^{j}\sum_{i=1}^{n^k}\chi_{ki}^j + \sum_{k=j+1}^{l}\sum_{i=1}^{n^k}\xi_{ki}^j\right)$$

subject to

$$\begin{aligned} &\boldsymbol{w}^\top \boldsymbol{x}_i^k - b_j \leq -1 + \chi_{ki}^j, \quad \chi_{ki}^j \geq 0, \\ &\qquad \text{for } k = 1, \ldots, j \text{ and } i = 1, \ldots, m_k; \\ &\boldsymbol{w}^\top \boldsymbol{x}_i^k - b_j \geq +1 - \xi_{ki}^j, \quad \xi_{ki}^j \geq 0 \\ &\qquad \text{for } k = j+1, \ldots, l \text{ and } i = 1, \ldots, m_k. \end{aligned} \tag{4.3}$$

Again, we adapt the existing problem from [CK05] and replace the existing regularization $\|\boldsymbol{w}\|_2$ with $\|\boldsymbol{w}\|_1$ to induce sparsity. In this definition, not only neighbouring classes are contributing to the overall loss of in between boundaries, but all other classes, as well. This can lead to more robust results in particular in the case of outliers, as shown in [CK05], but higher computational demand.

Modelling Relevance Bounds for ORP

In the following, we introduce feature relevance bounds for the explicit variant which is an extension from existing work for linear classification in [Göp+18] and Chapter 3. The definition for the implicit variant is very similar and can be found in Appendix A.1.1.

Assume a training set X is given. We denote an optimum solution of problem (Equation (4.1)) as $(\tilde{\boldsymbol{w}}, \tilde{\boldsymbol{b}}, \tilde{\boldsymbol{\xi}}, \tilde{\boldsymbol{\chi}})$. This solution induces the value

$$\mu_X := \frac{1}{2}\|\tilde{\boldsymbol{w}}\|_1 + C \cdot \sum_{i,j} \left(\tilde{\chi}_i^j + \tilde{\xi}_i^j \right)$$

which is uniquely determined by X. The quantity μ_X is unique by definition, albeit the solution $(\tilde{\boldsymbol{w}}, \tilde{\boldsymbol{b}}, \tilde{\boldsymbol{\xi}}, \tilde{\boldsymbol{\chi}})$ is not.

We are interested in the class of equivalent good hypotheses, i. e. all weight vectors $\boldsymbol{w}$ which yield (almost) the same quality as regards the regression error and generalization ability as the function induced by $\tilde{\boldsymbol{w}}$. This class might contain an infinite number of alternative hypotheses: in the context of correlated features, for example, we can trade one feature for the other. However, the function class cannot explicitly be computed, since the generalization ability is unknown for future data. We use the following surrogate induced by μ_X

$$\mathcal{F}_\delta(X) := \{\boldsymbol{w} \in \mathbb{R}^n \mid \exists \boldsymbol{b}, \boldsymbol{\xi}, \boldsymbol{\chi} \text{ such that constraints Equation (4.2) hold,} \\ \frac{1}{2}\|\boldsymbol{w}\|_1 + C \cdot \sum_{i,j} \left(\xi_i^j + \chi_i^j \right) \leq (1+\delta) \cdot \mu_X \}$$

These constraints ensure the following properties:

1. The empirical error of equivalent functions in $\mathcal{F}_\delta(X)$ is minimum, as measured by the slack variables.

2. The loss of the generalization ability is limited, as guaranteed by a small L_1-norm of the weight vector and learning theoretical guarantees as provided, e. g. by Theorem 7 in [Aga08] and Corollary 5 in [Zha02].

The parameter $\delta \geq 0$ quantifies the tolerated deviation to accept a function as yet good enough, C is determined by Problem (Equation (4.1)).

Solutions $\boldsymbol{w}$ in $\mathcal{F}_\delta(X)$ are sparse in the sense that irrelevant features are uniformly weighted as 0 for all solutions in $\mathcal{F}_\delta(X)$. Relevant but potentially redundant features can be weighted arbitrarily, disregarding sparsity, similar in spirit to the EN; yet the latter weights mutually redundant features equally and can therefore hide the relevance in the case of many redundant features [ZH05].

Feature selection and classification can be done with the rules from Section 3.2.1.

A feature is irrelevant for $\mathcal{F}_\delta(X)$ if it is neither strongly nor weakly relevant. The questions of strong and weak relevance can be answered via the following optimization problems:

Problem minRel(ℓ)**:**

$$\min_{\boldsymbol{w},\boldsymbol{b},\chi,\xi} |\boldsymbol{w}_\ell| \tag{4.4}$$

s.t. for all i, j Equation (4.2) holds and

$$\frac{1}{2}\|\boldsymbol{w}\|_1 + C \cdot \sum_{k,l} \left(\chi_k^l + \xi_k^l\right) \leq (1+\delta) \cdot \mu_X$$

Here $|\boldsymbol{w}_\ell|$ denotes the absolute value of feature ℓ in $\boldsymbol{w}$. Feature ℓ is strongly relevant for $\mathcal{F}_\delta(X)$ iff minRel(ℓ) yields an optimum larger than 0.

Problem maxRel(ℓ)**:**

$$\max_{\boldsymbol{w},\boldsymbol{b},\chi,\xi} |w_\ell|$$

s.t. for all i, j Equation (4.2) holds and

$$\frac{1}{2}\|\boldsymbol{w}\|_1 + C \cdot \sum_{k,l} \left(\chi_k^l + \xi_k^l\right) \leq (1+\delta) \cdot \mu_X$$

Now, features can be classified using the statistical threshold proposed in Section 3.2.1.

These two optimization problems span a real-valued interval for every feature ℓ with the result of minRel(ℓ) as lower and maxRel(ℓ) as upper bound. This interval characterizes the range of weights for ℓ occupied by good solutions in $\mathcal{F}_\delta(X)$. Hence, besides information about a feature's relevance, some indication about the degree up to which a feature is relevant or can be substituted by others is given. Note, however, that the solutions are in general not consistent estimators of an underlying 'true' weight vector as regards its exact value, as has been discussed, e. g. for Lasso [ZY06]. For consistency, it is advisable to use L$_2$ regularization after the selection of a set of relevant features.

Generalization Bounds At the beginning of Section 4.1.2 we introduced the set $\mathcal{F}_\delta(X)$ of all equivalent good hypotheses which yield (almost) the same quality regarding regression error and generalization ability. However, the impact of the norm of $\boldsymbol{w}$ and the high loss $\sum_{i,j}\left(\tilde{\chi}_i^j + \tilde{\xi}_i^j\right)$ are not considered separately, i. e. a low norm of $\boldsymbol{w}$ allows a high loss and vice versa. We would like to control the generalization error with L_1-regularization. To do so, we consider both quantities separately, i. e. we define

$$\begin{aligned}
&\mathcal{H}_\delta(\tilde{\boldsymbol{w}}) := \\
&\quad \Big\{\boldsymbol{w} \in \mathbb{R}^d \mid \exists \boldsymbol{b}, \boldsymbol{\xi}, \boldsymbol{\chi} \text{ such that constraints in Equation (4.2) hold,} \\
&\qquad\qquad \|\boldsymbol{w}\|_1 \leq (1+\delta)\|\tilde{\boldsymbol{w}}\|_1 \,, \sum_{i,j}\left(\xi_i^j + \chi_i^j\right) \leq \sum_{i,j}\left(\tilde{\xi}_i^{\,j} + \tilde{\chi}_i^j\right) \Big\}
\end{aligned}$$

This allows us to extend the results from [Göp+18] to our scenario, i. e. show that the generalization error of all hypothesis with the same or a lower high loss is bounded through the L_1-regularization. A proof is provided in Appendix A.1.2.

Feature Relevance Bounds as Linear Problem

The problems in the previous section are not yet linear, but they can be transferred to linear optimization problems, for which particularly efficient solvers are available.

Theorem 1. *Problem* minRel(ℓ) *is equivalent to the following linear optimization problem:*

$$\begin{aligned} \text{minRel}^*(\ell) : \min_{\boldsymbol{w},\boldsymbol{\hat{w}},\boldsymbol{b},\boldsymbol{\chi},\boldsymbol{\xi}} \; & \hat{\boldsymbol{w}}_\ell \textit{ s.t. for all } i,j \\ & \textit{Equation (4.2) holds} \\ & \frac{1}{2}\sum_k \hat{w}_k + C\cdot\sum_{k,l}\left(\chi_k^l + \xi_k^l\right) \leq (1+\delta)\cdot\mu_X && (4.5) \\ & \boldsymbol{w}_i \leq \hat{\boldsymbol{w}}_i,\; -w_i \leq \hat{\boldsymbol{w}}_i && (4.6) \end{aligned}$$

Problem maxRel(ℓ) *can be solved by taking the optimum of the following two linear optimization problems:*

$$\begin{aligned} \text{maxRel}^*_{pos}(\ell) : \max_{\boldsymbol{w},\boldsymbol{\hat{w}},\boldsymbol{b},\boldsymbol{\chi},\boldsymbol{\xi}} \; & \hat{\boldsymbol{w}}_\ell \textit{ s.t. for all } i,j \\ & \textit{Equation (4.2) holds} \\ & \frac{1}{2}\sum_k \hat{w}_k + C\cdot\sum_{k,l}\left(\chi_k^l + \xi_k^l\right) \leq (1+\delta)\cdot\mu_X \\ & \boldsymbol{w}_i \leq \hat{\boldsymbol{w}}_i,\; -w_i \leq \hat{\boldsymbol{w}}_i \\ & \hat{\boldsymbol{w}}_\ell \leq \boldsymbol{w}_\ell && (4.7) \end{aligned}$$

and the problem

$$\begin{aligned} \text{maxRel}^*_{neg}(\ell) : \max_{\boldsymbol{w},\boldsymbol{\hat{w}},\boldsymbol{b},\boldsymbol{\chi},\boldsymbol{\xi}} \; & \hat{\boldsymbol{w}}_\ell \textit{ s.t. for all } i,j \\ & \textit{Equation (4.2) holds} \\ & \frac{1}{2}\sum_k \hat{w}_k + C\cdot\sum_{k,l}\left(\chi_k^l + \xi_k^l\right) \leq (1+\delta)\cdot\mu_X \\ & \boldsymbol{w}_i \leq \hat{\boldsymbol{w}}_i,\; -w_i \leq \hat{\boldsymbol{w}}_i \\ & \hat{\boldsymbol{w}}_\ell \leq -\boldsymbol{w}_\ell. \end{aligned}$$

The proof can be found in Appendix A.1.3.

In practice, it might be a good strategy to split the model constraint such as (4.5) into two, limiting the weight vector separately

$$\frac{1}{2}\sum_k \hat{w}_k \leq (1+\delta)\cdot\|\tilde{\boldsymbol{w}}\|_1$$

and error term

$$\sum_{k,l}\left(\chi_k^l + \xi_k^l\right) \leq \sum_{k,l}\left(\tilde{\chi}_k^l + \tilde{\xi}_k^l\right)$$

where the symbols marked $\tilde{\cdot}$ refer to the optimum solution of the original margin-based ordinal regression problem. This split enables better control of the loss of generalization ability and error terms, and it also mediates the dependency on the hyper-parameter C of the space of equivalent good functions. At a small down-side, this split depends on the found solution and it is no longer uniquely defined by the given training data, albeit we did not observe large variation in practical applications.

4.1.3 *Evaluation*

In this section, we show the quality of feature selection by evaluating the results of both the explicit and the implicit variant of our method, on theoretically generated data with known ground truth. Also, we compare both

Table 4.1: Artificially created data sets with known ground truth. The model of which the data is drawn from is based on the strongly relevant features. The weakly relevant features are linear combinations of strong ones. Characteristics of the sets are taken from [Göp+18] and [GPH17]. All sets have target variables with five ordinal classes.

Dataset	#Instances	#Strong	#Weak	#Irrelevant
Set 1	150	6	0	6
Set 2	150	0	6	6
Set 3	150	3	4	3
Set 4	256	6	6	6
Set 5	512	1	2	11
Set 6	200	1	20	0
Set 7	200	1	20	20
Set 8	1000	10	20	10
Set 9	1000	10	20	200

variants concerning their classification accuracy and run time on standard benchmark datasets. The accuracy is measured using the Macro-averaged Mean Absolute Error (MMAE) which is specifically designed for ordinal regression data with imbalanced classes:

$$MMAE = \frac{1}{l} \sum_{j=1}^{l} \frac{\sum_{i=1}^{m_j} \left| j - f(x_i^j) \right|}{m_j}, \tag{4.8}$$

where l is the number of bins, f refers to the bin the sample x_i^j is assigned to by the learned model, and m_j refers to the number of samples in class j.

Feature Selection Performance

We adapt the generation method presented in [Göp+18] and Section 3.3.1 for ordinal regression. By using equal frequency binning we convert a continuous regression variable into an ordered discrete target variable with five ordinal classes. The data is generated from a suitable set of informative features. From those, we form strongly relevant features by simply picking the desired number out of the informative set. Weakly relevant features are created as linear combinations of informative features. Finally, irrelevant features are drawn from random Gaussian noise. All features are normalized to zero mean and unit variance. The exact characteristics of the datasets used in our experiments are shown in Table 4.1.

For evaluation, we use the F_1-measure as defined in Section 3.3.2 to quantify the detection of the all relevant feature set found by our method (dubbed feature relevance interval - FRI)[1] concerning the true all-relevant features of the data. Our method utilizes the statistical feature selection threshold proposed in Section 3.2.1.

Because of the lack of other feature selection methods in this context we emulate the behaviour of Lasso [Tib96] and the EN (EN) [ZH05]. For that we utilize a Recursive Feature Elimination with Cross-Validation (RFECV)[2], using the ordinal regression model given by Equation (4.1) with an EN penalty and parameter p. The parameter p, controlling the ratio between

[1] Implementation in Python: https://github.com/lpfann/fri

[2] Implementation in Python: RFECV from scikit-learn

the L_1 and L_2 norm of the EN model, is optimized with a search over the values $p \in \{0, 0.01, 0.1, 0.2, 0.5, 0.7, 1\}$. Setting $p = 0$ corresponds to a Lasso like sparsity constraint, and we test that scenario explicitly. Our surrogates are called $\mathrm{M}_e^{L_1}$ (Lasso) and $\mathrm{M}_e^{L_1+L_2}$ (EN), both based on the explicit variant. Hyper-parameters are selected according to 5-fold cross-validation, and all scores are averaged over 30 independent runs.

The results are given in Tables 4.2 and 4.3, where FRI_e and FRI_i denote the explicit and the implicit variant respectively. Because Lasso and EN performed nearly identical we only give the results for the EN. The results show that FRI in both variants is superior to $\mathrm{M}_e^{L_1+L_2}$ on every data set, especially for clean data where it scores nearly perfect on every measure. It only shows slightly worse precision in Set 9 where the feature space is big. $\mathrm{M}_e^{L_1+L_2}$ on the other hand is very precise in that setting but selects only 37% of relevant features. Having shown that, we are now interested in which of the two FRI variants is performing better. Since they both score perfectly on clean data, we increase the challenge by adding Gaussian noise with a standard deviation of $\sigma = 0.5$ to all sets. The theory, as given in [CK05], indicates that the implicit variant should perform better on noisy data, because for every decision boundary to be determined it has access to more data samples than the explicit variant, thus gaining an advantage in stability. However, our experiments do not support this notion as both variants of FRI perform equally well on noisy data. Interestingly, the $\mathrm{M}_e^{L_1+L_2}$ improved its performance on those sets with a lot of weakly relevant features. This could be explained by assuming that the model has to rely on more of the weak, thus inter-correlated features, to regain the information that was lost due to the introduction of the noise.

Table 4.2: Artificially created data sets with known ground truth and evaluation of the identified relevant features by the methods as compared to all relevant features. The score is averaged over 30 independent runs. $\mathrm{M}_e^{L_1+L_2}$ represents the surrogate model for the EN with RFECV.

		Clean data		
score	data	$\mathrm{M}_e^{L_1+L_2}$	FRI_e	FRI_i
F_1	Set 1	0.94	1.0	1.0
	Set 2	0.79	1.0	1.0
	Set 3	0.81	1.0	1.0
	Set 4	0.83	1.0	1.0
	Set 5	0.83	1.0	1.0
	Set 6	0.25	1.0	1.0
	Set 7	0.49	1.0	1.0
	Set 8	0.95	1.0	1.0
	Set 9	0.53	0.98	0.98
Precision	Set 1	0.90	1.0	1.0
	Set 2	0.86	1.0	1.0
	Set 3	0.95	1.0	1.0
	Set 4	0.95	1.0	1.0
	Set 5	0.89	1.0	1.0
	Set 6	1.0	1.0	1.0
	Set 7	0.97	1.0	1.0
	Set 8	0.91	1.0	1.0
	Set 9	1.0	0.97	0.97
Recall	Set 1	1.0	1.0	1.0
	Set 2	0.82	1.0	1.0
	Set 3	0.74	1.0	1.0
	Set 4	0.77	1.0	1.0
	Set 5	0.84	1.0	1.0
	Set 6	0.15	1.0	1.0
	Set 7	0.41	1.0	1.0
	Set 8	1.0	1.0	1.0
	Set 9	0.37	1.0	1.0

Table 4.3: Artificially created data sets with known ground truth and evaluation of the identified relevant features by the methods as compared to all relevant features. The data was generated and Gaussian noise (standard deviation $\sigma = 0.5$) was added to the predictors. The score is averaged over 30 independent runs. $M_e^{L_1+L_2}$ represents the surrogate model for the EN with RFECV.

		Noisy data		
score	data	$M_e^{L_1+L_2}$	FRI_e	FRI_i
F_1	Set 1	0.92	0.95	0.98
	Set 2	0.89	0.97	0.98
	Set 3	0.85	0.97	0.96
	Set 4	0.80	0.96	0.97
	Set 5	0.86	1.0	1.0
	Set 6	0.56	0.94	0.94
	Set 7	0.46	0.90	0.91
	Set 8	0.80	0.98	0.98
	Set 9	0.60	1.0	1.0
Precision	Set 1	0.87	1.0	1.0
	Set 2	0.86	1.0	1.0
	Set 3	0.90	1.0	1.0
	Set 4	0.91	1.0	1.0
	Set 5	0.81	1.0	1.0
	Set 6	1.0	1.0	1.0
	Set 7	0.84	1.0	1.0
	Set 8	0.95	1.0	1.0
	Set 9	1.0	1.0	1.0
Recall	Set 1	0.99	0.92	0.96
	Set 2	0.94	0.96	0.96
	Set 3	0.83	0.95	0.93
	Set 4	0.74	0.93	0.94
	Set 5	0.99	1.0	1.0
	Set 6	0.40	0.89	0.89
	Set 7	0.35	0.84	0.86
	Set 8	0.70	0.97	0.97
	Set 9	0.43	1.0	1.0

Real Data

Here, we evaluate the model on benchmark data as described in [ZH05; Sán+13] without regarding feature selection. The imbalanced ordinal regression data sets used in the experiments are listed in Table 4.4. All samples are normalized to zero mean and unit variance.

Model Accuracy We replicate the experiments which have been presented in [FT12; TT17] to evaluate the performance of our two possible underlying SVM models as stated in Section 4.1. Our models, which we will call $\mathrm{M}_e^{L_1}$ and $\mathrm{M}_i^{L_1}$ in the following, were tuned using 5-fold cross-validation and used all available features previous feature selection, i. e. the models do not use the procedure described in Section 3.2.1 and the scores are based on all features without retraining. The results are averaged over the same 30 folds as used in [TT17] and evaluation is based on the MMAE as defined in (4.8). We compare our models with p-OGMLVQ and a-OGMLVQ, the best performing methods for the given data as stated in [FT12]. Results for the EN surrogate $\mathrm{M}_e^{L_1+L_2}$ were omitted because they were nearly identical to $\mathrm{M}_e^{L_1}$.

The outcomes for MMAE are reported in Table 4.5. Overall the explicit variant $\mathrm{M}_e^{L_1}$ outperforms the implicit variant $\mathrm{M}_i^{L_1}$ in all cases except one when considering MMAE. Similarly, the runtime is given in Table 4.6. We can see that $\mathrm{M}_e^{L_1}$ is at least two times faster, in some cases even over 20 times faster. When comparing with the existing results of a-OGMLVQ, we can see $\mathrm{M}_e^{L_1}$ outperforming it in 5 cases while being worse in 5 others, it can beat p-OGMLVQ in 6 cases and closely ties in one case (TAE).

Feature Set Size For feature relevance, no ground truth is available for the given data, rendering us unable to perform the same evaluation as for the artificial sets. We are only able to compare the number of features provided by our method with feature selection (FRI) and the previously used model $\mathrm{M}_e^{L_1+L_2}$ as a surrogate for EN with RFECV. Table 4.7 lists the average number of features identified as relevant for both techniques. For three data sets (Squash-stored, Squash-unstored, TAE), FRI identifies a smaller number of relevant features than the alternative, while yielding the same accuracy. For three further data sets (Automobile, Eucalyptus, Pasture), FRI identifies more (weakly relevant) features. In all cases, FRI potentially offers more information

Table 4.4: Real ordinal regression benchmark data sets with imbalanced classes taken from [Sán+13], where d is the number of features, and K is the number of classes.

Dataset	#Instances	d	K	Ordered Class Distribution
Automobile	205	71	6	(3,22,67,54,32,27)
Bondrate	57	37	5	(6,33,12,5,1)
Contact-lenses	24	6	3	(15,5,4)
Eucalyptus	736	91	5	(180,107,130,214,105)
Newthyroid	215	5	3	(30,150,35)
Pasture	36	25	3	(12,12,12)
Squash-stored	52	51	3	(23,21,8)
Squash-unstored	52	52	3	(24,24,4)
TAE	151	54	3	(49,50,52)
Winequality-red	1599	11	6	(10,53,681,638,199,18)

Table 4.5: Comparison of both proposed variants of ordinal regression models from Section 4.1. Benchmark on real ordinal datasets [Sán+13] by averaged MMAE. Folds are identical to [TT17] and are comparable.

	MMAE			
data	p-OGMLVQ	a-OGMLVQ	$\text{M}_e^{L_1}$	$\text{M}_i^{L_1}$
Automobile	0.482	**0.446**	0.532	0.516
Bondrate	0.768	**0.737**	0.939	0.949
Contact-lenses	0.243	0.221	**0.190**	0.265
Eucalyptus	0.450	0.477	**0.390**	**0.390**
Newthyroid	0.124	0.097	**0.043**	0.045
Pasture	**0.307**	0.318	0.374	0.430
Squash-stored	0.415	0.411	**0.371**	**0.371**
Squash-unstored	0.488	**0.228**	0.280	0.300
TAE	0.553	**0.537**	0.552	0.664
Winequality-red	1.078	1.069	0.868	**0.790**

Table 4.6: Comparison of both proposed variants of ordinal regression models from Section 4.1. Recorded is the aggregated runtime in seconds over 30 folds on real ordinal datasets [Sán+13].

	Runtime (s)	
data	$\text{M}_e^{L_1}$	$\text{M}_i^{L_1}$
Automobile	151.6	876.8
Bondrate	49.7	133.6
Contact-lenses	23.7	53.9
Eucalyptus	768.7	3280.3
Newthyroid	37.5	92.3
Pasture	28.6	57.0
Squash-stored	36.0	68.9
Squash-unstored	35.9	69.4
TAE	43.3	83.4
Winequality-red	349.4	8359.4

than EN by discriminating between weakly and strongly relevant features and giving more candidate features to consider which can then be verified in practice.

Table 4.7: Average feature set size of FRI model with explicit constraints and EN surrogate model ($M_e^{L_1+L_2}$) with RFECV on real datasets [ZH05; Sán+13]. FRI allows extra discrimination between strong (FRI^s) relevance and weak (FRI^w) relevance.

	Average Feature Set Size			
data	FRI_e^s		FRI_e^w	$M_e^{L_1+L_2}$
Automobile	4.5	∪	12.6	4.0
Bondrate	0.0	∪	5.4	2.0
Contact-lenses	0.9	∪	1.1	2.0
Eucalyptus	2.1	∪	33.2	15.6
Newthyroid	0.0	∪	4.7	2.0
Pasture	0.0	∪	15.5	6.0
Squash-stored	2.4	∪	7.9	11.1
Squash-unstored	1.8	∪	3.3	8.0
TAE	1.9	∪	5.4	16.8
Winequality-red	0.0	∪	7.6	5.4

Qualitative Evaluation on COMPAS

To showcase a possible application of our approach, we use FRI to examine the COMPAS dataset. This data was created by Propublica, a journalistic collective from New York, and consists of personal information regarding the criminal history of 11757 people from Broward County in Florida. Data like this has been used to predict an individuals risk of recidivism after a criminal offence. Hereby, previous analyses have shown [Ang+16] that racial bias is incorporated in at least one standard algorithmic prediction tool, meaning that African American individuals receive higher risk scores than Caucasian people. While it remains an open research question if and how an algorithm should use socially sensitive attributes [HD13; Har+16] we are now interested which information is used by our linear ordinal regression model based on the FRI analysis on the given data. As such we try to find possible causes for direct or indirect discrimination [PRT08] and facilitate careful model design, which seems to be necessary when aiming for long term impact of fair machine learning[Liu+18]. From the originally 28 features of the dataset, we scale down to ten by eliminating all identifying and time-related information, which do not contribute information to the prediction task. These features are described in detail in Appendix A.1.7. We build a predictive model on the data, showing the relevancy of our features to that model. The result is shown in the upper plot in Figure 4.1. In this kind of plot, the relevance intervals are shown as vertical bars such that the maximum and minimum heights represent maxRel and minRel. For better comparison, the values are normalized to the L_1 norm of the optimal model ($\|\tilde{w}\|_1$). We also add the maximum element in $\Pi(\text{maxRel})$ as horizontal dashes, which represents the threshold which is used to classify between weakly relevant and irrelevant features.

The predictive accuracy is 66.73% which is line with results from the Propublica analysis. Note that the models used in practice deviate from the ones considered here, and the former are not available to us. Thus, we discuss properties of the linear models found by the proposed ORP only, not any other model.

Two features are strongly relevant, namely, the count of prior charges and the age group 17-25 which show a big contribution in absolute terms. Many

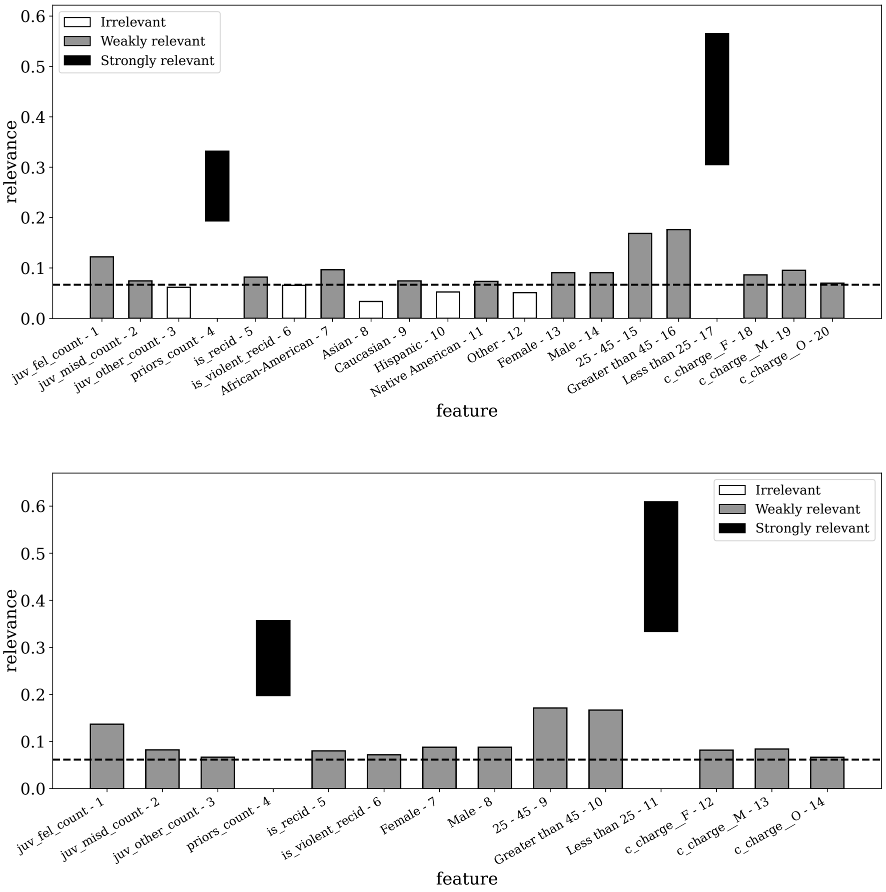

Figure 4.1: Relevance plots for the COMPASS dataset. Top: Relevance intervals (bars) for all features including ethnicity. Bottom: Relevance intervals for all features when ethnicity is eliminated from the data. Ethnicity is not a relevant factor for the model on top, so if those variables are eliminated, the relevancy of the other features do not change profoundly. The y-axis represents the computed feature relevance normalized to the L_1 norm of the optimal model.

other features, such as the count of juvenile felonies and misdemeanours, or the degree of criminal changes are weakly relevant. More interestingly, socially sensitive features such as sex and race are also considered weakly relevant. In the case of sex, both male and female exhibit the same maximal relevance which hints at the anti-correlation between the two features. In the case of race, being African-American, Caucasian or Native American is considered weakly relevant. When compared with the Propublica analysis, our relevance bounds are in line with their results.

To measure the contribution of the ethnic features in the model, we repeat the experiment with all those features removed. Hereby, the accuracy does not drop significantly, yielding 65.99%. The bottom plot of Figure 1 shows the relevance for all remaining features. Compared to the previous model, there are two notable changes. The count of juvenile offences and the information about violent recidivism become relevant which are intuitively much more important to the problem at hand and do not reiterate a potential bias in society.

4.2 Learning using privileged information

4.2.1 *Background*

Context	
Problem	*ordinal regression*
Model	*linear*
Type	*privileged*

The scenario of privileged information phrases the situation, that some features are available during the training phase, but not during the testing phase, e. g. due to the costs, computational load or any other restrictions. In classical machine learning, it is commonly assumed, that training and test set have an identical statistical distribution and utilize the same predictive features. In contrast, the Learning using privileged information (LUPI) paradigm [VV09] considers additional privileged information only available at training time. This paradigm can be understood as an intelligent teacher feeding the learner extra information to improve the learning process [VI15]. Additional information could be the output of another model ('machines-teaching-machines') or input from a human expert itself, who intuitively knows which examples in the data are hard to discriminate. Examples are medical measurements which require invasive techniques or measurements which require too much time in daily use but would be affordable for training.

To incorporate privileged information the authors in [VV09] proposed a variant of the SVM that uses privileged information for training. The modelling replaces or enriches slack variables, which are required by the soft-margin SVM to correct for hard training samples. This specific approach is known as *similarity control* [VI15]. The approach in [VV09] introduces the SVM+ in which a smooth function based on the Privileged Information (PI) is used at training time to improve learning in non-separable classification settings. The method [Tan+15] refrained from fully replacing the slack variables and combined them with a smooth function based on PI. It achieved better generalization ability and lower complexity models. Furthermore, this approach also extends the SVM+ to ordinal regression problems.

While approaches to incorporate privileged information exist, and it has been shown that LUPI has the potential to speed up learning [PV10], the analysis of feature relevancies in the context of redundant feature information is still new. Especially the solution to the ARFS is not considered anywhere.

In this section, we will expand upon the work done in Section 4.1 and introduce feature relevance bounds for privileged ordinal regression to solve the ARFS and answer research question 3 from Section 1.2 and allows us to define relevance for privileged features.

4.2.2 *Methodology*

Let us shortly recall the classical setting considered so far: Given ordered class labels $L = \{1, 2, \dots, l\}$ and training data $X = \{\boldsymbol{x}_i^j \in \mathbb{R}^n \mid i = 1, \dots, m_j, j \in L\}$ where data point x_i^j is assigned the class label $j \in L$. The full data set has size $m := m_1 + \cdots + m_l$. Here the index j refers to the ordinal target variable (represented by b_j) the data point x_i^j belongs to.

In the LUPI setting, we work with two sets of data X and $X^* = \{\boldsymbol{x}_i^{*j} \in \mathbb{R}^{n^*} \mid i = 1, \dots, m_j\ ;\ j \in L\}$ which is a set of additional information PI where n^* is the number of privileged features we have available. The information is privileged in the sense that it is not available in the testing and prediction phase, and it is only present when training the model. This fact does not necessarily imply that the privileged information is of higher quality or exhibits correlation with the label at all. Rather, there are reasons why it cannot be gathered at prediction time: examples are too costly computations (such as extensive feature preprocessing), unavailability of sensors, unavailability of the information (such as information which is available only in retrospective), or privacy issues which prevent gathering the data (such as personal information). X and X^*, in general, do not have to share the same space or modality. As an example, X could cover numerical features, and X^* could be textual input from an expert.

Modelling Slacks in Ordinal Regression There are several ways to integrate privileged information into the learning model [Lop+16]. In the following, we only consider *similarity control* where privileged information is interpreted as the teacher giving hints about the difficulty for each training example. These hints can be incorporated into an SVM through slack variables which was shown in [Tan+15] already. In the following, we will extend our *explicit* definition of ordinal regression to handle privileged information by adapting similarity control as used in [Tan+15].

We recall that in the explicit variant two types of slacks are used. Each slack value represents a deviation from the classification rule. In the LUPI case, we replace χ_i^j by

$$p_\chi^j(\boldsymbol{x}_i^*) := \left(\boldsymbol{w}_\chi^* \cdot \boldsymbol{x}_i^{*j} + d_\chi\right)$$

and ξ_i^j by the function

$$p_\xi^j(\boldsymbol{x}_i^*) := \left(\boldsymbol{w}_\xi^* \cdot \boldsymbol{x}_i^{*j} + d_\xi\right).$$

Then the model is defined as

$$\begin{aligned}
\min_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{w}^*,\mathbf{d}} \ & \frac{1}{2}\|\boldsymbol{w}\|_1 + \frac{\gamma}{2}(\|\boldsymbol{w}_\chi^*\|_1 + \|\boldsymbol{w}_\xi^*\|_1) + C\sum_{j=1}^{l}\sum_{i=1}^{n^k}\left(p_\chi^j(\boldsymbol{x}_i^*) + p_\xi^j(\boldsymbol{x}_i^*)\right) \\
& \text{s.t. for every } j = 1,\dots,l-1 \\
& \boldsymbol{w}^\top \boldsymbol{x}_i^j - b_j \le -1 + p_\chi^j(\boldsymbol{x}_i^*) \\
& \boldsymbol{w}^\top \boldsymbol{x}_i^{j+1} - b_j \ge +1 - p_\xi^{j+1}(\boldsymbol{x}_i^*) \\
& b_j \le b_{j+1} \\
& p_\chi^j(\boldsymbol{x}_i^*) \ge 0,\ p_\xi^j(\boldsymbol{x}_i^*) \ge 0\,.
\end{aligned} \tag{4.9}$$

The parameter γ scales the influence of privileged information. This allows us to reject nonsense PI by simplifying the model and relying solely on X when considering a cross-validation scheme where we expect better generalization ability by a simpler model. The adaption of [Tan+15] now enables us to define relevance bounds as in Section 4.1.2.

Formalization

We now consider two sets of features. In the following, we define bounds for both regarding their relevance to the machine learning procedure when both sets are present. Because PI features are not present while predicting they are always irrelevant for that phase. They are relevant to speed up learning by mediating the distribution of slack variables.

Assume a training set

$$X = \{\boldsymbol{x}_i^j \in \mathbb{R}^n\}$$

with PI

$$X^* \ = \ \{\boldsymbol{x}_i^{*j} \in \mathbb{R}^{n^*}\}.$$

We define

$$\mathcal{L} := C \sum_{j=1}^{l} \sum_{i=1}^{n^k} \left(p_\chi^j(\boldsymbol{x}_i^*) + p_\xi^j(\boldsymbol{x}_i^*) \right)$$

as the total slack loss of problem (Equation (4.9)). Denote an optimum solution to the problem as

$$(\tilde{\boldsymbol{w}}, \tilde{\boldsymbol{b}}, \tilde{\boldsymbol{w}}_\chi^*, \tilde{\boldsymbol{w}}_\xi^*, \tilde{d_\chi}, \tilde{d_\xi})$$

and its total loss as $\tilde{\mathcal{L}}$. Analogous to Section 4.1.2, this solution induces the value

$$\mu_{X,X^*} := \frac{1}{2}\|\tilde{\boldsymbol{w}}\|_1 + \frac{\gamma}{2}(\|\tilde{\boldsymbol{w}}_\chi\|_1 + \|\tilde{\boldsymbol{w}}_\xi\|_1) + \tilde{\mathcal{L}}.$$

Furthermore, we use the following proxy induced by μ_{X,X^*}

$$\begin{aligned}\mathcal{F}_\delta(X, X^*) := \Big\{ &\boldsymbol{w} \in \mathbb{R}^n,\ \boldsymbol{w}_\chi^*, \boldsymbol{w}_\xi^* \in \mathbb{R}^{n^*} \mid \exists \boldsymbol{b}, d_\chi, d_\xi \\ &\text{s.t. Equation (4.9) holds and} \\ &\frac{1}{2}\|\boldsymbol{w}\|_1 + \frac{\gamma}{2}(\|\boldsymbol{w}_\chi^*\|_1 + \|\boldsymbol{w}_\xi^*\|_1) + \mathcal{L} \leq (1+\delta) \cdot \mu_{X,X^*} \Big\}\end{aligned}$$

This proxy allows us to define similar feature relevancy rules as proposed in [Göp+18]. While the rules are defined in relation to a numerical 0, in practice we instead use the statistical bound proposed in Section 3.2.1, which yields higher accuracy. For brevity, we here utilize the original formulation, and define the rules for non-privileged feature ℓ in X as:

Strong relevance of feature ℓ for $\mathcal{F}_\delta(X, X^*)$: Is feature ℓ relevant for all hypotheses in $\mathcal{F}_\delta(X, X^*)$, i. e. all weight vectors $\boldsymbol{w} \in \mathcal{F}_\delta(X, X^*)$ yield $\mathcal{F}_\ell \neq 0$?

Weak relevance of feature ℓ for $\mathcal{F}_\delta(X, X^*)$: Is feature ℓ relevant for at least one hypothesis in $\mathcal{F}_\delta(X, X^*)$ in the sense that one weight vector $\boldsymbol{w} \in \mathcal{F}_\delta(X, X^*)$ exists with $\mathcal{F}_\ell \neq 0$, but this does not hold for all weight vectors in $\mathcal{F}_\delta(X, X^*)$?

Irrelevance of feature ℓ for $\mathcal{F}_\delta(X, X^*)$: Is feature ℓ irrelevant for every hypothesis in $\mathcal{F}_\delta(X, X^*)$, i. e. all weight vectors $\boldsymbol{w} \in \mathcal{F}_\delta(X, X^*)$ yield $\mathcal{F}_\ell = 0$?

and similarly for feature p in X^* with

$$\boldsymbol{w}_\bullet^* := \{\boldsymbol{w}_\chi^*, \boldsymbol{w}_\xi^* \mid (\boldsymbol{w}^*, \boldsymbol{w}_\chi^*, \boldsymbol{w}_\xi^*) \in \mathcal{F}_\delta(X, X^*)\} :$$

Strong relevance of feature p for $\mathcal{F}_\delta(X, X^*)$: Is feature p relevant for all hypotheses in $\mathcal{F}_\delta(X, X^*)$, i. e. for all $\boldsymbol{w}_\bullet^*$ in $\mathcal{F}_\delta(X, X^*)$ at least one weight vector in $\boldsymbol{w}_\bullet^*$ for one bin of the ordered classes yields $\boldsymbol{w}_{\bullet p}^* \neq 0$?

Weak relevance of feature p for $\mathcal{F}_\delta(X, X^*)$: Is feature p relevant for at least one hypothesis in $\mathcal{F}_\delta(X, X^*)$ in the sense that one weight vector $\boldsymbol{w}_\bullet^*$ exists with $\boldsymbol{w}_{\bullet p}^* \neq 0$, but this does not hold for all $\boldsymbol{w}_\bullet^*$ in $\mathcal{F}_\delta(X, X^*)$?

Irrelevance of feature p for $\mathcal{F}_\delta(X, X^*)$: Is feature p irrelevant for every hypothesis in $\mathcal{F}_\delta(X, X^*)$, i. e. all weight vectors $\boldsymbol{w}_\bullet^*$ yield $\boldsymbol{w}_{\bullet p}^* = 0$?

A feature is irrelevant for $\mathcal{F}_\delta(X, X^*)$ if it is neither strongly nor weakly relevant.

The questions of strong and weak relevance can be answered via the following optimization problems:

Problem minRel(p)**:**

$$\max_{\bullet \in \{\chi, \xi\}} \min_{\boldsymbol{w}, \boldsymbol{w}^*_\bullet, \boldsymbol{b}, d_\bullet} |\boldsymbol{w}^*_{\bullet p}| \text{ s.t. for all } i, j \text{ Equation (4.9) holds and}$$

$$\frac{1}{2}\|\boldsymbol{w}\|_1 + \frac{\gamma}{2}(\|\boldsymbol{w}^*_\chi\|_1 + \|\boldsymbol{w}^*_\xi\|_1) + \mathcal{L} \leq (1+\delta) \cdot \mu_{X,X^*}$$

Because of two slack functions and the corresponding weights $\boldsymbol{w}^*_\chi$ and $\boldsymbol{w}^*_\xi$ we need to optimize two inner feature relevancies $|\boldsymbol{w}^*_{\bullet p}|$. To aggregate them to a global feature relevance we take the maximum to express that a feature could be used only in one of both functions, i. e. it is not relevant for all slack functions but at least in one. One could define an additional relevance classification by taking into account cases where the $\min\min > 0$, i. e. the feature is relevant for all slack functions. In the following, we limit ourselves to the former case.

Feature p is strongly relevant for $\mathcal{F}_\delta(X, X^*)$ iff minRel(p) yields an optimum larger than 0.

Problem maxRel(p)**:**

$$\max_{\bullet \in \{\chi, \xi\}} \max_{\boldsymbol{w}, \boldsymbol{w}^*_\bullet, \boldsymbol{b}, \boldsymbol{\chi}, \boldsymbol{\xi}} |\boldsymbol{w}^*_{\bullet p}| \text{ s.t. for all } i, j \text{ Equation (4.9) holds and}$$

$$\frac{1}{2}\|\boldsymbol{w}\|_1 + \frac{\gamma}{2}(\|\boldsymbol{w}^*_\chi\|_1 + \|\boldsymbol{w}^*_\xi\|_1) + \mathcal{L} \leq (1+\delta) \cdot \mu_{X,X^*}$$

Similar to the first problem we consider the maximum inner feature relevance to express the global feature relevance.

Feature p is weakly relevant for $\mathcal{F}_\delta(X, X^*)$ iff minRel(p) yields an optimum 0 and maxRel(p) yields an optimum larger than 0.

Linear Problem Formulation

Both problems from the previous section can be transferred to linear optimization problems:

Theorem 2. *Problem* minRel(p) *is equivalent to taking the maximum over following two linear optimization problems*

$$\mathrm{minRel}^*_\chi(p) : \min_{\substack{\boldsymbol{w},\hat{\boldsymbol{w}},\boldsymbol{w}^*_\chi,\widehat{\boldsymbol{w}^*_\chi},\boldsymbol{w}^*_\xi,\widehat{\boldsymbol{w}^*_\xi},\\ \boldsymbol{b},d_\chi,d_\xi}} \hat{\boldsymbol{w}}^*_{\chi p}$$

s.t. for all i, j *Equation* (4.9) *holds and*

$$\begin{aligned}
&\frac{1}{2}\sum_k \hat{\boldsymbol{w}}_k + \frac{\gamma}{2}\sum_k \hat{\boldsymbol{w}}^*_{\chi k} + \frac{\gamma}{2}\sum_k \hat{\boldsymbol{w}}^*_{\xi k} + \mathcal{L} \le (1+\delta)\cdot \mu_X \\
&\boldsymbol{w}_i \le \hat{\boldsymbol{w}}_i,\ -\boldsymbol{w}_i \le \hat{\boldsymbol{w}}_i \\
&\chi_i \le \hat{\chi}_i,\ -\chi_i \le \hat{\chi}_i \\
&\xi_i \le \hat{\xi}_i,\ -\xi_i \le \hat{\xi}_i
\end{aligned}$$

and

$$\mathrm{minRel}^*_\xi(p) : \min_{\substack{\boldsymbol{w},\hat{\boldsymbol{w}},\boldsymbol{w}^*_\chi,\widehat{\boldsymbol{w}^*_\chi},\boldsymbol{w}^*_\xi,\widehat{\boldsymbol{w}^*_\xi},\\ \boldsymbol{b},d_\chi,d_\xi}} \hat{\boldsymbol{w}}^*_{\xi p}$$

s.t. for all i, j *Equation* (4.9) *holds and*

$$\begin{aligned}
&\frac{1}{2}\sum_k \hat{\boldsymbol{w}}_k + \frac{\gamma}{2}\sum_k \hat{\boldsymbol{w}}^*_{\chi k} + \frac{\gamma}{2}\sum_k \hat{\boldsymbol{w}}^*_{\xi k} + \mathcal{L} \le (1+\delta)\cdot \mu_X \\
&\boldsymbol{w}_i \le \hat{\boldsymbol{w}}_i,\ -\boldsymbol{w}_i \le \hat{\boldsymbol{w}}_i \\
&\chi_i \le \hat{\chi}_i,\ -\chi_i \le \hat{\chi}_i \\
&\xi_i \le \hat{\xi}_i,\ -\xi_i \le \hat{\xi}_i\ .
\end{aligned}$$

For maxRel(p) *we define the linear optimization problem*

$$\mathrm{maxRel}^*_{\lambda,\bullet}(p) : \max_{\substack{\boldsymbol{w},\hat{\boldsymbol{w}},\boldsymbol{w}^*_\chi,\widehat{\boldsymbol{w}^*_\chi},\boldsymbol{w}^*_\xi,\widehat{\boldsymbol{w}^*_\xi},\\ \boldsymbol{b},d_\chi,d_\xi}} \hat{\boldsymbol{w}}^*_{\bullet p}$$

s.t. for all i, j *Equation* (4.9) *holds and*

$$\begin{aligned}
&\frac{1}{2}\sum_k \hat{\boldsymbol{w}}_k + \frac{\gamma}{2}\sum_k \hat{\boldsymbol{w}}^*_{\chi k} + \frac{\gamma}{2}\sum_k \hat{\boldsymbol{w}}^*_{\xi k} + \mathcal{L} \le (1+\delta)\cdot \mu_X \\
&\boldsymbol{w}_i \le \hat{\boldsymbol{w}}_i,\ -\boldsymbol{w}_i \le \hat{\boldsymbol{w}}_i \\
&\chi_i \le \hat{\chi}_i,\ -\chi_i \le \hat{\chi}_i \\
&\xi_i \le \hat{\xi}_i,\ -\xi_i \le \hat{\xi}_i \\
&\hat{w}^*_{\bullet p} \le \lambda \cdot w^*_{\bullet p}\ .
\end{aligned}$$

Then

$$\mathrm{maxRel}(p) := \max_{\substack{\lambda\in\{-1,+1\},\\ \bullet\in\{\chi,\xi\}}} \mathrm{maxRel}^*_{\lambda,\bullet}(p),$$

is the maximum of four linear problems.

The proof for this is analogous to Appendix A.1.5.

To improve the stability of feature selection we utilize the same procedure already proposed in Section 3.2.1.

Time complexity

In the following, we outline the scaling behaviour of our proposed method for feature selection. Our method can be divided into three separate computational steps which differ in their algorithmic complexity. We consider a problem with n samples and d features.

The initial baseline solution is analogue to a standard ordinal regression SVM solution which can be solved using the sequential minimal optimization (SMO) algorithm [Pla98; CK07] which is in $\mathcal{O}(n^3)$. The relevance bounds are given by a set of LPs for which interior point methods exist [Kar84; Vai89; CLS19] which are in $\mathcal{O}(n^{2.5})$. This complexity bound is very general and one could reformulate and adapt these problems using existing outlines [Joa06; Hsi+08]. In the normal setting, we consider the constant $z = 3$ for the number of LPs needed (Section 4.1.2) and $z = 6$ in the LUPI setting (Section 4.2.2) such that the relevance interval for each feature is in $\mathcal{O}(zn^{2.5})$. This results in $\mathcal{O}(dzn^{2.5})$ for all relevance bounds. Additionally, we employ a permutation test approach which adds a constant c additional LPs to achieve statistical stability which is overall in $\mathcal{O}(cn^{2.5})$. Overall our method is in $\mathcal{O}(n^3 + (dz + c)n^{2.5})$ when considering $n > d$.

Because the $dz + c$ LPs are a significant factor, we propose to solve them in parallel, which we evaluate in Appendix A.1.6.

4.2.3 *Evaluation*

The following section evaluates our approach for the LUPI paradigm, i. e. our method handling privileged information, that we denote FRI^*. From here, we focus on the explicit variant, after showing its superiority over the implicit version in Section 4.1.3 as regards computational complexity, leading to the notation FRI_e^*. Again, we show the quality of our feature selection by testing on artificially created data with known ground truth. Due to a lack of specific LUPI benchmark datasets, we conclude this section with a semantic analysis of a FRI_e^* model on one demonstrative example.

Artificial Data

We use the generation method presented in [Lop+16] to create artificial datasets containing regular as well as privileged information by sampling triplets (x_i, x_i^*, y_i) from

$$\begin{aligned} x_i^* &\sim \mathcal{N}(0, I_d) \\ \epsilon_i &\sim \mathcal{N}(0, I_d) \\ x_i &\leftarrow x_i^* + \epsilon \\ y_i &\leftarrow f(\langle \omega, x_i^* \rangle) \end{aligned}$$

where f denotes a function that assigns the correct ordinal bin to the label y_i based on the value of the dot product between the weight vector and a privileged sample x_i^*.

Hereby, the privileged information X^* consists of clean versions of the noisy regular features X. Both the regular and the privileged feature space contain strong, weak and irrelevant features. These are created in the same way as described in Section 4.1.3. The characteristics of the data used in our experiments are shown in Table 4.8. The last two sets differ from the generation method mentioned above. Their regular information is created similarly to the sets in Table 4.1, to which three irrelevant privileged features are added from random Gaussian noise. All features are normalized to zero mean and unit variance.

Table 4.8: Artificially created data with regular and privileged features under known ground truth. For the first six sets, the privileged features consist of clean versions of the regular information. The last two sets are regular ordinal regression sets with random noise as additional privileged information.

		Regular Features			Privileged Features		
Dataset	*#Instances*	*#Str*	*#Weak*	*#Irr*	*#Str*	*#Weak*	*#Irr*
Set 1	200	6	0	3	6	0	3
Set 2	200	0	12	3	0	12	3
Set 3	200	6	6	0	6	6	0
Set 4	200	3	6	0	3	6	0
Set 5	200	1	4	0	1	4	0
Set 6	200	1	40	10	1	40	10
Set 7	200	4	2	2	0	0	3
Set 8	200	0	4	2	0	0	3

Evaluation closely follows Section 4.1.3. Again, we use the F_1-measure as a quantifying metric for the detection of the all-relevant feature set and compare our method to the EN surrogate model $\mathrm{M}_e^{L_1+L_2}$. While FRI_e^* differentiates between the two feature spaces in the data, the EN receives both the regular and the privileged set as one. With that, we want to showcase the advantages of a LUPI model for feature selection over a purely regular model.

The results are given in Table 4.9. FRI_e^* achieves a perfect score on the regular feature set and only stumbles once, for set 6, on the privileged information. The EN, on the other hand, performs considerably worse on the regular set but shows significant improvements on the privileged set, albeit it cannot match the performance of our method. The improvements on the privileged data are easy to explain since this information is the clear original information as opposed to the noisy features in the regular set.

Table 4.9: Artificially created datasets with known ground truth and evaluation of the identified relevant features by the methods as compared to all existing relevant features. The EN surrogate model ($\mathrm{M}_e^{L_1+L_2}$) receives both feature sets as one but the evaluation is done separately for the regular and privileged feature set. The score is averaged over 10 independent runs.

		Regular Features		Privileged Features	
score	data	$\mathrm{M}_e^{L_1+L_2}$	FRI_e^*	$\mathrm{M}_e^{L_1+L_2}$	FRI_e^*
F_1	Set 1	0.44	1.0	0.89	1.0
	Set 2	0.48	1.0	0.85	1.0
	Set 3	0.65	1.0	0.91	1.0
	Set 4	0.58	1.0	0.88	1.0
	Set 5	0.67	1.0	0.92	1.0
	Set 6	0.40	1.0	0.69	0.99
	Set 7	0.93	1.0	1.0	1.0
	Set 8	0.70	1.0	1.0	1.0
Precision	Set 1	0.72	1.0	0.91	1.0
	Set 2	0.75	1.0	0.98	1.0
	Set 3	1.0	1.0	1.0	1.0
	Set 4	0.90	1.0	1.0	1.0
	Set 5	0.80	1.0	1.0	1.0
	Set 6	0.98	1.0	0.97	1.0
	Set 7	0.94	1.0	1.0	1.0
	Set 8	1.0	1.0	1.0	1.0
Recall	Set 1	0.37	1.0	0.88	1.0
	Set 2	0.38	1.0	0.78	1.0
	Set 3	0.52	1.0	0.84	1.0
	Set 4	0.48	1.0	0.80	1.0
	Set 5	0.62	1.0	0.88	1.0
	Set 6	0.26	0.99	0.54	0.98
	Set 7	0.93	1.0	1.0	1.0
	Set 8	0.55	1.0	1.0	1.0

Semantic Analysis

Performing evaluations similar to Section 4.1.3 on real data is not possible because of the lack of public LUPI benchmarks. Alternatively, we consider one illustrative example to demonstrate the semantic implications of the FRI framework for LUPI. We generate a set with 400 samples and six features. Initially, there are three strongly relevant features and three irrelevant ones drawn from random Gaussian noise. We divide the samples into four groups, each with 100 members. The first group has Gaussian noise with a standard deviation of 0.1 added to the first strongly relevant feature. The second group has a noise level of 0.5 added to the second feature. Similarly, the third one has Gaussian noise on the last strong feature with a standard deviation of 2. The data in the last group is noise-free. The idea is to provide the insight which samples of the dataset are hard to classify as privileged information to the model. Therefore, the privileged set consists of three features, incorporating the noise that was added to the groups, with the first privileged feature corresponding to the first group and so on.

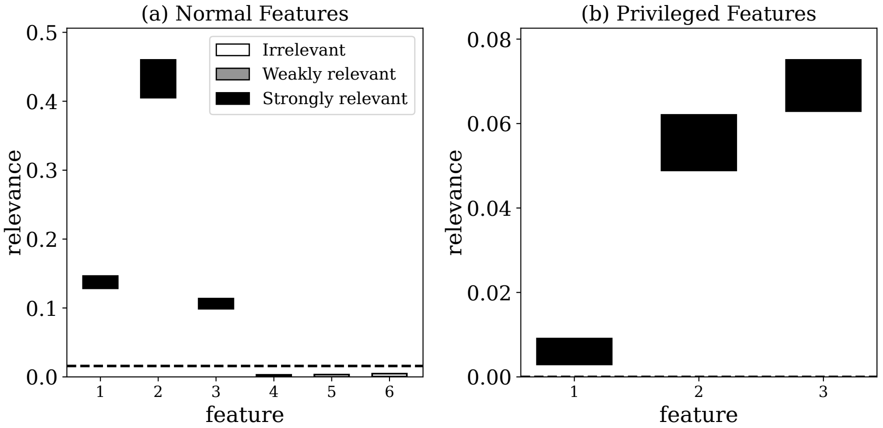

Figure 4.2: Relevance plots for the semantic analysis. (a) Relevance of the regular features for the LUPI model. (b): Relevance of the privileged features for the LUPI model.

The plots in Figure 4.2 show the relevancy for the (a) regular features and (b) privileged features. Our method correctly dismisses the three irrelevant features and also classifies all strongly relevant features. More importantly, all privileged features were also correctly classified, and their relevance correlates with the noise level. With that, we show that FRI_e^* can discriminate between the usefulness of multiple privileged features and utilize those that are necessary for this setting.

4.3 CONCLUSIONS

In this chapter, we presented the adaption of the feature relevance bounds approach to ordinal regression data using the *explicit* order variant. The optimization problem was phrased by approximating the generalization ability of the model with a bound on the L_1-margin. The resulting problem can be transferred to a linear problem. For its solution, we used another approximation by splitting the objective into the margin and slack variables separately, for larger robustness. Further, we applied our resampling-based method from Chapter 3 to allow precise feature selection. Based on the experiments we showed that the *explicit* variant is comparable to the *implicit* variant for this use case on the given data as regards the accuracy and more efficient. Our method can provide a near-perfect all-relevant feature set approximation while being significantly faster than the other variant. Although not many feature selection approaches exist for that specific context we could also showcase the feature selection performance in comparison with another popular approach on toy and real data. The feature sets produced by our approach represents additional information useful in analytic use cases for model and experiment design, subject for further evaluation, and it constitutes a possible starting point to investigate, e. g. the information which restricted or protected features can provide for the class of linear ORP models.

Furthermore, we also defined feature relevance bounds when additional information is present in the context of learning using privileged information. Here we defined a feature's relevance in relation to the training phase itself. Similar to the classical context, our method achieved very good feature selection sensitivity in both the regular and privileged feature set, this way enabling a strategy to choose suitable features or teacher information to facilitate training.

5 NON-LINEAR FEATURE SELECTION AND CLASSIFICATION

Parts of this chapter are based on:

- Lukas Pfannschmidt and Barbara Hammer. "Sequential Feature Classification in the Context of Redundancies". In: Apr. 15, 2020. arXiv: `2004.00658`. URL: `http://arxiv.org/abs/2004.00658`

5.1 BACKGROUND

Context	
Problem	*classification* *regression*
Model	*non-linear*
Type	*classical*

In Section 2.2 we described several types of feature subsets given a specific class of functions. These are the set of strongly relevant features ($\mathcal{S}$), the set of weakly relevant features ($\mathcal{W}$) and irrelevant features ($\mathcal{I}$). The union of $\mathcal{S}$ and $\mathcal{W}$ is the set of all-relevant features $\mathcal{A} := \mathcal{S} \cup \mathcal{W}$. The goal of all relevant feature selection is finding all features belonging to $\mathcal{A}$, and not to identify the detailed composition of $\mathcal{S}$ and $\mathcal{W}$. Most existing methods only produce information about membership of $\mathcal{A}$, except the relevance bounds method described in Section 2.3.2, which yields both $\mathcal{S}$ and $\mathcal{W}$ in the linear case. For the non-linear case, there exists no approach which can make this distinction. Knowing about a feature's type can not only improve understanding of causal factors but also help in biomarker design, where robust feature sets are needed [HY10].

One existing all-relevant selection method is called Boruta [KR10]. It builds on the information metrics acquired by observing the single trees of an RF model. Because the scores are not consistent with their real significance [Rud+06], Boruta employs an extended information system. Extended in the way, that additional to normal features Boruta adds shadow features. Shadow features are randomly shuffled clones of existing features to remove correlation with the target. Through the addition of those, one can estimate a contrast distribution of features, which are by design irrelevant. It then tests the real features against the shadow features iteratively, increasing the significance threshold, until all features are tested conclusively. Boruta's comparison with a null distribution inspired one aspect of our proposed method, which we come back to later. More importantly, by comparing with a null distribution multiple times and introducing stochastic noise by repeatedly running an RF model, Boruta can identify the $\mathcal{A}$ set with high precision.

In this chapter, we present a method which extends the Boruta method with a feature classification step which produces the same discrimination between strong and weak relevance as seen in Chapter 3 for linear models and answer research question 2 from Section 1.2.

We combine the advantages of Boruta and FRI while achieving more efficient run times. Decisions about the feature relevance can be made based on the accuracy of the model when excluding a single feature in question. In Section 5.2.1 we describe our approach of efficiently decomposing the overall feature set into subsets with different characteristics to produce the distinction between strong and weak relevant features without testing all available features.

We improve existing methods to find set $\mathcal{M}$ by using statistical based-thresholds and introduce a robust score testing scheme in Section 5.2.2 to improve feature classification.

In Section 5.3 we analyse the general feature selection performance against other established approaches in the context of redundant features and answer research question 4.

5.2 METHODS

5.2.1 *Loss-based Feature Set Decomposition*

Let X be data set $X := \left\{x_i \in \mathbb{R}^d; i = 1, \ldots, n\right\}$ with n samples and with $\mathcal{D} := \{\ell \in \mathbb{Z}; \ell = 1, \ldots, d\}$ as the set of all features such that cardinality $|\mathcal{D}| = d$. Target variable $y \in \mathbb{R}^n$ is distributed according to some unknown function dependent on X such that $g(X) = y$. Without loss of generality, we limit y to be continuous and g to be a regression function. Let $f^\star := \hat{g}$ be the optimal estimator of g with minimal estimation loss. Consider the estimation loss as

$$\mathcal{L}(X, f) := \sum_{\forall i} |f(x_i) - y_i|$$

where $x \in X$ which denotes the deviation from g if no random noise is involved.

We are interested in all functions, including non-linear ones, with similar $\mathcal{L}$ and possibly different composition of input features:

$$\mathcal{F}^\star := \{f \in \mathcal{F} \mid \mathcal{L}(X, f) \approx \mathcal{L}(X, f^\star)\}$$

Before, we observed the set of data X with all features in $\mathcal{D}$, i. e.

$$X = X_{\mathcal{D}}.$$

Now we also consider the data set with specific subsets of features, e.g. the dataset with feature ℓ removed is denoted as

$$X_{\mathcal{D} \setminus \ell}.$$

Having found the set of all relevant features $\mathcal{A}$ for function class $\mathcal{F}^\star$, one could trivially classify all features in $\mathcal{D}$ which are not in $\mathcal{A}$, as irrelevant:

$$\mathcal{I} := \mathcal{D} \setminus \mathcal{A}.$$

To decompose $\mathcal{A}$ into $\mathcal{S}$ and $\mathcal{W}$ we have to identify membership with at least one of them. We can check for strong relevance by repeatedly fitting models and checking the behaviour of the loss function similar to the approach in [Nil+07]. A feature ℓ is strongly relevant if

$$\min_f \mathcal{L}(X_{\mathcal{D} \setminus \ell}, f) > \min_h \mathcal{L}(X, h). \tag{5.1}$$

This comparison would have to be performed for all $\ell \in \mathcal{A}$. While $|\mathcal{A}| \ll |\mathcal{D}|$ in most cases, i. e. the feature space is often sparse and most of the features can be ignored, we still can improve on this naive approach.

Earlier, we considered the specific problem of all-relevant feature selection. In general when feature selection is performed, one considers the *minimal-optimal* feature set ($\mathcal{M}$). In the following, we are using a new efficient importance value threshold approach to find $\mathcal{M}$. Importance values are internal parameters which correspond to input features similar to the weights of linear models and schemes such as Lasso. Because we utilize Boruta, we consider the importance values of an RF model. In Section 5.2.3

we describe in detail how we find $\mathcal{M}$ with a sparse parameterization and statistical method to overcome inconsistent importance values. For now, we consider $\mathcal{M}$ as given by this efficient approach, but alternatives such as RFE could also be applied.

By definition, $\mathcal{M} \subset \mathcal{A}$ and $\mathcal{S} \subset \mathcal{M}$. All features in the minimal optimal set $\mathcal{M}$ are also included in $\mathcal{A}$ and furthermore, all strongly relevant features are included in $\mathcal{M}$. Instead of iterating and testing all features in $\mathcal{A}$ we only have to consider the features in $\mathcal{M}$. In most cases when redundant relevant features are preset, i. e. $|\mathcal{M}| \ll |\mathcal{A}|$, it is much more efficient to only check features in the subset $\mathcal{M}$.

Therefore, it is sufficient to identify $\mathcal{S}$ in $\mathcal{M}$ through comparing the loss after the elimination of each feature and identify $\mathcal{W}$ through

$$\mathcal{W} := \mathcal{A} \setminus \mathcal{S}. \tag{5.2}$$

Having said that, comparison of the loss is not straightforward as it requires a robust threshold to test against.

5.2.2 *Robust Loss Comparison*

In practice, we cannot perform a simple comparison between a reduced (set with feature ℓ removed) and a normal feature set. Due to the stochastic nature of RFs, even fitting the same data set without feature removal can lead to variable models and thus to differing average losses. In the following, we assume and estimate a normal distribution of $\mathcal{L}$. The distribution should represent likely values which we regard as insignificant changes, similar to how our model would change if an irrelevant feature would be removed. Then, we test for the deviation from this using a predictive interval test as described in [Gei93, Chapter 2]. This approach is similar to the one from Section 3.2.1, where we applied this for feature relevancies.

Remember that we fit an RF model on a reduced dataset where one feature is eliminated and then observe the loss of the resulting model. The distribution should emulate this given setting and as such we would have to eliminate one feature in our sampling procedure as well. However, we cannot remove any feature from the initial feature set without the possibility of removing another relevant feature by chance. To emulate the changes in model size we, therefore, permute a randomly chosen feature (via the uniform distribution $\mathcal{U}$) and add it to the dataset. We define $\operatorname{perm}(X_\ell)$ as the random permutation of values in X_ℓ and

$$X \diamond \ell_k := \begin{cases} X, & \text{if } k \neq \ell \\ \operatorname{perm}(X_\ell), & \text{otherwise} \end{cases}$$

as the dataset where only ℓ was replaced by its random permutation. A feature created by $\operatorname{perm}(X_\ell)$ has no dependence on the target variable and represents an irrelevant feature.

We then define the random population

$$\widehat{\pi}(X, f^\star, \alpha) := \big(\mathcal{L}(X \diamond \ell, f^\star)_i\big)_{i \in \{1,\dots,\alpha\}} \quad \ell \sim \mathcal{U}(1, d) \tag{5.3}$$

with parameter $\alpha \in \mathbb{Z}$ as the number of samples i used in the following shortened to $\widehat{\pi}_{(\alpha)}$.

We define an interval of plausible values, based on a normal distribution and the likely deviation from the mean with a t-distribution, as

$$\Pi(\alpha,\tau)_{\max} := \overline{\widehat{\pi}_{(\alpha)}} + \mathcal{T}_{\alpha-1}(\tau)\cdot\sigma(\widehat{\pi}_{(\alpha)})\sqrt{1+(1/\alpha)} \tag{5.4}$$

and

$$\Pi(\alpha,\tau)_{\min} := \overline{\widehat{\pi}_{(\alpha)}} - \mathcal{T}_{\alpha-1}(\tau)\cdot\sigma(\widehat{\pi}_{(\alpha)})\sqrt{1+(1/\alpha)}. \tag{5.5}$$

Here $\overline{\pi}_\alpha$ denotes the sample mean and $\sigma(\cdot)$ the standard deviation, and $\mathcal{T}$ represents Student's t-distribution with $\alpha - 1$ degrees of freedom. Together these two values define the interval

$$\Pi := \Big[\Pi(\alpha,\tau)_{\min},\ \Pi(\alpha,\tau)_{\max}\Big]. \tag{5.6}$$

The size of Π depends on the parameter $\tau \ll 1$ which represents the percentile of the distribution which is used to reject values, and the number of samples α. In our experiments $\alpha \geq 50$ yields robust thresholds for common feature set sizes. The interval only has to be computed once and is valid for all feature comparisons as it represents the distribution of irrelevant features and not a feature specific one.

With this interval, we can now make robust comparisons for individual features. Feature ℓ is strongly relevant if

$$\mathcal{L}(X_{\mathcal{D}\setminus\ell}, f^\star) < \Pi$$

which means we only have to check the lower bound *min* of the prediction interval.

Using this procedure and checking all features in $\mathcal{M}$ leads to the complete set of strongly relevant features $\mathcal{S}$ and therefore also $\mathcal{W}$ as seen in Equation (5.2).

5.2.3 *Applications of Random Forest Importance Values*

Deep learning models with many hidden layers can be opaque in their attribution of the input features in relation to the output layer. For Random Forest (RF) models exist several measures of feature importance. They commonly express a feature's importance by averaging an information measure over all splits in the decision forest, which were part of the ensemble. Examples are the average information gain of the objective function or the number of correct classifications with and without the feature. In the following let's consider the information gain measure as

$$\operatorname{imp}(X, f^\star) \in \mathbb{R}^d. \tag{5.7}$$

The improvement of the splitting criterion averaged over all trees and splits where feature ℓ is used as the split feature is then $\operatorname{imp}(X, f^\star)_\ell$.

If an importance measure correlates to the relevance of the input feature, we could use it in deciding which features are relevant. Also, we would expect correlated or identical features to exhibit the same feature importance. In practice, some features are implicitly preferred because of small differences in information content or pure stochastic reasons. We demonstrate this in Figure 5.3 where some features have much bigger importance values and others only have small or zero importance. This is an example of correlation bias [TL11] which is a common problem for many importance measures in general.

Algorithm 1: Estimating stochastic bounds for loss and irrelevant feature importance

```
Data: X, y
Input: Model, $N_{\text{Samples}}$
$\widehat{\pi} \leftarrow \varnothing$ (empty set)
$\widehat{\gamma} \leftarrow \varnothing$
s ← 0
while $s < N_{\text{Samples}}$; $s$++ do
    s ← generatePermFeature(X)
    $X \diamond \ell \leftarrow$ X ∪ s

    // Fit model
    ext-model ← Model($Xs$, $y$)

    // loss samples Section 5.2.2
    loss ← loss($ext\text{-}model$, $X \diamond \ell$, y)
    // importance samples Section 5.2.3
    irrel_imp ← importance (ext-model, s)

    $\widehat{\pi} \leftarrow \widehat{\pi} \cup$ loss
    $\widehat{\gamma} \leftarrow \widehat{\gamma} \cup$ irrel_imp
end
// Equation (5.6)
LossBounds ← t-statistic($\widehat{\pi}$)
IrrelBounds ← t-statistic($\widehat{\gamma}$)
Output: LossBounds, IrrelBounds
```

An important parameter in fitting an RF is the *feature fraction* which denotes how many features are included for each tree bootstrap. When this fraction is high (≈ 100%), most features are included. To circumvent correlation bias, we use a feature fraction of only 10% in our experiments which yields a more even distribution of importances and is close to the recommended optimum of $\sqrt{d}$ [DA06] as is demonstrated in Figure 5.2. While being more homogeneous, a decrease in feature fraction leads to higher variance. In the next two sections, we utilize statistical distributions to make the applications of feature importance values for RF robust.

Minimal Feature Set

As mentioned in Section 5.2.1, we do not use an existing minimal feature selection method to decide which features are part of $\mathcal{M}$. Instead, we propose to use feature importance values from the RF model to efficiently decide which features are relevant. In contrast to Boruta, we are interested in a sparse solution of the feature set without redundant features, which represents $\mathcal{M}$.

Similar to linear models using a sparse regularization, we could force redundant features to exhibit low importance in the model. Lasso uses L1 regularization which leads to many zero entries in the model's weight-vector and features with non-zero weights are considered part of the feature set. By parameterizing the RF to have a high *feature fraction* we force a similar sparsity as can be seen on page 72 in Figure 5.4. There, the majority of features (0–14) are relevant to the target but it's apparent in the figure that

Algorithm 2: Iterative Decomposition Algorithm

```
**Data:** X,y
**Input:** RF
$\mathcal{S} \leftarrow \varnothing$
$\mathcal{W} \leftarrow \varnothing$
$\mathcal{A} \leftarrow$ Boruta(RF, $X$, $y$)
$\mathcal{M} \leftarrow$ ImpSelection(RF, $X$, $y$) (Equation (5.12))

// Reduce dataset X to relevant features
$\mathcal{V} \leftarrow$ select(X, $\mathcal{A}$)

// loss bounds using Algorithm 1
$\Pi \leftarrow$ lossBounds(RF, $\mathcal{V}$, y)

// iterate over subset M
for feature $\ell$ in $\mathcal{M}$ do
    // Remove current feature
    $\mathcal{I} \leftarrow \mathcal{V}$ without feature $\ell$
    // find best model without $\ell$
    reduced_model $\leftarrow$ fit(RF, $\mathcal{I}$, y)
    // compute score
    $loss_j \leftarrow$ loss(reduced_model, $\mathcal{I}$, y)
    // add current feature to $\mathcal{S}$ if significantly worse
    if $loss_j$ not in $\Pi$ then
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{\ell\}$
    end
end
// decomposition(Equation (5.2))
$\mathcal{W} \leftarrow \mathcal{A} \setminus \mathcal{S}$
**Output:** $\mathcal{S}$, $\mathcal{W}$
```

even irrelevant features do not have an importance value equal to zero and a fixed threshold at zero would lead to noise. Thus, important for this approach is a well-defined threshold to decide which value is considered irrelevant. A simple measure like the mean of all importance values (blue horizontal line in the figure) does not work in general. If we could characterize the behaviour of importance values for irrelevant features with a statistical distribution, we could use it to set the threshold depending on the parameters.

In Section 3.2.1 we proposed a statistics-based approach to estimate a dynamic feature threshold for the linear models. We used randomly permuted real features to simulate irrelevant variables and observe their relevancies in the model over many samples. In Section 5.2.2 we used a similar approach to test for insignificant loss changes in the RF model. We now extend the statistics from Section 5.2.2 to include the importance values of randomly permuted features. Again, we assume that the importance values of irrelevant features follow a normal distribution given many samples.

In Equation (5.3) we already used randomly permuted features for the loss distribution, where we already fit models on the data and feature importance values can be recorded. Therefore, in practice, we can reuse the same sample population for efficient computation. For the distribution in Section 5.2.2, we focused on the score, whereby here we focus on the feature importance values, i. e. we use the same models to create the distributions. For clarity, we define the importance value statistics explicitly as a separate sample population.

The samples for the distributions are defined as

$$\widehat{\gamma}_s(X, f^\star, \alpha) := \Big(\text{imp}(X \diamond \ell, f^\star, \ell)_i\Big)_{i \in \{1,\dots,\alpha\}} \qquad \ell \sim \mathcal{U}(1, d) \tag{5.8}$$

where $f^\star$ represents the optimal model on $X \diamond \ell$, $\text{imp}(X \diamond \ell, f^\star, \ell)$ represents the importance value of feature ℓ in $f^\star$, and parameter $\alpha \in \mathbb{Z}$ is the number of samples used. Each sample i consists only of the importance of a single permuted feature ℓ in the model $f^\star$. In the following, we shorten all samples to $\widehat{\gamma}_{s(\alpha)}$. The bounds are then defined as

$$\Gamma_s(\alpha, \tau)_{\max} := \overline{\widehat{\gamma}_{s(\alpha)}} + \mathcal{T}_{\alpha-1}(\tau) \cdot \sigma(\widehat{\gamma}_{s(\alpha)})\sqrt{1 + (1/\alpha)} \tag{5.9}$$

and

$$\Gamma_s(\alpha, \tau)_{\min} := \overline{\widehat{\gamma}_{s(\alpha)}} - \mathcal{T}_{\alpha-1}(\tau) \cdot \sigma(\widehat{\gamma}_{s(\alpha)})\sqrt{1 + (1/\alpha)}. \tag{5.10}$$

Here $\overline{\widehat{\gamma}_{s(\alpha)}}$ denotes the sample mean and $\sigma(\widehat{\gamma}_{s(\alpha)})$ the standard deviation, and $\mathcal{T}$ represents Student's t-distribution with $\alpha - 1$ degrees of freedom. Together these two values define the interval

$$\Gamma_s := \Big[\Gamma_s(\alpha, \tau)_{\min}, \ \Gamma_s(\alpha, \tau)_{\max}\Big]. \tag{5.11}$$

To produce the minimal-optimal feature set $\mathcal{M}$ we fit a random forest with a high allowed feature fraction. This leads to the behaviour seen in Figure 5.3 where only a subset of correlated features shows significant importance. We then compare each feature's importance value with Γ_s which represents the distribution of importance values we consider as irrelevant. The minimal-optimal set is then given as

$$\mathcal{M} := \{\text{imp}(X, f^\star)_\ell > \Gamma_s \mid \forall \ell \in \mathcal{A}\}. \tag{5.12}$$

Figure 5.1 shows the upper bound (5.9) of the interval in use with an RF model on toy data.

5.3 RESULTS

5.3.1 *Implementation*

For the experiments, we implemented the algorithms in Section 5.2 in Python utilizing several existing libraries. The methods such as Boruta or our importance selection method are wrappers which require the definition of an inner model. Because we are fitting this model many times, we opted for a very efficient RF implementation, which is provided by the *LightGBM* library [Ke+17].

The Boruta [KR10] method is implemented in the *boruta_py* library[1]. Other utility functions are used from *scikit-learn* [Ped+11]. The complete implementation (nicknamed *'Squamish'*) and source code is publicly available.[2]

[1] https://github.com/scikit-learn-contrib/boruta_py

[2] https://github.com/lpfann/squamish

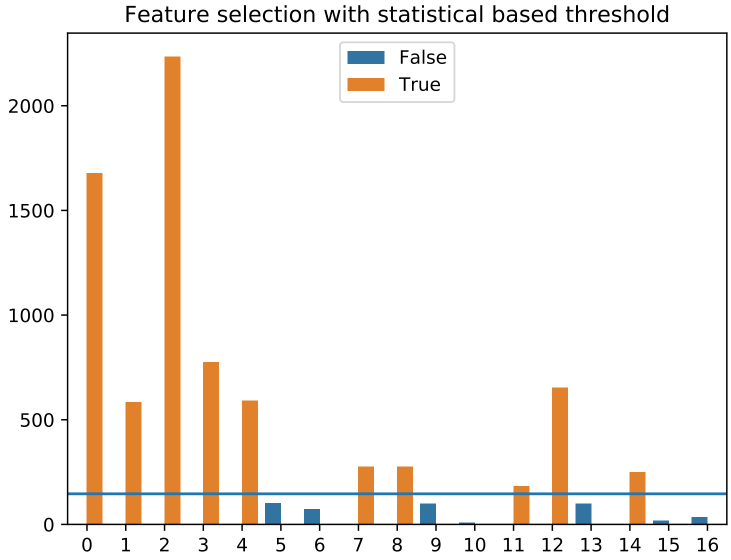

Figure 5.1: Importance values per feature (bars) of an RF model and upper feature selection threshold of Γ_s (5.9). The lower threshold as defined in (5.10) is below zero and excluded in this figure. Colours denote the membership of features to $\mathcal{M}$.

5.3.2 *Benchmark Models*

To show the characteristics of our methods we ran several benchmarks against established feature selection methods. Here we specifically focus on the performance of the all-relevant feature selection in a setting where redundant features are present.

We compare the following approaches

EN A linear model using the EN method which combines both L_2 and L_1 regularization. The combination can be weighted linearly and allows for more sensitivity of redundant features. In our experiments, we fit this parameter through grid search in combination with cross-validation. Features are selected according to RFE which is guided by a cross-validated model performance [ZH05].

RF An RF model (*LightGBM*) with RFE as the selection method where the number of features is decided by cross-validation [Ke+17].

FRI The feature relevance interval method [PJ20; Pfa+19a] as the only representative with a distinction between strong and weak relevance.

The sequential feature class decomposition method presented in Algorithm 2 and implemented as described in Section 5.3.1. Parameters for the statistical test where $\alpha = 50$ and $\tau = 10^{-6}$

Hyper-parameters for all methods were decided by cross-validation. The tree models are based on *LightGBM* and are using the default parameters except:

`num_leaves` = 32

`max_depth` = 5

`boosting_type` = rf

`bagging_fraction` = 0.632

`bagging_freq` = 1

The `feature_fraction` parameter was left to default = 1 for the RF method. In SQ for the loss comparison, it was set to 0.8. SQ utilizes Boruta which in turn also encapsulates another tree model. This inner tree model was set to `feature_fraction`= 0.1.

All scripts and data generation methods are publicly available and results can be reproduced.[3]

5.3.3 *Stability of Feature Importance Values*

The proposed method is utilizing RF importance values to decide about the relevance of features. In the following, we perform a short analysis of the variance of these importance values over multiple model fits on the same dataset. We employ an RF model as described in the section before with two different parameter choices. The first choice is a low `feature_fraction` = 0.1 and the second the maximum `feature_fraction` = 1 such that all features are allowed in each tree generation.

We generate a simple linear classification dataset with 17 features and 300 samples. Features 0-4 are considered strongly relevant, features 5-14 are weakly relevant with in-between correlations, and features 15-16 are irrelevant.

The test consisted of fitting the model on the dataset 10 times in a row and record the feature importance gain measure as defined in Section 5.2.3. The resulting distributions are visualized in Figure 5.4 for each parameter choice. One can see, that even without any variation of the data, the values show high variance in both settings and do not correlate to the real mutual information with the target variable. Furthermore, the choice of a high feature fraction leads to some variables overshadowing the importance of others such as seen in feature 12 which contains the same information as all other weakly relevant in this case. A lower fraction leads to a more evenly distributed importance signature.

5.3.4 *Parameterization for Feature Selection*

Based on the evaluation in Section 5.3.3 we also perform further analysis on the consequences of the parameter for feature selection. We extend the experiment with a feature selection step. Compared is RFE guided by cross-validation with Boruta.

We record the number of times each feature was selected in the feature set. This results in the frequency of selection or the probability that a feature is selected. The frequencies for `feature_fraction` = 0.1 are given in Figure 5.7 and for `feature_fraction` = 1 in Figure 5.8. We can see that a lower feature fraction is beneficial in the case of all-relevant feature selection where the Boruta model recognizes all relevant features 0–14 without selecting random features. The RFE procedure on the other hand suffers in this case and loses precision by selecting irrelevant features 50% of the time. It performs better with a high feature fraction parameter and selects strongly relevant features (0–4) consistently but shows a higher variance in the case of weakly relevant features.

[3] instructions available at `https://github.com/lpfann/squamish_experiments`

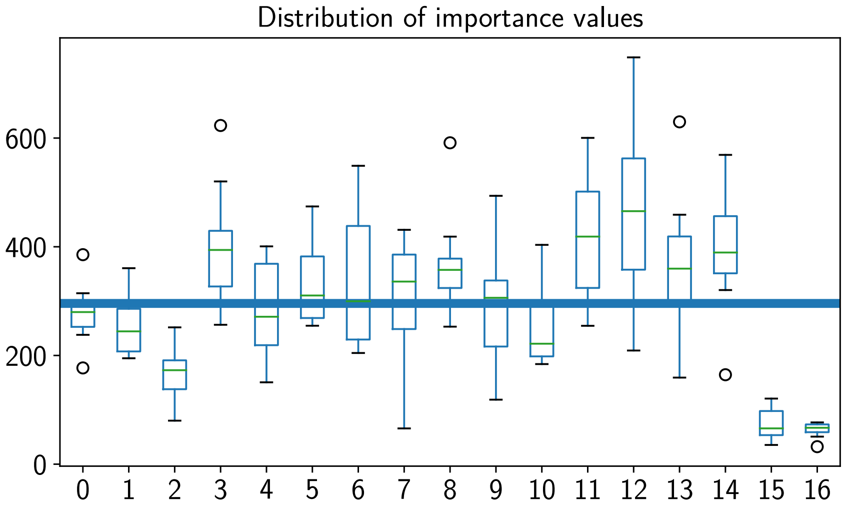

Figure 5.2: `feature_fraction` = 0.1

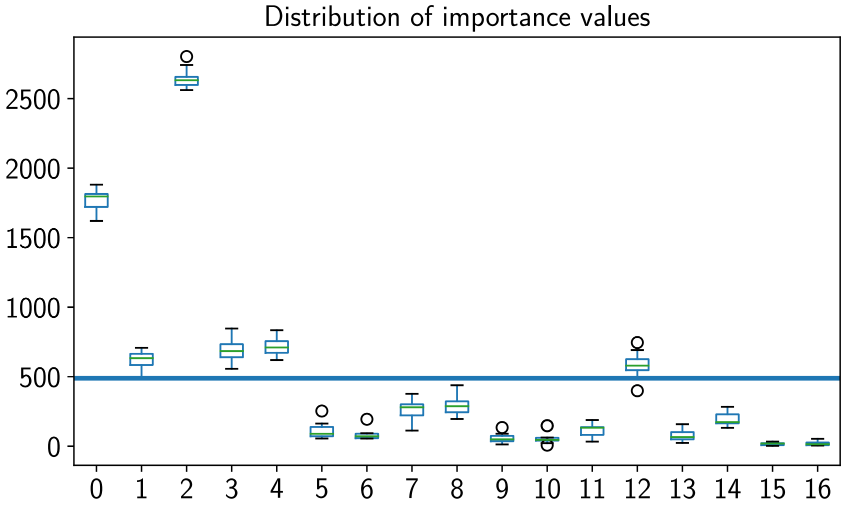

Figure 5.3: `feature_fraction` = 1.0

Figure 5.4: Distribution of feature gain importance values of RF classifier over multiple bootstrap iterations on toy example where features 0-14 are correlated and 15-16 are irrelevant. Mean of all importance values is given as blue horizontal line. Subplots 5.2 and 5.3 represent the different fraction of features allowed in tree construction.

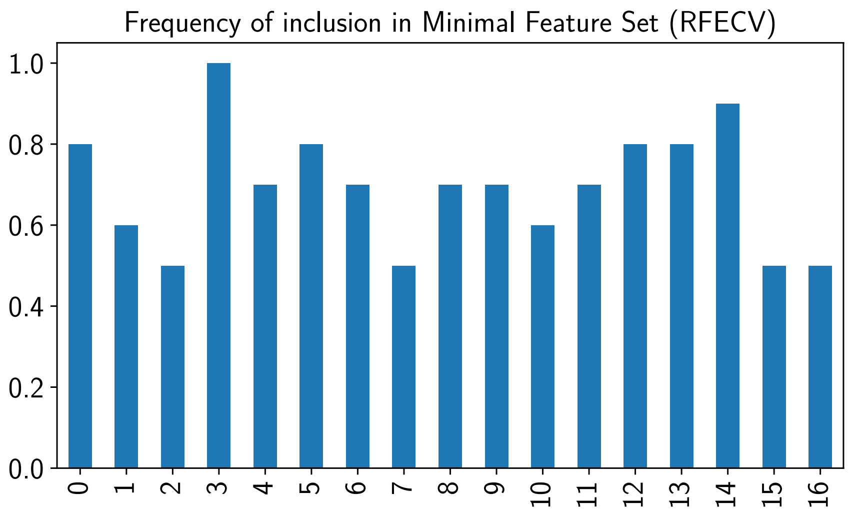

Figure 5.5: RFECV, `feature_fraction` = 0.1

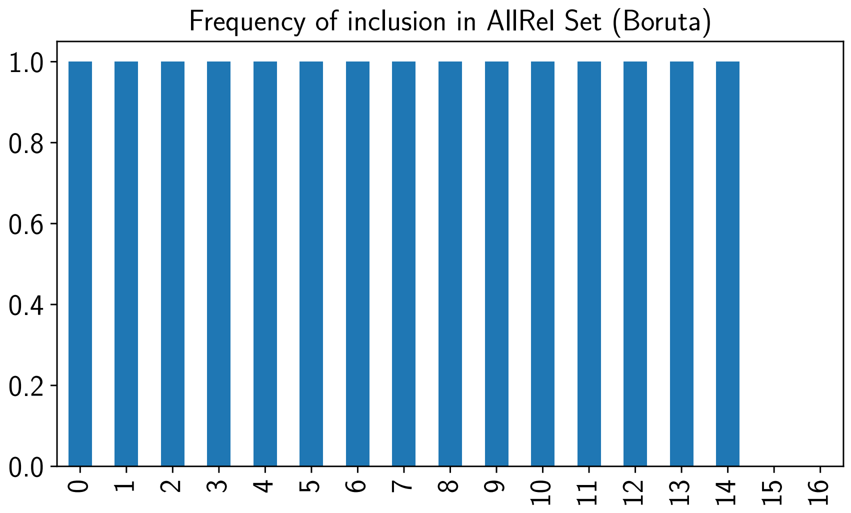

Figure 5.6: Boruta, `feature_fraction` = 0.1

Figure 5.7: Frequency of feature selection for a dataset with 5 strongly relevant features (0-4), 10 weakly relevant features (5-14) and 2 irrelevant features (15-16) as described in Section 5.3.3. Vertical bars represent the probability that each feature was included in the selected feature set for RFECV and Boruta. The RF model used a different setting for `feature_fraction`.

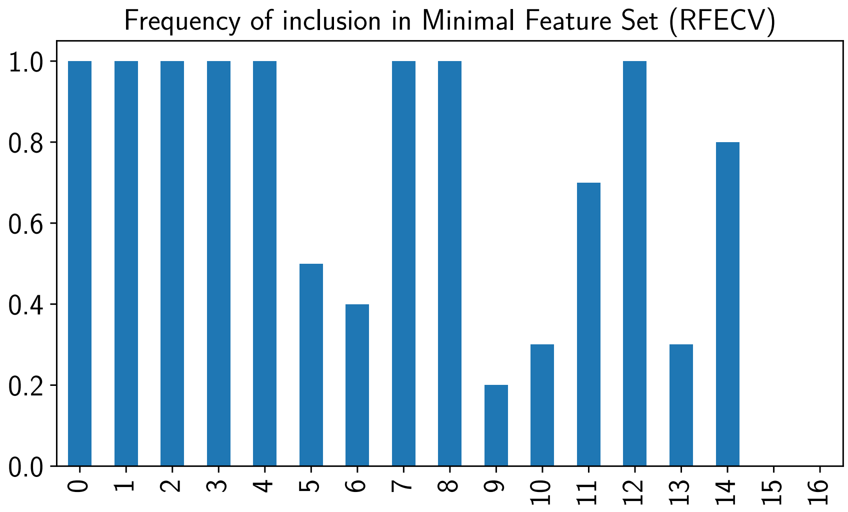

Figure: RFECV, `feature_fraction` $= 1$

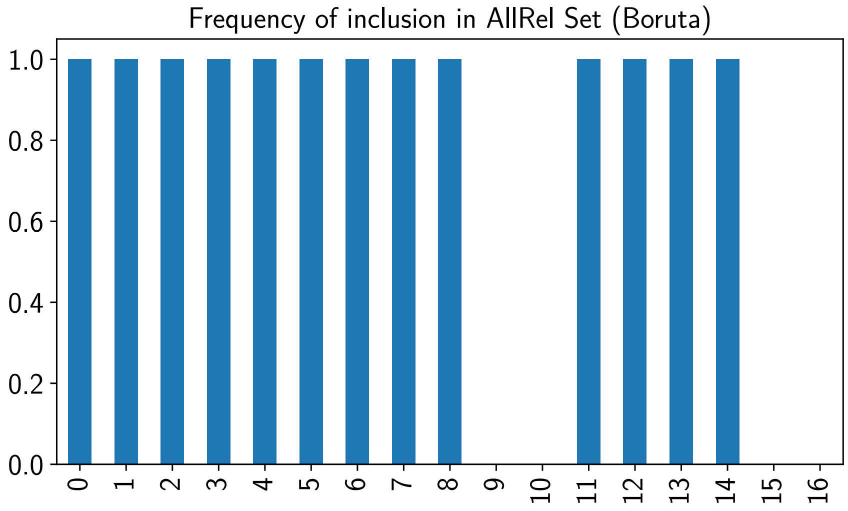

Figure: Boruta, `feature_fraction` $= 1$

Figure 5.8: Frequency of feature selection for a dataset with 5 strongly relevant features (0-4), 10 weakly relevant features (5-14) and 2 irrelevant features (15-16) as described in Section 5.3.3. Vertical bars represent the probability that each feature was included in the selected feature set for RFECV and Boruta. The RF model used a different setting for `feature_fraction`.

5.3.5 *Linear Feature Selection Accuracy*

The most relevant metric for feature selection methods is the accuracy of selected features. When the ground truth is known, we can explicitly evaluate the validity of the selected features. We focus on the all-relevant feature selection problem where we use the following measures to evaluate the match of the detected feature set and the known ground truth of all relevant features: precision and recall. The recall is defined by `TP / (TP+FN)` with `TP` = number of true positives and `FN` = number of false negatives. It denotes how many of the relevant features were selected which is crucial when looking for the all relevant feature set. Precision is defined by `TP / (TP+FP)` with `FP` = number of false positives and describes the frequency of false positives part of the feature set. One can use the F_1 measure as the harmonic mean of precision and recall:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{5.13}$$

In this evaluation, we compare the methods from Section 5.3.2 by highlighting the F_1 measure.

First, we compare all methods on a linear classification dataset to allow a fair comparison with the linear models. To generate a multidimensional classification problem, we use a randomly generated prototype vector which defines a hyperplane. The defining features of this plane are strongly relevant. Sample points are generated in this feature space and the class is determined by the side of the hyperplane the points lie on. Weakly relevant features are constructed by replacing a feature of the original feature space with its linear combination. The elements of this combination are highly correlated and produce a set of redundant features. By removing the original feature and replacing it with those elements we achieve weak relevance by definition. Irrelevant features are sampled from a standard normal distribution.

We generate 8 datasets with a different feature set composition as given by Table 5.1. For example, Set 1 consists of 150 samples (n), 6 strongly relevant features (strong), no weak relevant features (weak) and 6 irrelevant random features (irr).

All models given in Section 5.3.2 are repeatedly fit on bootstraps of these datasets, resulting in 10 results per dataset per model which are averaged in the following. The prediction accuracy on the datasets is listed in Table 5.2 with sufficient accuracy for all models.

The results of the F_1 measure evaluation are given in Table 5.3. Additionally, we recorded the runtime of all methods while performing feature selection which is given in Table 5.4.

Most evident is the perfect score of FRI in this setting while being the slowest method. SQ follows second and performs very good feature selection in all cases while being the second-fastest. The RFE scheme using an RF (RF) performed worst, not selecting weakly relevant features such as in Set 6 and 7. The EN also not as sensitive in this experiment, but showing the fastest runtime given its simplicity.

Table 5.1: Parameters of synthetically generated datasets for a *linear separable* classification problem as described in Section 5.3.5. Columns denote the number of features with corresponding characteristics: *n* (number of samples), *strong* (number of strongly relevant features), *weak* (number of weakly relevant features), *irr* (number of irrelevant features).

	n	strong	weak	irr
Set				
Set 1	150	6	0	6
Set 2	150	0	6	6
Set 3	150	3	4	3
Set 4	256	6	6	6
Set 5	512	1	2	11
Set 6	200	1	20	0
Set 7	200	1	20	20
Set 8	2000	10	10	50

Table 5.2: Training accuracy of models on *linearly separable* classification data generated according to Table 5.1.

	ElasticNet	FRI	RF	SQ
Set 1	0.97	0.99	0.83	1.00
Set 2	0.99	0.99	1.00	1.00
Set 3	0.98	0.99	0.86	1.00
Set 4	0.98	0.99	0.87	1.00
Set 5	0.98	1.00	0.95	1.00
Set 6	0.97	1.00	0.89	0.99
Set 7	0.98	0.99	0.90	1.00
Set 8	0.98	1.00	0.91	1.00

Table 5.3: Average F_1 measure on *linearly separable* data sets regarding feature classification.

		model	ElasticNet	FRI	RF	SQ
	type	**data**				
score	**f1**	**Set 1**	0.91	1.00	0.86	0.98
		Set 2	0.75	1.00	0.29	0.92
		Set 3	0.83	1.00	0.67	0.97
		Set 4	0.86	1.00	0.68	0.93
		Set 5	0.85	1.00	0.77	0.99
		Set 6	0.52	1.00	0.17	0.99
		Set 7	0.38	1.00	0.17	0.95
		Set 8	0.83	1.00	0.65	0.99

Table 5.4: Runtime in seconds (rounded) for the experiment described in Section 5.3.5.

	model	ElasticNet	FRI	RF	SQ
	data				
runtime	**Set 1**	0	2	0	1
	Set 2	0	2	0	1
	Set 3	0	2	0	1
	Set 4	0	3	1	3
	Set 5	0	5	2	6
	Set 6	0	4	1	2
	Set 7	0	6	3	3
	Set 8	2	201	138	80

5.3.6 *Non-Linear Feature Selection Accuracy*

While many problems can be tackled using linear models, many relations are non-linear in nature. In this experiment, we generate data which can not be separated with a linear hyperplane. Our assumption is, that the linear models EN and FRI should not perform well in this case.

We utilize the classification data generation function from scikit-learn[4] to create binary classification data with multiple opposing clusters of samples (parameter `n_clusters_per_class` = 2). We then process the informative features to produce weakly relevant (redundant) features and additional irrelevant features.[5]

Again, we generate sets with different feature configurations which are given in Table 5.5. We fit all models on 20 newly generated sets and compute the average metric values. The combined metrics are given in Table 5.6 per dataset and more concise in Table 5.7 averaged over all sets. Both linear models show low training accuracy at $\approx$ 70% which hints that the linear models can not replicate the non-linear relation. The RF-based models (SQ, RF) fare better with accuracies $\geq$ 83%. While not exceptional, it highlights the difficulty of this toy classification problem.

First, we analyse the general feature selection accuracy without discriminating between the relevance class subsets. Overall the selection accuracy

[4] `https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_classification.html`

[5] Generation function available at `https://github.com/lpfann/arfs_gen`

Table 5.5: Parameters of generated datasets for *non-linearly* separable classification data. Numeric difference between *n_features* and strong (*n_strel*) and weak (*n_redundant*) relevant features is filled with irrelevant features, i. e. NL 1 contains 10 irrelevant features.

Set	NL 1	NL 2	NL 3	NL 4
n_features	20	20	50	80
n_strel	10	4	10	10
n_redundant	0	10	10	10

Table 5.6: Statistics of a benchmark with *non-linearly* separable classification data as generated according to Table 5.5. *Precision*, *recall* and *f1* quantify the feature selection performance, whereby *accuracy* denotes the training accuracy (quality of model fit).

	model **dataset**	ElasticNet	FRI	RF	SQ
precision	**NL 1**	0.88	1.00	0.71	0.97
	NL 2	0.86	1.00	0.67	1.00
	NL 3	0.41	1.00	1.00	0.80
	NL 4	0.21	1.00	1.00	0.62
recall	**NL 1**	0.70	0.53	1.00	1.00
	NL 2	0.86	1.00	0.86	1.00
	NL 3	0.55	0.77	0.55	0.63
	NL 4	0.50	0.89	0.45	1.00
f1	**NL 1**	0.78	0.69	0.83	0.98
	NL 2	0.86	1.00	0.75	1.00
	NL 3	0.47	0.87	0.71	0.70
	NL 4	0.30	0.94	0.62	0.77
accuracy	**NL 1**	0.66	0.67	0.79	0.82
	NL 2	0.70	0.75	0.81	0.83
	NL 3	0.60	0.66	0.87	0.89
	NL 4	0.73	0.74	0.86	0.90

Table 5.7: Mean over all datasets of a benchmark with non-linearly separable classification data as in Table 5.6.

model	precision	recall	f1	accuracy
ElasticNet	0.59	0.65	0.60	0.67
FRI	1.00	0.80	0.88	0.71
RF	0.85	0.71	0.73	0.83
SQ	0.85	0.91	0.86	0.86

of all methods got worse which is apparent in Table 5.7. The EN scores last, with an average recall of 0.65 followed by the RF with greedy feature elimination (RF) with 0.71. It is beat by FRI with a recall of 0.8 even though it can not handle non-linear separable data. SQ has the highest recall with 91% of relevant features recognized. When also considering the precision, we see that FRI scores perfect precision with no false positives such that the overall result is better than SQ here with an F_1 of 0.88. This emphasizes, that FRI with its generalization bounds is much more sensitive to the true feature relevance but misses out on features with a non-linear contribution because of its roots as a linear SVM.

5.3.7 *Relevance Classification*

The general feature selection accuracy evaluation in the sections before does not consider the difference between strong and weak relevance. From

Table 5.8: Analysis of feature selection accuracy grouped by relevance class subsets on *linearly separable data*.

		Weakly		Strongly	
		FRI	SQ	FRI	SQ
precision	**Set 1**	1.00	1.00	1.00	1.00
	Set 2	1.00	0.87	1.00	1.00
	Set 3	1.00	0.99	1.00	0.74
	Set 4	1.00	0.99	1.00	0.86
	Set 5	0.99	0.99	1.00	0.49
	Set 6	1.00	1.00	1.00	0.50
	Set 7	1.00	0.99	1.00	0.46
	Set 8	0.99	1.00	1.00	0.90
recall	**Set 1**	1.00	1.00	1.00	0.95
	Set 2	1.00	1.00	1.00	1.00
	Set 3	1.00	0.72	1.00	0.99
	Set 4	1.00	0.68	1.00	0.99
	Set 5	1.00	0.50	1.00	1.00
	Set 6	1.00	0.93	1.00	1.00
	Set 7	1.00	0.88	1.00	1.00
	Set 8	1.00	0.85	1.00	1.00

Table 5.9: Mean feature selection accuracy metrics grouped by relevance class subsets and averaged over all *linearly separable* datasets. (Detailed: Table 5.8)

	Weakly		Strongly	
	FRI	SQ	FRI	SQ
precision	1.00	0.98	1.00	0.74
recall	1.00	0.82	1.00	0.99

all models considered before, only FRI and SQ provide can provide this distinction. We now present the precision and recall on the subsets $\mathcal{S}$ and $\mathcal{W}$ recorded in the experiments in the previous evaluations on linearly and non-linearly separable datasets. For that, we reuse the same metrics as before and independently evaluate each subset.

In Table 5.8 we see the metrics for all linear datasets and in Table 5.9 their mean. The recall for strongly relevant features is near perfect for our proposed method (SQ). The precision is not perfect though and sometimes FP selections occur, as can be seen in Set 2 where the recall is 100% (all relevant features were selected) but the precision is at 87% which hints at irrelevant features being selected as well. Additionally, in some cases such as in Set 5 SQ tends to select weakly relevant features as strongly relevant.

We also compare both methods in the much harder task from Section 5.3.6. The detailed results are given in Table 5.10 and their average in Table 5.11. Here the results are mixed. FRI achieves perfect recall for weakly relevant features but misses a lot of strongly relevant ones. This is extreme in sets NL 3 and NL 4 where it has an average recall of 0.04 and 0.01. It is more inclined to classify strongly relevant features as weakly relevant because the precision of the latter is decreased. SQ is also showing many false negative weakly relevant features while also selecting false positives which hurts its score

Table 5.10: Analysis of feature selection accuracy grouped by relevance class subsets on *non-linearly* separable data.

		Weakly		Strongly	
		precision	recall	precision	recall
dataset	**model**				
NL 1	**FRI**	-	-	1.00	0.11
	SQ	-	-	1.00	0.59
NL 2	**FRI**	0.83	1.00	1.00	0.50
	SQ	0.99	1.00	1.00	0.97
NL 3	**FRI**	0.67	1.00	1.00	0.04
	SQ	0.20	0.14	0.86	0.87
NL 4	**FRI**	0.57	1.00	1.00	0.01
	SQ	0.18	0.39	0.65	0.87

Table 5.11: Mean feature selection accuracy metrics grouped by relevance class subsets and averaged over all *non-linearly* separable datasets. (Detailed: Table 5.10)

	Weakly		Strongly	
	precision	recall	precision	recall
model				
FRI	0.69	1.00	1.00	0.16
SQ	0.46	0.51	0.88	0.82

in that setting. On the other hand, it is quite balanced in the classification of strongly relevant features and correctly selects and classifies over 80% of them and the runtime (Table 5.4) also scales better with dataset size.

5.4 CONCLUSION

In this chapter, we presented a new feature selection approach which builds upon Boruta and statistical methods to find the all-relevant feature set including the distinction between strong and weak relevance. We could demonstrate the general selection accuracy in the linear and non-linear case which outperforms several existing approaches, which is in line with the original method. More interestingly, we compared the identification of relevance classes with FRI from Chapter 3. In an evaluation on linear data, it came very close to the perfect results of FRI while being much better than other existing methods. For the non-linear case, the results were not as clear, as SQ could recall most of the relevant features, more than FRI and the alternatives, but was less precise than FRI. In the discrimination between strong and weak relevance, SQ's results were balanced but not perfect, compared to FRI which had a very bad recall of strong relevant features but overall better precision. Given the overall performance, we can recommend using SQ for all-relevant feature selection in both linear and non-linear problems in a use case where analysis of the feature space is important.

6 CONCLUSION

Discussion In this thesis, we covered several feature selection algorithms for analytical applications, focused on aspects posed in the research questions in Section 1.2. In the following, we discuss the results.

> **RQ 1**: How to uncover all relevant features in a machine learning setting, where a degree of redundancy in the feature space is present, with *high precision* and *efficient* runtime?

In this thesis, we utilized the existing method of feature relevance bounds with their theoretical guarantees and a definition of relevance which allows for the selection of all relevant features in the linear problem space. In practice, they exhibit inaccuracies because of numerical problems with the LP solvers and are sensitive to parameter choices.

We overcame the inaccuracies using statistical thresholds to better differentiate between irrelevant and relevant features and could achieve very good accuracies on synthetic and real data.

The original feature relevance bounds were only defined for classification data. We proposed a definition for the class of ordinal regression using explicit order constraints.

While the linear relevance bounds are quite efficient using linear programming for an all-relevant feature selection method, we further improved upon this by providing a parallel implementation which makes it possible to use them interactively on small datasets. On medium to large datasets, the implementation also allows cluster computing over many compute nodes. Still, the complexity for extremely large datasets is high and a conservative preprocessing should be used to filter out likely irrelevant features, without rejecting possible redundancies.

For non-linear problems, we looked into the existing Boruta method, which uses efficient Random Forests and their importance values for all-relevant feature selection, including redundant features. Its general high accuracy was already proven in literature and in our comparison on linear problem data it was only beat by the more specific linear relevance bounds method.

> **RQ 2**: In the presence of weakly relevant features, which imply shared information, can we identify those features and their relatives?

The original relevance bounds method could distinguish between strong and weak relevance, which allowed insight into feature set composition, including weak relevant features. Their presence implies that alternative features can be used, but the identity of those alternatives is unknown and manual experimentation is required, growing in complexity with dimensionality. We proposed a new grouping mechanism, which automatically tests relevant feature alternatives using feature constraints in the LP definition and recording the variances. They can be clustered for visualization to allow analysing non-trivial dataset sizes. We also tested using two simpler methods directly on the feature space and clustering them similarly. While the relevance bound approach is slightly more accurate in uncovering feature groups than simple clustering methods, its complexity and runtime is much higher.

In the non-linear problem space we extended Boruta with a sequential feature classification, which could decompose the all-relevant feature set into strong and weak relevant features.

> **RQ 3**: Can the relevance of privileged features in a Privileged Information (PI) setting be computed similarly to regular features?

Together with the work on ordinal regression, we also introduced the concept of relevance for privileged features at the time of training. We defined the feature relevance bounds in that context and also proposed an all-relevant feature selection method for it, analogous to the non-privileged case. On synthetic data, it performed better than a feature selection model without the distinction in normal and privileged information.

Overall, we have shown that an approximation of all-relevant feature selection for small to medium size datasets is feasible to compute with high accuracy. It can be applied to various settings and in conjunction with two highly relevant models of the biomedical domain to properly select compact feature sets including redundancies. It can also be the foundation for further analysis of the feature set, as we have shown with an automatic grouping of related features and as a tool for the analysis of fairness.

Outlook Now after concluding with the work achieved, it is also necessary to describe further avenues and aspects, which could not be researched in the scope of this thesis.

One big aspect of all-relevant feature selection is the computational complexity involved in the solution. As said before, the approximation of the ARFS for extremely high dimensional datasets is still intractable. Here, a filtering preprocessing step would be necessary and it’s not clear which filter would be the most conservative one, keeping a maximum number of relevant features.

Another aspect is the grouping of related features in the non-linear problem space for which our thesis does not propose a solution. While we experimented with a grouping procedure in the context of Random Forests and their importance values, the results were very inconsistent and further research is necessary.

Appendices

APPENDIX A

APPENDIX

A.1 RELEVANCE BOUNDS FOR ORDINAL REGRESSION

A.1.1 *Feature Relevance Bounds for Ordinal Regression with Implicit Order*

In the following, we are defining the relevance bounds for the implicit variant from Section 4.1. The definition is very similar to Section 4.1.2, and the following will be very concise.

Assume a training set X. Denote an optimum solution of problem in (4.3) as $(\tilde{\boldsymbol{w}}, \tilde{b}_j, \tilde{\xi}_i^j, \tilde{\chi}_i^j)$. We define

$$\mathcal{L} := \sum_{j=1}^{l-1} \left(\sum_{k=1}^{j} \sum_{i=1}^{n^k} \chi_{ki}^j + \sum_{k=j+1}^{l} \sum_{i=1}^{n^k} \xi_{ki}^j \right)$$

as the sum of all slack variables. The optimum solution induces the value

$$\mu_X := \frac{1}{2} \|\tilde{\boldsymbol{w}}\|_1 + C \cdot \mathcal{L}$$

which is uniquely determined by X.

The class of equivalent good hypotheses is proxied by

$$\begin{aligned} F_\delta(X) \quad := \quad & \{\boldsymbol{w} \in \mathbb{R}^n \mid \exists \boldsymbol{\xi}, \boldsymbol{\chi}, b \text{ such that constraints in (4.3) hold,} \\ & \frac{1}{2}\|\boldsymbol{w}\|_1 + C \cdot \mathcal{L} \leq (1+\delta) \cdot \mu_X\} \end{aligned}$$

Problem minRel(ℓ)**:**

$$\begin{aligned} \min_{w,b,\chi,\xi} \quad & |w_\ell| & \text{(A.1)} \\ \text{s.t. for all } i,j \quad & \text{conditions in (4.3) hold} & \\ & \frac{1}{2}\|\boldsymbol{w}\|_1 + C \cdot \mathcal{L} \leq (1+\delta) \cdot \mu_X & \text{(A.2)} \end{aligned}$$

Problem maxRel(ℓ)**:**

$$\begin{aligned} \max_{w,b,\chi,\xi} \quad & |w_\ell| & \text{(A.3)} \\ \text{s.t. for all } i,j \quad & \text{conditions in (4.3) hold} & \\ & \frac{1}{2}\|\boldsymbol{w}\|_1 + C \cdot \mathcal{L} \leq (1+\delta) \cdot \mu_X & \text{(A.4)} \end{aligned}$$

As before, this problem can be equivalently phrased as an LP.

A.1.2 *Proof of Generalization Bounds*

This is proof for the generalization bounds in Section 4.1.2 as taken from [Pfa+20]. Recall Theorem 26.15 from Understanding Machine Learning [SB14]:

Theorem 3. *Suppose that $\mathcal{D}$ is a distribution on $X \times Y$ such that with probability 1 we have $\|x\|_\infty \leq R$. Let $\mathcal{H} = \{\boldsymbol{w} \in \mathbb{R}^d \mid \|\boldsymbol{w}\|_1 \leq B\}$ and let $l : \mathcal{H} \times X \times Y \to \mathbb{R}$ be of the form $l(\boldsymbol{w}, (x,y)) = \phi(\langle \boldsymbol{w}, x \rangle, y)$ where $\phi : \mathbb{R} \times Y \to \mathbb{R}$ is such that for all $y \in Y$, the function $a \mapsto \phi(a,y)$ is η-Lipschitz and such that $\max_{a \in [-RB,RB]} |\phi(a,y)| \leq c$. Then, for any $\tau \in (0,1)$ with probability of at least $1 - \tau$ over the choice of i.i.d. sample of size n, for all $\boldsymbol{w} \in \mathcal{H}$,*

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[l(\boldsymbol{w}, x, y)] \leq \frac{1}{n} \sum_{i=1}^{n} l(\boldsymbol{w}, x_i, y_i) + 2\eta RB \sqrt{\frac{2\log(2d)}{n}} + c\sqrt{\frac{2\ln(2/\tau)}{n}}.$$

To apply this theorem we have to reformulate our classifier as a collection of binary classifiers. Since all classes use the same subspace spanned by $\boldsymbol{w}$ it is enough to distinguish neighbouring classes, i. e. every b_j gives rise to a classifier that allows us to decide whenever x belongs to one of $0, \dots, j$ or $j+1, \dots, |L|$. Consider the ramp loss

$$\begin{aligned} l_{\prec j}(\boldsymbol{w}, \mathbf{b}, x, y) &= \min\{1, \max\{0, 1 - \mathbf{1}_{y \prec j}(\boldsymbol{w}^\top x - b_j)\}\}, \\ l_j(\boldsymbol{w}, \mathbf{b}, x, y) &= l_{\leq j}(\boldsymbol{w}, \mathbf{b}, x, y) + l_{\geq j}(\boldsymbol{w}, \mathbf{b}, x, y), \\ l(\boldsymbol{w}, \mathbf{b}, x, y) &= l_y(\boldsymbol{w}, \mathbf{b}, x, y) \end{aligned}$$

where $\mathbf{1}_{y \prec j} = 1$ if $y \prec j$ and -1 otherwise for some comparison operation $\cdot \prec \cdot$. Notice that l corresponds to the implicit order constrains, which is an upper bound for the explicit loss where only neighbouring classes are considered, rather than all classes. By using this loss function it is clear that the loss of the original classifier is bounded by the sum of all those binary classifiers. Since the ramp loss is 1-Lipschitz and maps to the interval $[0, 1]$ we may apply Theorem 3 to obtain

$$\begin{aligned} \mathbb{E}_{(x,y)\sim\mathcal{D}}[l(\boldsymbol{w}, x, y)] &\leq \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\sum_{j=1}^{|L|} (l_{\leq j}(\boldsymbol{w}, x, y) + l_{\geq j}(\boldsymbol{w}, x, y)) \right] \\ &= \sum_{j=1}^{|L|} \left(\mathbb{E}_{(x,y)\sim\mathcal{D}} \left[l_{\leq j}(\boldsymbol{w}, x, y) \right] + \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[l_{\geq j}(\boldsymbol{w}, x, y) \right] \right) \\ &\leq \sum_{j=1}^{|L|} \left(\frac{1}{n} \sum_{i=1}^{n} (l_{\leq j}(\boldsymbol{w}, x_i, y_i) + l_{\geq j}(\boldsymbol{w}, x_i, y_i)) \right. \\ &\qquad \left. + 4RB\sqrt{\frac{2\log(2d)}{n}} + 2\sqrt{\frac{2\ln(2/\tau)}{n}} \right) \end{aligned}$$

for all $\boldsymbol{w}$ such that $\|\boldsymbol{w}\|_1 \leq B$ with probability $1 - \tau$ over the choice of sample. In particular, setting $\rho_j = \sum_i \tilde{\xi}_i^j + \tilde{\chi}_i^j$ and $\rho = \sum_j \rho_j$ to the hinge loss of the baseline classifier and using the fact that the hinge loss upper bounds ramp loss, this gives rise to

$$L_{\mathcal{D}}(\tilde{\boldsymbol{w}}, \tilde{\mathbf{b}}) \leq |L| \left(\frac{\rho}{n} + 4\|\tilde{\boldsymbol{w}}\|_1 R\sqrt{\frac{2\log(2d)}{n}} + 2\sqrt{\frac{2\ln(2/\tau)}{n}} \right)$$

for the generalization error of the baseline linear classifier $(\tilde{\boldsymbol{w}}, \tilde{\mathbf{b}})$ and

$$L_{\mathcal{D}}(h) \leq |L| \left(\frac{\rho}{n} + 4(1+\delta)\|\tilde{\boldsymbol{w}}\|_1 R\sqrt{\frac{2\log(2d)}{n}} + 2\sqrt{\frac{2\ln(2/\tau)}{n}} \right)$$

for all $h \in \mathcal{H}_\delta(\tilde{\boldsymbol{w}})$, with probability at least $1 - \tau$ over the choice of training sample, i. e. our choice of constraints allow the generalization error upper bound to increase by $4\delta\|\tilde{\boldsymbol{w}}\|_1 |L| R\sqrt{\frac{2\log(2d)}{n}}$.

A.1.3 *Proof of Theorem 1*

Here we give proof for Theorem 1. We rely on Theorem 4 in [Göp+18], which states the following: Assume two optimization problems

$$\begin{aligned} \text{Problem A} : \min_x h_1(x) \text{ s.t. } x \in A_1 \\ \text{Problem B} : \min_y h_2(y) \text{ s.t. } y \in A_2 \end{aligned}$$

Assume mappings $f : A_1 \to A_2$ and $g : A_2 \to A_1$ exist such that for all $x \in A_1, y \in A_2$

$$\begin{aligned} h_2(y) < h_2(f(x)) \quad &\Rightarrow \quad h_1(g(y)) < h_1(x) \\ h_1(x) < h_1(g(y)) \quad &\Rightarrow \quad h_2(f(x)) < h_2(y) \end{aligned}$$

Then the two problems A and B are equivalent in the sense that the mappings f and g establish direct correspondences of their global optima.

A.1.4 *Equivalence of* minRel(ℓ) *and* minRel$^*(\ell)$

Solutions of minRel(ℓ) have the form

$$\Big(\boldsymbol{w} = (w_1, \dots, w_d), \boldsymbol{b} = (b_1, \dots, b_{l-1}), \chi = (\chi_1^1, \dots, \chi_{m_{l-1}}^{l-1}), \xi = (\xi_1^2, \dots, \xi_{m_l}^l)\Big)$$

minRel$^*(\ell)$ combines this form with an additional vector $\hat{\boldsymbol{w}} = (\hat{w}_1, \dots, \hat{w}_d)$. Define the mapping

$$f : (\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\chi}, \boldsymbol{\xi}) \mapsto \big(\boldsymbol{w}, \hat{\boldsymbol{w}} = |\boldsymbol{w}| := (|w_1|, \dots, |w_d|), \boldsymbol{b}, \boldsymbol{\chi}, \boldsymbol{\xi}\big)$$

and the mapping

$$g : (\boldsymbol{w}, \hat{\boldsymbol{w}}, \boldsymbol{b}, \boldsymbol{\chi}, \boldsymbol{\xi}) \mapsto (\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\chi}, \boldsymbol{\xi}).$$

f is a mapping in between feasible sets. The same holds for g, since the constraints (4.6) ensure $\sum_k \hat{w}_k \geq \|\boldsymbol{w}\|_1$.

Given an element of the feasible set of the two problems, denoted by $x := (\boldsymbol{w}^A, \boldsymbol{b}^A, \boldsymbol{\chi}^A, \boldsymbol{\xi}^A)$ and $y := (\boldsymbol{w}^B, \hat{\boldsymbol{w}}^B, \boldsymbol{b}^B, \boldsymbol{\chi}^B, \boldsymbol{\xi}^B)$, respectively. Assume $h_2(y) < h_2(f(x))$, i. e. $\hat{w}_\ell^B < |w_\ell^A|$. Then constraints (4.6) ensure $|w_\ell^B| \leq \hat{w}_\ell^B$, hence $|w_\ell^B| < |w_\ell^A|$, i. e. $h_1(g(y)) < h_1(x)$.

Conversely, $h_1(x) < h_1(g(y))$ implies $|w_\ell^A| < |w_\ell^B|$ hence constraints (4.6) ensure $|w_\ell^A| < \hat{w}_\ell^B$, i. e. $h_2(f(x)) < h_2(y)$.

A.1.5 *Equivalence of* maxRel(ℓ) *and the optimum of* maxRel$^*_{pos}(\ell)$ *and* maxRel$^*_{neg}(\ell)$

We consider two problems which are associated to maxRel(ℓ):

- maxRel$_{pos}(\ell)$ equals maxRel(ℓ) with the additional constraint $w_\ell \geq 0$
- maxRel$_{neg}(\ell)$ equals maxRel(ℓ) with the additional constraint $w_\ell \leq 0$

Since these two auxiliary problems decompose the feasible set of the original one into two halves, we can solve those and take whichever solution is best instead of solving maxRel(ℓ). Thus, we can show equivalence of these two subproblems to the versions as introduced in Theorem 1. Instead of maximization, we can focus on the minimization of the respective negative of the original objectives, to phrase the setting within the notation of Theorem 4 in [Göp+18].

We show equivalence of maxRel$_{pos}(\ell)$ and maxRel$^*_{pos}(\ell)$. Define the mapping f as identity for $(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\chi}, \boldsymbol{\xi})$ and $\hat{\boldsymbol{w}} = |\boldsymbol{w}| := (|w_1|, \dots, |w_d|)$. Define the mapping g as projection of $(\boldsymbol{w}, \hat{\boldsymbol{w}}, \boldsymbol{b}, \boldsymbol{\chi}, \boldsymbol{\xi})$ onto all elements but $\hat{\boldsymbol{w}}$. f and g are mappings in between the feasible sets. Note that constraints $w_\ell \geq 0$ and $\hat{w}_\ell \leq w_\ell$ are required at this step.

Given elements of the feasible sets of the problems

$$\begin{aligned} x &:= (\boldsymbol{w}^A, b^A, \chi^A, \xi^A) \\ y &:= (\boldsymbol{w}^B, \hat{\boldsymbol{w}}^B, b^B, \chi^B, \xi^B). \end{aligned}$$

Assume $h_2(y) < h_2(f(x))$, i. e. $-\hat{w}^B_\ell < -|w^A_\ell|$. Then the constraints (4.6) and (4.7) ensure $\hat{w}^B_\ell = w^B_\ell$ and $\hat{w}^B_\ell \geq 0$, hence $-|w^B_\ell| < -|w^A_\ell|$, i. e. $h_1(g(y)) < h_1(x)$.

Conversely, $h_1(x) < h_1(g(y))$ implies $-|w^A_\ell| < -|w^B_\ell|$. Hence, $-|w^A_\ell| < -\hat{w}^B_\ell$, i. e. $h_2(f(x)) < h_2(y)$ due to constraints (4.6) and (4.7).

Similarly, equivalence of $\text{maxRel}_{neg}(\ell)$ and $\text{maxRel}^*_{neg}(\ell)$ can be shown. f and g are as above. These are mappings in between feasible sets. Note that constraints $w_\ell \geq 0$ and $\hat{w}_\ell \leq w_\ell$ are required at this step.

A.1.6 *Scaling of Ordinal Regression Feature Selection with Privileged Information*

Here we evaluate the scaling of our implementation in the setting without privileged information. We already discussed the theoretical time complexity bounds in Section 4.2.2 where we concluded that the overall method with feature selection is in $\mathcal{O}(n^3 + (dz + c)n^{2.5})$. We now run two separate experiments where we generate artificial sets as described earlier and scale up their size to the number of instances n and number of features d. In the first experiment we set $d = 20$ and scale n between 10 and 10000 and in the second we set $n = 500$ and scale d between 10 and 500. Our implementation is using the high-level library *cvxpy*[1] and the ECOS solver [DCB13] and presents no specific adaption for the problems at hand. The implementation runtime is measured on a modern Intel Xeon processor. Additionally, because relevance bounds can be computed in parallel, we run both experiments with one single thread and 8 threads in parallel.

In Figure A.3 results for both experiments are given. One can see that the complexity can limit the application of the method to small to medium-sized problems. This is in line with other *all* relevant feature selection methods [Pfa+19a] which exhibit much higher runtimes than simple sparse methods. While slightly bigger sets with, e.g. $n > 10^4$ or $d > 500$ are feasible, multiprocessing is recommended. For bigger data sets, further optimization or filtering of the feature space is necessary.

A.1.7 *Features of the COMPAS dataset*

This section describes all the features of the COMPAS dataset that we use in our analysis in Section 4.1.3. The features are listed in Table A.1. All categorical variables are One-Hot-Coded for the analysis. The ethnicities are one of {African-American, Caucasian, Hispanic, Asian, Native American, Other}, the sexes are male or female, the age is grouped into {less than 25, 25–45, greater than 45} and the charge can be one of {felony, misdemeanour, offence}. The total number of features fed into the first model is 20. After eliminating all ethnic information, the count reduces to 14.

[1] https://www.cvxpy.org/

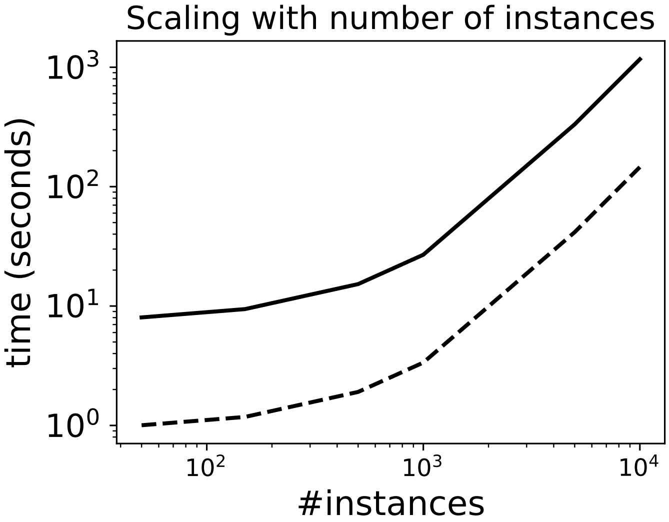

Figure A.1: Instances

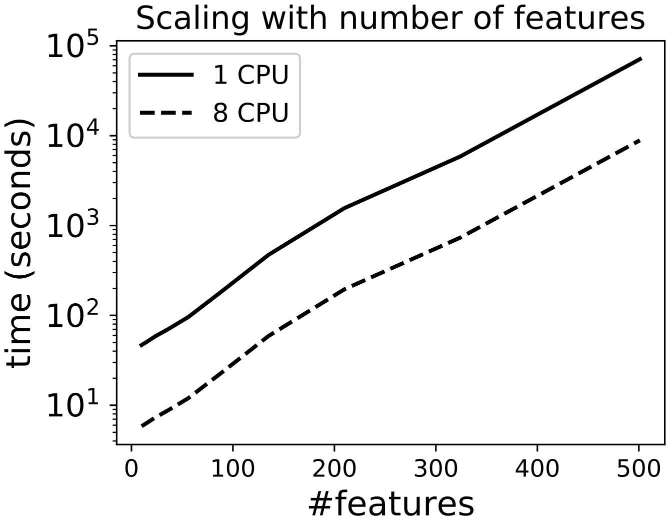

Figure A.2: Features

Figure A.3: Plot of runtime scaling concerning the number of instances (a) and number of features (b). Additionally, both show a comparison between single thread (1 CPU) and multi-threaded run (8 CPU).

Table A.1: Description of features of the COMPAS dataset used for the analysis in Section 4.1.3.

Feature Name	Type	Description	One-Hot encoding
Juv_fel_count	Numerical	# Felonies as a juvenile	No
Juv_misd_count	Numerical	# Misdemeanour as a juvenile	No
Juv_other_count	Numerical	# Offences as a juvenile	No
Priors_count	Numerical	# Prior convictions	No
Is_recid	Binary	If recidivism happened	No
Is_violent_recid	Binary	If violent recidivism happened	No
Ethnicity	Categorical	One of 6 ethnicities	Yes
Sex	Categorical	One of 2 sexes	Yes
Age	Categorical	One of 3 age groups	Yes
C_charge	Categorical	One of 3 charge groups	Yes

GLOSSARY

$\mathcal{A}$ Set of all relevant Features. 10, 11, 21, 63–65, 68, 69, 109

$\mathcal{I}$ Set of all irrelevant features. 10, 11, 63, 64, 109

$\mathcal{M}$ Minimal optimal feature set. 11, 63–70, 109

$\mathcal{S}$ Set of all strongly relevant features. 9–11, 63–66, 68, 79, 109

$\mathcal{W}$ Set of all weakly relevant features. 9–11, 63–66, 68, 79, 109

$\mathcal{D}$ All Features. 11, 109

F_1 Harmonic mean between Precision and Recall. 14, 29, 30, 47, 48, 60

Boruta An ensemble based wrapper model for feature selection utilizing a statistical test to control noise. 13, 28–32, 36, 63, 67–69, 71, 73, 74, 80, 81, 109–111

corr-clust Agglomerative clustering using Pearson correlation of the feature space.. 35

HDBSCAN Hierarchical Density-Based Spatial Clustering of Applications with Noise. 33, 35, 111

Lasso Sparse model regularization using L_1 Norm. 12, 14, 28, 40, 41, 43, 45, 46, 64, 67

precision Ratio of correctly selected Features. Shows amount of noise included. 29–31, 46–48, 60, 71, 75, 78, 79, 112

recall Ratio of relevant Features selected. 29–31, 47, 48, 60, 75, 78, 79, 112

SQ Sequential Feature Classification. 71, 75, 77–80

V-measure Harmonic mean between homogeneity and completeness measures for clustering.. 33, 35, 111

ACRONYMS

FN False Negative. 28, 29, 75

FP False Positive. 28, 29, 75, 79

TN True Negative. 28

TP True Positive. 28, 29, 75

ARFS all-relevant feature selection problem. 2, 4, 10–16, 28, 53, 82

AUC area under the curve. 31, 111

EFS Ensemble Feature Selection. 13, 28–32, 36, 111

EN ElasticNet. 12, 28–32, 36, 40, 42, 45–51, 59, 60, 70, 75, 77, 78, 111, 112

FRI Feature Relevance Intervals method. 17, 18, 22, 28–32, 35–37, 46, 49, 51, 60, 63, 70, 75, 77–80, 109, 111

FS Feature Selection. 2, 10, 12, 13, 30

LP Linear Program. 22, 58, 81, 86

LUPI Learning using privileged information. 53, 54, 58–61, 109

MFS minimal-optimal feature selection problem. 11

MI mutual information. 8, 11

MMAE Macro-averaged Mean Absolute Error. 45, 49, 50, 111

ORP Ordinal Regression Problem. 39–41, 51, 61

PI Privileged Information. 53, 54

RF Random Forest. 8, 13, 29, 30, 63–65, 67–75, 77, 78, 109–111

RFE Recursive Feature Elimination. 12, 65, 70, 71, 75

RFECV Recursive Feature Elimination with Cross-Validation. 45, 73, 74, 110

ROC receiver operation characteristics. 31, 111

SES statistically equivalent signature. 13

SMOTE Synthetic Minority Over-sampling Technique. 28

SS Stability Selection. 28–32, 36

SVM Support Vector Machine. 12, 14, 15, 39, 40, 49, 53, 54, 58

BIBLIOGRAPHY

[ABR00] Davide Anguita, Andrea Boni, and Sandro Ridella. "Evaluating the Generalization Ability of Support Vector Machines through the Bootstrap". In: *Neural Processing Letters* 11.1 (Feb. 1, 2000), pp. 51–58. ISSN: 1370-4621, 1573-773X. DOI: 10.1023/A:1009636300083.

[Aga08] Shivani Agarwal. "Generalization Bounds for Some Ordinal Regression Algorithms". In: *Algorithmic Learning Theory, 19th International Conference, ALT 2008, Budapest, Hungary, October 13-16, 2008. Proceedings*. Ed. by Yoav Freund, László Györfi, György Turán, and Thomas Zeugmann. 2008, pp. 7–21. DOI: 10.1007/978-3-540-87987-9_6.

[Ang+16] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. "How We Analyzed the COMPAS Recidivism Algorithm". In: (2016). URL: https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm.

[Bat94] R. Battiti. "Using Mutual Information for Selecting Features in Supervised Neural Net Learning". In: *IEEE Trans. Neural Netw.* 5.4 (July 1994), pp. 537–550. ISSN: 1941-0093. DOI: 10.1109/72.298224.

[BES10] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. "Feature Selection for Ordinal Regression". In: *Proceedings of the 2010 ACM Symposium on Applied Computing* (New York, NY, USA). SAC '10. New York, NY, USA: ACM, 2010, pp. 1748–1754. ISBN: 978-1-60558-639-7. DOI: 10.1145/1774088.1774461.

[BF16] A. Bibal and Benoît Frénay. "Interpretability of Machine Learning Models and Representations: An Introduction". In: *ESANN*. 2016.

[BL97] Avrim L. Blum and Pat Langley. "Selection of Relevant Features and Examples in Machine Learning". In: *Artificial Intelligence* 97.1-2 (Dec. 1997), pp. 245–271. ISSN: 00043702. DOI: 10.1016/S0004-3702(97)00063-5.

[BPM04] Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data". In: *SIGKDD Explor Newsl* 6.1 (June 2004), pp. 20–29. ISSN: 1931-0145. DOI: 10.1145/1007730.1007735.

[Büh+13] Peter Bühlmann, Philipp Rütimann, Sara van de Geer, and Cun-Hui Zhang. "Correlated Variables in Regression: Clustering and Sparse Estimation". In: *J. Stat. Plan. Inference* 143.11 (Nov. 2013), pp. 1835–1858. ISSN: 0378-3758. DOI: 10.1016/j.jspi.2013.05.019.

[BWG01] S. C. Bagley, H. White, and B. A. Golomb. "Logistic Regression in the Medical Literature: Standards for Use and Reporting, with Particular Attention to One Medical Domain". In: *J Clin Epidemiol* 54.10 (Oct. 2001), pp. 979–985. ISSN: 0895-4356. pmid: 11576808.

[BZ08] Riccardo Bellazzi and Blaz Zupan. "Predictive Data Mining in Clinical Medicine: Current Issues and Guidelines". In: *International Journal of Medical Informatics* 77.2 (Feb. 1, 2008), pp. 81–97. ISSN: 1386-5056. DOI: 10.1016/j.ijmedinf.2006.11.006.

[Cha+02] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. "SMOTE: Synthetic Minority Over-Sampling Technique". In: *1* 16 (June 1, 2002), pp. 321–357. ISSN: 1076-9757. DOI: 10.1613/jair.953.

[CK05] Wei Chu and S. Sathiya Keerthi. "New Approaches to Support Vector Ordinal Regression". In: *Proceedings of the 22nd International Conference on Machine Learning*. ACM, 2005, pp. 145–152. URL: http://www.gatsby.ucl.ac.uk/~chuwei/paper/icmlsvor.pdf.

[CK07] Wei Chu and S. Sathiya Keerthi. "Support Vector Ordinal Regression". In: *Neural Comput.* 19.3 (Mar. 2007), pp. 792–815. ISSN: 0899-7667. DOI: 10.1162/neco.2007.19.3.792.

[CL08] Yin-Wen Chang and Chih-Jen Lin. "Feature Ranking Using Linear SVM". In: *Causation and Prediction Challenge*. Causation and Prediction Challenge. Dec. 31, 2008, pp. 53–64. URL: http://proceedings.mlr.press/v3/chang08a.html (visited on 08/01/2018).

[CLS19] Michael B. Cohen, Yin Tat Lee, and Zhao Song. "Solving Linear Programs in the Current Matrix Multiplication Time". In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2019. Phoenix, AZ, USA: Association for Computing Machinery, June 23, 2019, pp. 938–942. ISBN: 978-1-4503-6705-9. DOI: 10.1145/3313276.3316303. arXiv: 1810.07896.

[CMR19] Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. *Rank-Consistent Ordinal Regression for Neural Networks*. Aug. 5, 2019. arXiv: 1901.07884 [cs, stat]. URL: http://arxiv.org/abs/1901.07884 (visited on 04/29/2020).

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, Aug. 26, 1991. 574 pp. ISBN: 978-0-471-06259-2. Google Books: CX9QAAAAMAAJ.

[CW06] Joseph A. Cruz and David S. Wishart. "Applications of Machine Learning in Cancer Prediction and Prognosis". In: *Cancer Inform.* 2 (Jan. 2006), p. 117693510600200030. ISSN: 1176-9351. DOI: 10.1177/117693510600200030.

[DA06] Ramón Díaz-Uriarte and Sara Alvarez de Andrés. "Gene Selection and Classification of Microarray Data Using Random Forest". In: *BMC Bioinformatics* 7.1 (Jan. 6, 2006), p. 3. ISSN: 1471-2105. DOI: 10.1186/1471-2105-7-3.

[DCB13] Alexander Domahidi, Eric Chu, and Stephen Boyd. "ECOS: An SOCP Solver for Embedded Systems". In: *2013 European Control Conference (ECC)*. 2013 European Control Conference (ECC). Zurich, Switzerland: IEEE, July 2013, pp. 3071–3076. ISBN: 978-3-033-03962-9. DOI: 10.23919/ECC.2013.6669541.

[DK17] Dua Dheeru and Efi Karra Taniskidou. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[Dor+13] Carsten F. Dormann et al. "Collinearity: A Review of Methods to Deal with It and a Simulation Study Evaluating Their Performance". In: *Ecography* 36.1 (2013), pp. 27–46. ISSN: 1600-0587. DOI: 10.1111/j.1600-0587.2012.07348.x.

[DSS19] Frauke Degenhardt, Stephan Seifert, and Silke Szymczak. "Evaluation of Variable Selection Methods for Random Forests and Omics Data Sets". In: *Briefings in Bioinformatics* 20.2 (Mar. 25, 2019), pp. 492–503. ISSN: 1477-4054. DOI: 10.1093/bib/bbx124.

[FCF17] Kelwin Fernandes, Jaime S. Cardoso, and Jessica Fernandes. "Transfer Learning with Partial Observability Applied to Cervical Cancer Screening". In: *IbPRIA*. 2017. DOI: 10.1007/978-3-319-58838-4_27.

[FDV13a] Benoît Frénay, Gauthier Doquire, and Michel Verleysen. "Is Mutual Information Adequate for Feature Selection in Regression?" In: *Neural Networks* 48 (Dec. 1, 2013), pp. 1–7. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2013.07.003.

[FDV13b] Benoît Frénay, Gauthier Doquire, and Michel Verleysen. "Theoretical and Empirical Study on the Potential Inadequacy of Mutual Information for Feature Selection in Classification". In: *Neurocomputing* 112 (July 2013), pp. 64–78. ISSN: 09252312. DOI: 10.1016/j.neucom.2012.12.051.

[FDV14] Benoît Frénay, Gauthier Doquire, and Michel Verleysen. "Estimating Mutual Information for Feature Selection in the Presence of Label Noise". In: *Computational Statistics & Data Analysis* 71 (Mar. 1, 2014), pp. 832–848. ISSN: 0167-9473. DOI: 10.1016/j.csda.2013.05.001.

[FH01] Eibe Frank and Mark Hall. "A Simple Approach to Ordinal Classification". In: *Machine Learning: ECML 2001*. Ed. by Luc De Raedt and Peter Flach. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2001, pp. 145–156. ISBN: 978-3-540-44795-5.

[Fra+07] D. François, F. Rossi, V. Wertz, and M. Verleysen. "Resampling Methods for Parameter-Free and Robust Feature Selection with Mutual Information". In: *Neurocomputing*. Advances in Computational Intelligence and Learning 70.7 (Mar. 1, 2007), pp. 1276–1288. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2006.11.019.

[FT12] Shereen Fouad and Peter Tiño. "Adaptive Metric Learning Vector Quantization for Ordinal Classification". In: *Neural Comput.* 24 11 (2012), pp. 2825–51.

[FZ16] Jianqing Fan and Wen-Xin Zhou. "Guarding against Spurious Discoveries in High Dimensions". In: *J Mach Learn Res* 17 (2016). ISSN: 1532-4435. pmid: 28936128. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5603346/ (visited on 09/10/2020).

[GE03] Isabelle Guyon and André Elisseeff. "An Introduction to Variable and Feature Selection". In: *J. Mach. Learn. Res.* 3 (Mar 2003), pp. 1157–1182. ISSN: ISSN 1533-7928.

[Gei93] Seymour Geisser. *Predictive Inference*. CRC Press, June 1, 1993. 280 pp. ISBN: 978-0-412-03471-8. Google Books: wfdlBZ_iwZoC. URL: https://books.google.de/books?id=wfdlBZ_iwZoC&printsec=frontcover&redir_esc=y#v=onepage&q&f=false.

[Gen+07] Xiubo Geng, Tie-Yan Liu, Tao Qin, and Hang Li. "Feature Selection for Ranking". In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA). SIGIR '07. New York, NY, USA: ACM, 2007, pp. 407–414. ISBN: 978-1-59593-597-7. DOI: 10.1145/1277741.1277811.

[GMS17] Baptiste Gregorutti, Bertrand Michel, and Philippe Saint-Pierre. "Correlation and Variable Importance in Random Forests". In: *Stat Comput* 27.3 (May 1, 2017), pp. 659–678. ISSN: 1573-1375. DOI: 10.1007/s11222-016-9646-1. arXiv: 1310.5726.

[Göp+18] Christina Göpfert, Lukas Pfannschmidt, Jan Philip Göpfert, and Barbara Hammer. "Interpretation of Linear Classifiers by Means of Feature Relevance Bounds". In: *Neurocomputing* 298 (July 12, 2018), pp. 69–79. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2017.11.074.

[GPH17] Christina Göpfert, Lukas Pfannschmidt, and Barbara Hammer. "Feature Relevance Bounds for Linear Classification". In: *Proceedings of the ESANN, 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. Ed. by Michele Verleysen. Bruges: Ciaco - i6doc.com, 2017, pp. 187–192.

[GSS11] Dinesh Garg, Sellamanickam Sundararajan, and Shirish Shevade. "A Game Theoretic Approach for Feature Clustering and Its Application to Feature Selection". In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Joshua Zhexue Huang, Longbing Cao, and Jaideep Srivastava. Vol. 6634. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 13–25. ISBN: 978-3-642-20840-9 978-3-642-20841-6. DOI: 10.1007/978-3-642-20841-6_2.

[Har+16] Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. "Equality of Opportunity in Supervised Learning". In: *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., 2016, pp. 3315–3323. arXiv: 1610.02413. URL: http://papers.nips.cc/paper/6374-equality-of-opportunity-in-supervised-learning.pdf.

[HD13] S. Hajian and J. Domingo-Ferrer. "A Methodology for Direct and Indirect Discrimination Prevention in Data Mining". In: *IEEE Trans. Knowl. Data Eng.* 25.7 (July 2013), pp. 1445–1459. ISSN: 1041-4347. DOI: 10.1109/TKDE.2012.72.

[HK16] F Maxwell Harper and Joseph A Konstan. "The Movielens Datasets: History and Context". In: *ACM Trans. Interact. Intell. Syst.* 5.4 (2016), p. 19.

[Hsi+08] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S. Sathiya Keerthi, and S. Sundararajan. "A Dual Coordinate Descent Method for Large-Scale Linear SVM". In: *Proceedings of the 25th International Conference on Machine Learning - ICML '08*. The 25th International Conference. Helsinki, Finland: ACM Press, 2008, pp. 408–415. ISBN: 978-1-60558-205-4. DOI: 10.1145/1390156.1390208.

[Hua+eb] Shujun Huang, Nianguang Cai, Pedro Penzuti Pacheco, Shavira Narrandes, Yang Wang, and Wayne Xu. "Applications of Support Vector Machine (SVM) Learning in Cancer Genomics". In: *Cancer Genomics Proteomics* 15.1 (2018 Jan-Feb), pp. 41–51. ISSN: 1790-6245. DOI: 10.21873/cgp.20063. pmid: 29275361.

[HY10] Zengyou He and Weichuan Yu. "Stable Feature Selection for Biomarker Discovery". In: *Computational Biology and Chemistry* 34.4 (Aug. 1, 2010), pp. 215–225. ISSN: 1476-9271. DOI: 10.1016/j.compbiolchem.2010.07.002.

[Joa06] Thorsten Joachims. "Training Linear SVMs in Linear Time". In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '06*. The 12th ACM SIGKDD International Conference. Philadelphia, PA, USA: ACM Press, 2006, p. 217. ISBN: 978-1-59593-339-3. DOI: 10.1145/1150402.1150429.

[Kam+16] Iman Kamkar, Sunil Kumar Gupta, Dinh Phung, and Svetha Venkatesh. "Stabilizing L1-Norm Prediction Models by Supervised Feature Grouping". In: *J. Biomed. Inform.* 59 (Feb. 2016), pp. 149–168. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2015.11.012.

[Kar84] N. Karmarkar. "A New Polynomial-Time Algorithm for Linear Programming". In: *Combinatorica* 4.4 (Dec. 1, 1984), pp. 373–395. ISSN: 1439-6912. DOI: 10.1007/BF02579150.

[Ke+17] Guolin Ke et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 3146–3154. URL: http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf (visited on 01/23/2020).

[Kea17] Michael Kearns. "Fair Algorithms for Machine Learning". In: *Proceedings of the 2017 ACM Conference on Economics and Computation*. EC '17. New York, NY, USA: ACM, 2017, pp. 1–1. ISBN: 978-1-4503-4527-9. DOI: 10.1145/3033274.3084096.

[KJ97] Ron Kohavi and George H. John. "Wrappers for Feature Subset Selection". In: *Artif Intell* 97.1-2 (Dec. 1997), pp. 273–324. ISSN: 0004-3702. DOI: 10.1016/S0004-3702(97)00043-X.

[Kon01] Igor Kononenko. "Machine Learning for Medical Diagnosis: History, State of the Art and Perspective". In: *Artif. Intell. Med.* 23.1 (Aug. 2001), pp. 89–109. ISSN: 0933-3657. DOI: 10.1016/S0933-3657(01)00077-X.

[KR10] Miron B. Kursa and Witold R. Rudnicki. "Feature Selection with the Boruta Package". In: *J. Stat. Softw.* 36.11 (2010). ISSN: 1548-7660. DOI: 10.18637/jss.v036.i11.

[KR11] Miron B. Kursa and Witold R. Rudnicki. "The All Relevant Feature Selection Using Random Forest". In: *CoRR* abs/1106.5112 (2011). URL: http://arxiv.org/abs/1106.5112.

[Kum14] Vipin Kumar. "Feature Selection: A Literature Review". In: *Smart Comput. Rev.* 4.3 (June 30, 2014). ISSN: 22344624. DOI: 10.6029/smartcr.2014.03.007.

[Lag+16] Vincenzo Lagani, Giorgos Athineou, Alessio Farcomeni, Michail Tsagris, and Ioannis Tsamardinos. "Feature Selection with the R Package MXM: Discovering Statistically-Equivalent Feature Subsets". In: *ArXiv E-Prints* 1611 (Nov. 2016), arXiv:1611.03227.

[Lau01] Jorma Laurikkala. "Improving Identification of Difficult Small Classes by Balancing Class Distribution". In: *Artificial Intelligence in Medicine*. Ed. by G. Goos, J. Hartmanis, J. van Leeuwen, Silvana Quaglini, Pedro Barahona, and Steen Andreassen. Vol. 2101. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 63–66. ISBN: 978-3-540-42294-5 978-3-540-48229-1. URL: http://link.springer.com/10.1007/3-540-48229-6_9 (visited on 04/19/2018).

[LdR18] Chang Li and Maarten de Rijke. "Incremental Sparse Bayesian Ordinal Regression". In: *Neural Networks* 106 (Oct. 1, 2018), pp. 294–302. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2018.07.015.

[LeC+95] Yann LeCun et al. "Learning Algorithms for Classification: A Comparison on Handwritten Digit Recognition". In: *Neural Netw. Stat. Mech. Perspect.* 261 (1995), p. 276.

[Leh+13] Maarit Lehti et al. "High-Density Lipoprotein Maintains Skeletal Muscle Function by Modulating Cellular Respiration in Mice". In: *Circulation* 128.22 (Nov. 2013), pp. 2364–2371. ISSN: 1524-4539. DOI: 10.1161/CIRCULATIONAHA.113.001551. pmid: 24170386.

[LHC20] Samuel Lalmuanawma, Jamal Hussain, and Lalrinfela Chhakchhuak. "Applications of Machine Learning and Artificial Intelligence for Covid-19 (SARS-CoV-2) Pandemic: A Review". In: *Chaos Solitons Fractals* 139 (Oct. 2020), p. 110059. ISSN: 0960-0779. DOI: 10.1016/j.chaos.2020.110059. pmid: 32834612.

[Liu+18] Lydia T. Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. "Delayed Impact of Fair Machine Learning". In: *International Conference on Machine Learning*. International Conference on Machine Learning. July 3, 2018, pp. 3150–3158. arXiv: 1803.04383. URL: http://proceedings.mlr.press/v80/liu18c.html (visited on 04/29/2020).

[Lop+16] David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. "Unifying Distillation and Privileged Information". In: *ICLR*. International Conference on Learning Representations. 2016. URL: https://arxiv.org/abs/1511.03643v3 (visited on 06/26/2019).

[MB10] Nicolai Meinshausen and Peter Bühlmann. "Stability Selection". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* (2010). ISSN: 1369-7412. URL: `http://agris.fao.org/agris-search/search.do?recordID=US201301874978` (visited on 03/01/2019).

[MH17] L. McInnes and J. Healy. "Accelerated Hierarchical Density Based Clustering". In: *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. Nov. 2017, pp. 33–42. DOI: `10.1109/ICDMW.2017.12`.

[MHA17] Leland McInnes, John Healy, and Steve Astels. "Hdbscan: Hierarchical Density Based Clustering". In: *J. Open Source Softw.* (Mar. 2017). DOI: `10.21105/joss.00205`.

[Neu+16] Ursula Neumann et al. "Compensation of Feature Selection Biases Accompanied with Improved Predictive Performance for Binary Classification by Using a Novel Ensemble Feature Selection Approach". In: *BioData Min.* 9 (Nov. 2016), p. 36. ISSN: 1756-0381. DOI: `10.1186/s13040-016-0114-4`.

[NGH17] Ursula Neumann, Nikita Genze, and Dominik Heider. "EFS: An Ensemble Feature Selection Tool Implemented as R-Package and Web-Application". In: *BioData Mining* 10 (June 27, 2017), p. 21. ISSN: 1756-0381. DOI: `10.1186/s13040-017-0142-8`.

[Nic+05] K. H. Nicolaides, K. Spencer, K. Avgidou, S. Faiola, and O. Falcon. "Multicenter Study of First-Trimester Screening for Trisomy 21 in 75 821 Pregnancies: Results and Estimation of the Potential Impact of Individual Risk-Orientated Two-Stage First-Trimester Screening: First-Trimester Screening for Trisomy 21". In: *Ultrasound Obstet. Gynecol.* 25.3 (Mar. 2005), pp. 221–226. ISSN: 09607692. DOI: `10.1002/uog.1860`.

[Nil+07] Roland Nilsson, José M. Peña, Johan Björkegren, and Jesper Tegnér. "Consistent Feature Selection for Pattern Recognition in Polynomial Time". In: *J Mach Learn Res* 8 (Dec. 2007), pp. 589–612. ISSN: 1532-4435. URL: `http://dl.acm.org/citation.cfm?id=1314498.1314519`.

[PBG17] Fabian Pedregosa, Francis Bach, and Alexandre Gramfort. "On the Consistency of Ordinal Regression Methods". In: *J. Mach. Learn. Res.* 18.55 (2017), pp. 1–35. arXiv: `1408.2327`. URL: `http://jmlr.org/papers/v18/15-495.html` (visited on 04/21/2020).

[Pea09] Judea Pearl. *Causality*. Cambridge University Press, Sept. 14, 2009. 486 pp. ISBN: 978-0-521-89560-6. Google Books: `f4nuexsNVZIC`.

[Ped+11] F. Pedregosa et al. "Scikit-Learn: Machine Learning in Python". In: *J. Mach. Learn. Res.* 12 (2011), pp. 2825–2830.

[Pfa+19a] Lukas Pfannschmidt, Christina Göpfert, Ursula Neumann, Dominik Heider, and Barbara Hammer. "FRI – Feature Relevance Intervals for Interpretable and Interactive Data Exploration". In: *2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. July 2019. DOI: 10.1109/CIBCB.2019.8791489. arXiv: 1903.00719.

[Pfa+19b] Lukas Pfannschmidt, Jonathan Jakob, Michael Biehl, Peter Tino, and Barbara Hammer. "Feature Relevance Bounds for Ordinal Regression". In: *ESANN 2019*. ESANN 2019. Bruges: i6doc, Feb. 20, 2019, ES2019–162. ISBN: 978-2-87587-065-0. URL: https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2019-162.pdf.

[Pfa+20] Lukas Pfannschmidt, Jonathan Jakob, Fabian Hinder, Michael Biehl, Peter Tino, and Barbara Hammer. "Feature Relevance Determination for Ordinal Regression in the Context of Feature Redundancies and Privileged Information". In: *Neurocomputing* (Apr. 9, 2020). ISSN: 0925-2312. DOI: 10.1016/j.neucom.2019.12.133. arXiv: 1912.04832.

[PH20] Lukas Pfannschmidt and Barbara Hammer. "Sequential Feature Classification in the Context of Redundancies". In: Apr. 15, 2020. arXiv: 2004.00658. URL: http://arxiv.org/abs/2004.00658.

[PJ20] Lukas Pfannschmidt and Jonathan Jakob. *Lpfann/Fri: Feature Relevance Intervals*. Zenodo, Mar. 31, 2020. DOI: 10.5281/zenodo.3734217.

[Pla98] John C. Platt. *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*. Technical Report MSR-TR-98-14. Redmond: Microsoft Research, Apr. 21, 1998.

[PRT08] Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. "Discrimination-Aware Data Mining". In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '08. Las Vegas, Nevada, USA: Association for Computing Machinery, Aug. 24, 2008, pp. 560–568. ISBN: 978-1-60558-193-4. DOI: 10.1145/1401890.1401959.

[PV10] Dmitry Pechyony and Vladimir Vapnik. "On the Theory of Learnining with Privileged Information". In: *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a Meeting Held 6-9 December 2010, Vancouver, British Columbia, Canada.* 2010, pp. 1894–1902.

[RH07] Andrew Rosenberg and Julia Hirschberg. "V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure". In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 410–420.

[Rod+20] Alfonso J. Rodriguez-Morales et al. "Clinical, Laboratory and Imaging Features of COVID-19: A Systematic Review and Meta-Analysis". In: *Travel Medicine and Infectious Disease* 34 (Mar. 1, 2020), p. 101623. ISSN: 1477-8939. DOI: 10.1016/j.tmaid.2020.101623.

[Rud+06] Witold R. Rudnicki, Marcin Kierczak, Jacek Koronacki, and Jan Komorowski. "A Statistical Method for Determining Importance of Variables in an Information System". In: *Rough Sets and Current Trends in Computing*. Ed. by Salvatore Greco et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006, pp. 557–566. ISBN: 978-3-540-49842-1. DOI: 10.1007/11908029_58.

[Sán+13] Javier Sánchez-Monedero, Gutiérrez, Pedro Antonio, Peter Tino, and César Hervás-Martínez. "Exploitation of Pairwise Class Distances for Ordinal Classification". In: *Neural Comput.* 25.9, MIT Press (2013).

[SB14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. New York, NY, USA: Cambridge University Press, 2014. ISBN: 1-107-05713-2 978-1-107-05713-5.

[Sha48] C. E. Shannon. "A Mathematical Theory of Communication". In: *Bell Syst. Tech. J.* 27.3 (July 1948), pp. 379–423. ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[SL02] Amnon Shashua and Anat Levin. "Ranking with Large Margin Principle: Two Approaches". In: *Proceedings of the 15th International Conference on Neural Information Processing Systems* (Cambridge, MA, USA). NIPS'02. Cambridge, MA, USA: MIT Press, 2002, pp. 961–968. URL: http://dl.acm.org/citation.cfm?id=2968618.2968738.

[Sow+13] Jan-Peter Sowa, Dominik Heider, Lars Peter Bechmann, Guido Gerken, Daniel Hoffmann, and Ali Canbay. "Novel Algorithm for Non-Invasive Assessment of Fibrosis in NAFLD". In: *PLOS ONE* 8.4 (Apr. 2013), e62439. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0062439.

[SS13] Rajen D. Shah and Richard J. Samworth. "Variable Selection with Error Control: Another Look at Stability Selection: *Another Look at Stability Selection*". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 75.1 (Jan. 2013), pp. 55–80. ISSN: 13697412. DOI: 10.1111/j.1467-9868.2011.01034.x.

[Tan+15] Fengzhen Tang, P Tino, PA Gutierrez, and H Chen. "The Benefits of Modelling Slack Variables in SVMs". In: *Neural Comput.* 27.4 (2015), pp. 954–981.

[Tib96] Robert Tibshirani. "Regression Shrinkage and Selection via the Lasso". In: *J. R. Stat. Soc. Ser. B Methodol.* 58.1 (1996), pp. 267–288. ISSN: 00359246. JSTOR: 2346178.

[TL11] Laura Toloşi and Thomas Lengauer. "Classification with Correlated Features: Unreliability of Feature Ranking and Solutions". In: *Bioinformatics* 27.14 (July 2011), pp. 1986–1994. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btr300.

[TT17] Fengzhen Tang and Peter Tiño. "Ordinal Regression Based on Learning Vector Quantization". In: *Neural Netw.* 93 (2017), pp. 76–88. DOI: 10.1016/j.neunet.2017.05.006.

[Vai89] P. M. Vaidya. "Speeding-up Linear Programming Using Fast Matrix Multiplication". In: *30th Annual Symposium on Foundations of Computer Science*. 30th Annual Symposium on Foundations of Computer Science. Oct. 1989, pp. 332–337. DOI: 10.1109/SFCS.1989.63499.

[vDam+18] Sipko van Dam, Urmo Võsa, Adriaan van der Graaf, Lude Franke, and João Pedro de Magalhães. "Gene Co-Expression Analysis for Functional Classification and Gene–Disease Predictions". In: *Briefings in Bioinformatics* 19.4 (July 20, 2018), pp. 575–592. ISSN: 1477-4054. DOI: 10.1093/bib/bbw139.

[Vel+12] Alfredo Vellido Alcacena, Martin Guerrero, Jose D, and Paulo J. G. Lisboa. "Making Machine Learning Models Interpretable". In: *ESANN 2012 Proceedings: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning: Bruges (Belgium), 25-27 April, 2012*. 2012, pp. 163–172. ISBN: 978-2-87419-047-6. URL: https://upcommons.upc.edu/handle/2117/18311.

[VI15] Vladimir Vapnik and Rauf Izmailov. "Learning Using Privileged Information: Similarity Control and Knowledge Transfer". In: *J. Mach. Learn. Res.* 16 (2015), pp. 2023–2049. URL: http://www.jmlr.org/papers/v16/vapnik15b.html (visited on 07/18/2018).

[VV09] Vladimir Vapnik and Akshay Vashist. "A New Learning Paradigm: Learning Using Privileged Information". In: *Neural Networks*. Advances in Neural Networks Research: IJCNN2009 22.5 (July 1, 2009), pp. 544–557. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2009.06.042.

[Wil72] D. L. Wilson. “Asymptotic Properties of Nearest Neighbor Rules Using Edited Data”. In: *IEEE Trans. Syst. Man Cybern.* SMC-2.3 (July 1972), pp. 408–421. ISSN: 0018-9472. DOI: 10.1109/TSMC.1972.4309137.

[Yao+17] Lan Yao, Feng Zeng, Dong-Hui Li, and Zhi-Gang Chen. “Sparse Support Vector Machine with Lp Penalty for Feature Selection”. In: *J. Comput. Sci. Technol.* 32.1 (Jan. 2017), pp. 68–77. ISSN: 1000-9000, 1860-4749. DOI: 10.1007/s11390-017-1706-2.

[YL03] Lei Yu and Huan Liu. “Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution”. In: *Proceedings of the Twentieth International Conference on International Conference on Machine Learning* (Washington, DC, USA). ICML’03. Washington, DC, USA: AAAI Press, 2003, pp. 856–863. ISBN: 1-57735-189-4. URL: http://dl.acm.org/citation.cfm?id=3041838.3041946.

[ZH05] Hui Zou and Trevor Hastie. “Regularization and Variable Selection via the Elastic Net”. In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 67.2 (2005), pp. 301–320. ISSN: 1467-9868. DOI: 10.1111/j.1467-9868.2005.00503.x.

[Zha02] Tong Zhang. “Covering Number Bounds of Certain Regularized Linear Function Classes”. In: *J. Mach. Learn. Res.* 2 (2002), pp. 527–550.

[ZSR02] Nela Zavaljevski, Fred J. Stevens, and Jaques Reifman. “Support Vector Machines with Selective Kernel Scaling for Protein Classification and Identification of Key Amino Acid Positions”. In: *Bioinforma. Oxf. Engl.* 18.5 (May 2002), pp. 689–696. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/18.5.689. pmid: 12050065.

[ZY06] Peng Zhao and Bin Yu. “On Model Selection Consistency of Lasso”. In: *J. Mach. Learn. Res.* 7 (Dec. 2006), pp. 2541–2563. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=1248547.1248637.

LIST OF FIGURES

LIST OF TABLES