UNIVERSITÄT BONN

Chair of Data Science and
Data Engineering

b-it

Multivariate Correlation Analysis for Supervised Feature Selection in High-Dimensional Data

zur Erlangung des Doktorgrades
(Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

von
Arvind Kumar Shekar
aus
Tamil Nadu, India

Bonn, 2019

Angefertigt mit Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn.

Erscheinungsjahr: 2020

Reviewers

First-supervisor
Prof. Dr. Emmanuel Mueller
Rheinische Friedrich-Wilhelms-Universität Bonn
Bonn-Aachen International Center for Information Technology

Second-supervisor
Prof. Dr. Eyke Hüllermeier
University of Paderborn
Intelligent Systems Group

Fachnahes Mitglied
Prof. Dr. Reinhard Klein
Rheinische Friedrich-Wilhelms-Universität Bonn
Institut fuer Informatik II

Fachfremdes Mitglied
Prof. Dr. Ulf-G Meißner
Rheinische Friedrich-Wilhelms-Universität Bonn
Helmholtz-Institut fuer Strahlen- und Kernphysik (Theorie)

Datum der mündlichen Prüfung
11. December. 2019

Dedicated to my parents.

Abstract

In today's scenario, several application domains involve collection of a large number of process variables also known as features. The high-dimensional feature space is commonly used for performing analytical tasks such as regression and classification. In such a high-dimensional feature space, not all features are relevant for the defined analytical task (or target) and several of them may be redundant to each other. Eventually, feature selection is applied to obtain better prediction quality and smaller set of relevant features. On the other hand, the idea of transforming the multivariate time series into feature spaces are common for data mining tasks like classification. This is often denoted as feature extraction. Similar to feature selection, it is performed by extracting the relevant and non-redundant information from the time series datasets. Overall, the topic of multivariate correlation analysis is of paramount importance for both feature selection and extraction tasks.

The main theme of this dissertation focuses on multivariate correlation analysis on different data types and we identify and define various research gaps in the same. For the defined research gaps we develop novel techniques that address relevance of features to the target and redundancy of features amidst themselves. Our techniques aim at handling homogeneous data, i.e., only continuous or categorical features, mixed data, i.e., continuous and categorical features, and time series.

Multiple views of the feature space exhibit different interactions between features and the target. Harnessing these interactions for the selection of relevant subsets may enrich the prediction model with novel information. Nevertheless, several existing feature selection algorithms focus on obtaining a single projection of the features and are not able to exploit the multiple local interactions from different feature subsets. In such datasets, few features by itself can have a small correlation with the target, but by combining these features with other features, they can be strongly correlated. Hence, it is necessary to evaluate the relevance of a feature based on its higher-order interactions in the dataset. By computing pairwise correlations, several existing works fail to address higher-order interactions between more than two features. For feature extraction in time series applications, the correlation analysis is performed without changing the inherent ordering of the

data. Hence, in addition to dimensionality, it demands extraction and evaluation of a high number of subsequences for feature extraction. That is, it is necessary to simultaneously extract relevant and novel multivariate subsequences and transform them into features. However, traditional feature transformation approaches are often unsupervised or require additional post-processing techniques. Addressing all aforementioned problems require novel algorithms that perform large number of complex statistical computations. This hinders the user understanding of multivariate correlations. Existing correlation analysis algorithms and tools provide only feature ranks or scores and the user perceive the algorithms as a black-box technique. Consequently, the final problem we intend to address is enhancing the transparency of multivariate correlation analysis. Hence, in addition to the algorithmic contributions, we aim to enhance the user's understandability of multivariate correlations in a dataset by presenting a novel software framework.

First, we present our algorithm called ***d**iverse **s**ubset **s**election **s**trategy* (DS3) that identifies diverse and complementary views of the dataset. We extend the concept of multiple views to our ***r**elevance **a**nd **r**edundancy* (RaR) ranking framework for mixed datasets which exhibit higher-order interactions. By evaluating the co-occurrence of subsequences in multiple dimensions, our ***ord**inal fea**ex**traction* (*ordex*) algorithm evaluates higher-order interactions in time series applications. Finally, we provide a software ***f**ramework for **ex**ploring and **u**nderstanding **m**ultivariate correlations* (FEXUM), to help users understand and evaluate the multivariate correlations in the data. In addition, this dissertation includes an extensive experimental and theoretical evaluation of the quality and scalability of our approaches with respect to the existing works. Apart from theoretical time complexity analysis, our evaluation methods are two-fold, i.e., we evaluate the proposed algorithms on synthetic and real world data. Overall, our findings show that our proposed contributions enhance the prediction accuracy and efficiency in comparison to several traditional approaches.

Zusammenfassung

In vielen Anwendungsbereichen werden heutzutage zahlreiche Prozessvariablen, auch Features genannt, gesammelt. Dieser hochdimensionale Feature-Raum wird gemeinhin für analytische Aufgaben wie Klassifikation oder Regression genutzt. Dabei ist es wichtig, diejenigen Informationen zu extrahieren, die für die zugrundeliegende Aufgabe relevant sind. Sowohl für Feature-Selektion als auch für Extraktionsaufgaben ist das Thema der multivariaten Korrelationsanalyse von besonderer Wichtigkeit.

Das primäre Thema der vorliegenden Arbeit ist die multivariate Korrelationsanalyse von verschiedenen Datentypen. In dieser Arbeit werden die bestehenden Lücken im Feld der multivariaten Korrelationsanalyse identifiziert, analysiert und aufgefüllt. Dazu wurden mehrere neuartige Techniken entwickelt, um die Korrelation der Features zu einem Target (d.h. die Relevanz) und die Korrelation der Features untereinander (d.h. die Redundanz) zu untersuchen, und zwar für verschiedene Datentypen wie kontinuierliche und kategorische Daten sowie Zeitreihen.

Bei verschiedenen Blickwinkeln auf den mehrdimensionalen Feature-Raum zeigen sich unterschiedliche Wechselwirkungen zwischen den Features und dem Target. Die Ausnutzung dieser Wechselwrikungen verspricht eine Verbesserung der Prognosemodelle anhand dieser neuartigen Informationen. Einige existierende Feature-Selektionsalgorithmen konzentrieren sich darauf, eine einzige Projektion der Features durchzuführen. Sie sind daher nicht in der Lage, die vielen lokalen Wechselwirkungen der verschiedenen Feature-Subsets auszunutzen. In solchen Datensätzen zeigen einzeln betrachtete Features nur geringe Korrelationen zum Target. Durch Kombination mit weiteren Features können jedoch starke Korrelationen zum Target zutage treten. Daher ist es notwendig, die Relevanz eines Features unter Berücksichtigung seiner Wechselwirkungen höherer Ordnung zu beurteilen. Auch das Berechnen der paarweisen Korrelation, wie es von einigen Algorithmen praktiziert wird, lässt keine Beurteilung der Wechselwirkungen höherer Ordnung zwischen mehr als zwei Features zu. Zusätzlich zur Dimensionalität muss aus einem Zeitreihen-Datensatz eine große Anzahl von Teilreihen extrahiert und ausgewertet werden. Dazu wird ein effizientes Framework benötigt, das es erlaubt, gle-

ichzeitig relevante und neuartige multivariate Teilreihen zu extrahieren und in Features zu transformieren. Traditionelle Ansätze der Feature-Transformation sind oft unüberwacht oder erfordern zusätzliche Nachbearbeitung.

Um alle erwähnten Probleme zu behandeln bedarf es eines neuartigen Frameworks, das aufwändige statistische Berechnungen anstellt. Dies erschwert das Verständnis der Benutzer für die komplexen multivariaten Korrelationen. Deshalb soll als letztes Problem die Transparenz der multivariaten Korrelationsanalyse verbessert werden. So wird zusätzlich zur Algorithmenentwicklung auch das Verständnis des Benutzers für die multivariate Korrelationsanalyse durch ein neues Software-Framework verbessert.

Zunächst wird der Algorithmus „diverse subset selection strategy (DS3)“ vorgestellt, der die verschiedenen Blickwinkel auf den mehrdimensionalen Datensatz identifiziert. Dieses Konzept der verschiedenen Blickwinkel wird auf das „relevance and redundancy (RaR)“ Ranking-Framework erweitert, das gemischte Datensätze mit Wechselwirkungen höherer Ordnung untersucht. Der “ordinal feature extraction (ordex)” Algorithmus untersucht Wechselwirkungen höherer Ordnung in Zeitreihenanalysen durch Auswertung der Kookkurrenz von Mustern in mehreren Dimensionen. Zuletzt wird das „framework for exploring and understanding multivariate correlations (FEXUM)” vorgestellt, das es Benutzern erlaubt, multivariate Korrelationen in den Daten zu verstehen und zu beurteilen. Zusätzlich enthält diese Dissertation ausführliche experimentelle und theoretische Vergleiche hinsichtlich Qualität und Skalierbarkeit der vorgestellten Ansätze zu bestehenden Arbeiten. Abgesehen von der theoretischen Komplexitätsanalyse werden zwei Evaluationsmethoden angewandt. Dazu werden die vorgestellten Algorithmwen sowohl mit synthetischen als auch echten Daten bewertet. Insgesamt wird gezeigt, dass die vorgestellten Methoden die Vorhersagegenauigkeit und –Effizienz gegenüber vielen traditionellen Methoden verbessern.

Acknowledgements

I express my earnest regards to my advisor Prof. Dr. Emmanuel Mueller at University of Bonn, who provided me the opportunity to be a part of his research group. Prof. Mueller provided me the chance to pursue my research interests. His interactive feedback has always been of great help and thought provoking.

I thank Prof. Dr. Eyke Huellermeier for his valuable comments to enhance the theoretical aspects of this work. The feedback from him and his chair was greatly helpful in improving this work.

I thank Dr. Patricia Iglesias Sanchez for her valuable support and discussions as a mentor. Her review comments on the various publications were highly constructive and always helped me to improve. A special thanks to Dr. Jens Thurso and Mr. Klaus Gönner (Bosch GmbH, PS-IG/ENS management) for their generous travel fundings and organizational support. I greatly respect and value the exchange from various data scientists, data providers and engineers at Robert Bosch GmbH. I thank Dr. Daniel Zander and Mr. Erik Van Winkle for their prudent comments as a proof-reader.

Most of the research approaches developed in this thesis are the result of team work. I want to thank everyone who worked together with me. Also many thanks to my graduands and bachelor project team for their support in the implementation of some of the solutions in this thesis. I also thank my colleagues at Hasso-Plattner Institute and University of Bonn for the interesting discussions and valuable comments that helped me to learn and improve.

I thank Prof. Dr. Felix Naumann, Prof. Dr. Stefan Kurz, Dr. Michael Hackner, Dr. Carsten Kopp and Dr. Nadine Jung for all their encouragement and support. Last but not the least, I want to especially thank my family for having supported me during all these times. In particular, I want to thank my mother for all her moral support and motivation.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

AI	Artificial Intelligence
ANN	Artificial Neural Networks
CFS	Correlation-based Feature Selector
CG	Cumulative Gain
CMIM	Conditional Mutual Information Maximization
CPU	Central Processing unit
dCor	Distance Correlation
DS3	Diverse Subset Selection Strategy
DTW	Dynamic Time Warping
ECG	Electrocardiography
FCBF	Fast Correlation-Based Filter
FEXUM	Framework for exploring and understanding the multivariate correlations
FFT	Fast Fourier Transform
GA	Genetic Algorithms
GP	Gaussian Processes
GPU	Graphical Processing Unit
H	Shannon entropy
HCTSA	Highly Comparative Time series Analysis
HiCS	High Contrast Subspaces

IG	Information Gain
IoT	Internet-of-Things
JMI	Joint Mutual Information
KDD	Knowledge Discovery in Databases
KLD	Kullback–Leibler Divergence
KNN	K-Nearest Neighbors
LSTM	Long Short-Term Memory
MAC	Multivariate Maximal Correlation Analysis
MCMR	Maximum Correlation and Minimum Redundancy
MFS	Mixed Feature Selection
MI	Mutual Information
MIC	Maximal Information Criterion
mRmR	maximum Relevance minimum Redundancy
mRW	modified Relief Weight
NRMSE	Normalized Root Mean Square Error
OLS	Ordinary Least Squares
Ordex	Ordinal feature extraction
PCA	Principal Component Analysis
PSO	Particle Swarm Optimization
RaR	Relevance and Redundancy
SAX	Symbolic aggregate approximation
SBE	Sequential Backward Elimination
SFFS	Sequential Forward Floating Selection
SFS	Sequential Forward Selection
SU	Symmetric Uncertainty
SV	Shapley value
SVM	Support Vector Machines

Feature selection notations

f	Feature
Y	Target
$\mathcal{F}$	Feature space
d	Number of features in a feature space
div	Divergence function
N	Number of instances or samples in a dataset
$\mathcal{F}_N$	Set of categorical features
$\mathcal{F}_C$	Set of continuous features
$\mathcal{D}$	Dataset comprising of feature space and target, i.e, $\mathcal{F} \cup Y$
$corr(f, Y)$	Correlation of a feature to target, also denoted as $corr(f)$
S	Feature subset

Feature extraction notations

X	A univariate time series
t	Time series index
T	A multivariate time series sample
m	Dimensionality of a multivariate time series sample
D	A time series dataset
n	Number of time series samples in a dataset
s	An ordinal pattern
$\mathcal{S}$	A set of ordinal patterns
$\not{d}$	Degree of an ordinal pattern

Chapter 1

Thesis Overview

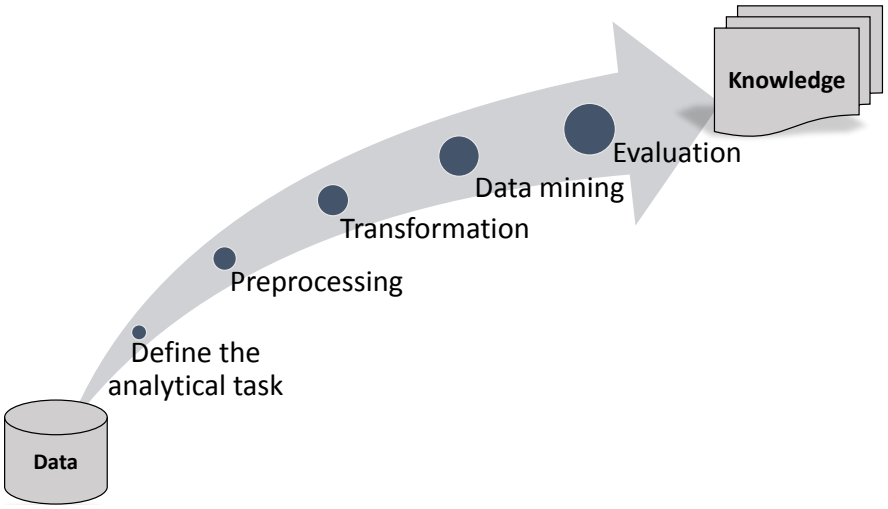


Figure 1.1: The task of training a machine learning model [FPSS96]

1.1 From Data to Knowledge

In the modern era of Internet-of-Things (IoT), data from a plethora of sensors are readily accessible. With the availability of such vast amounts of data in various applications such as bio-informatics, automotive, finance, media and medical, the first question that arises is: *How to extract useful knowledge from the data*? The Knowledge Discovery in Databases (KDD) process delineates the different steps for acquiring valuable knowledge from the data [FPSS96]. As shown in Figure 1.1, this process involves five major steps. The first step is understanding the domain and defining the analytical task. Having fixed a tangible goal, the second step involves data preprocessing, i.e., removal of noisy data and treatment of missing values. The dimensionality of the preprocessed data is reduced at the transformation step by selection or extraction of information that are relevant to the analytical task. *This supervised transformation phase of the KDD process is the primary focus of the dissertation.* Finally, the transformed or reduced data is used as an input for various data mining algorithms, e.g., classification and regression, and the results are evaluated for obtaining useful knowledge from the data.

As we focus on supervised transformation, our predominant application will be prediction systems with a target. The KDD process supports in systematic building of such prediction systems and we provide an example from our application. In automotive domain, data from hundreds of sensors in the car and driving char-

acteristics are used to predict the health of an automotive fuel system [SdSFS18]. In this case, the health of the automotive fuel system serves as a target vector that is to be predicted. Using the data from various sensors as features and the target, a concrete analytical objective is defined for the prediction system, i.e., step one of the KDD process. Typically, sensor data contain a lot of incomplete and noisy information. The data is thus cleaned and structured, i.e., the preprocessing step of the KDD process. However, not all sensors are influenced by the health-state of a fuel system. Hence, relevant information like fuel quantity and temperature are selected from the cleaned data, i.e., the transformation step of the KDD process. As a supervised learning task, the transformed data and the target vector representing the health-state of fuel system are fed into the classification or regression algorithms. The algorithm learns the latent function, between the transformed data and the target vector, i.e., the data mining step of the KDD process. The results are evaluated by domain experts to gain insights on the health-state of fuel system, i.e., evaluation step of the KDD process.

The transformation step is one of the most time consuming in the entire KDD process chain [FPSS96, ON14]. Selection of relevant information prior to application of classification or regression algorithms have several advantages such as:

- Reduced dimensionality [SSM17, SBS$^+$17].
- Increased efficiency and accuracy of prediction systems [MBN02, YL03].
- Enhanced user understandability of prediction systems [DP05].
- Reduced susceptibility to over-fitting [GE03].
- Reduced storage and measurement cost [HS98].

In short, problem of selecting information relevant to the target, i.e., transformation step, is of great importance for prediction systems. Hence, this work targets on qualitative and quantitative enhancements in the transformation step. Moreover, this dissertation will substantiate the need for such enhancements and introduce novel algorithmic frameworks for the same.

1.2 Goals

Feature selection and extraction are the pivotal components of the transformation step. The main goal of multivariate feature selection and extraction is to provide a small and predictive subset of features based on its correlation with the target. They are proven to be useful in several application domains [HLY08, SBS+17, RSA+18, SdSFS18] and has therefore been an extensively researched topic in the data mining community [Qui14, NAM01, GE03, RŠK03, WSH06, KMB12]. However, there are multiple open research questions that enhance the selection and extraction processes. In this section, we motivate these research questions and introduce our goals using examples from various application domains such as automotive, medical, marketing, aerospace, economical analysis and bio-informatics. However, they are not limited to these application domains. Following the summary of thesis goals, in the next section, we elaborate the challenges that we are confronted with. Finally, we provide a summary of the contributions we present in this work.

Multi-view

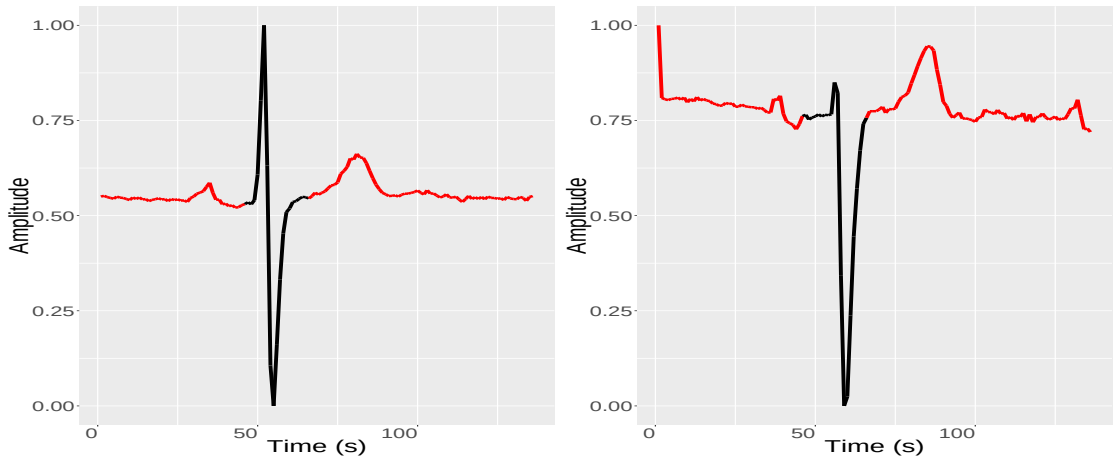
In large datasets that contain features acquired from multiple sensor sources, different feature combinations, i.e, multiple views, can exhibit different type of correlation with the target [SSM17]. For example, in automotive domain, a wide range of sensors such as pressure sensor, thermistor and potentiometer are used for data acquisition. Due to the heterogeneity of the sources, different feature subsets interact differently with the target. In other words, each feature subset shows a different type of correlation with the target. For example, the features representing the air system of an automobile correlates differently to the target when compared to the features representing the fuel system. Selecting such multiple views of the dataset can improve the prediction quality in comparison to selection of a single large feature subset. This necessitates an algorithm to exploit the correlations in multiple views of the data. Such a framework acknowledges the local interactions in the high-dimensional feature space to enrich the prediction model and improve its accuracy. Hence, we aim to systematically generate and evaluate multiple views of the feature space to capture these local interactions.

Higher-order interactions

In dependency-oriented datasets, i.e., datasets that exhibit multivariate correlations with the target, individual features may show a low magnitude of relevance to the target. These individual features in combination with other features in the dataset can be strongly relevant for the target prediction. This means that multiple features exhibit higher-order interactions among themselves. Performing bivariate correlation analysis in such cases can lead to distorted knowledge about the feature's relevance [SBS+17]. Hence, it is necessary to estimate its relevance based on the higher-order interactions for better prediction accuracy. Let us assume the task of identifying features that correlate with the health of a particular component in an aircraft. In such a system, features representing the individual components and environmental conditions exhibit higher-order interactions [RSA+18]. Hence, evaluating the relevance of a feature without analyzing its interactions with the other features is misleading. We aim to develop an algorithm that scores a feature's relevance based on its interaction with several other feature combinations.

Mixed data types

Several datasets from real world applications such as medical, marketing and economical analysis contain different data types, i.e., continuous and categorical [HLY08]. For example, let us assume a categorical variable $Nationality$ with three different categories $\{Indian, German, French\}$. During the analysis of multivariate correlations, a subset of features can have a mix of both categorical and continuous features. In such scenarios, the categories cannot be treated as numerical values because each state or category denotes a qualitative property. This implies that their relevance for predicting the target needs to be evaluated differently in comparison to the continuous feature values. However, evaluation of continuous and categorical features with different correlation functions can be problematic because they are not directly comparable to each other [TM07]. Hence, our goal is to evaluate the relevance of a mixed feature subset based on a single criterion function that is not affected by the data type.



(a) A time series representing class 1

(b) A time series representing class 2

Figure 1.2: Time series representing two classes, where the subsequence of interest is highlighted in black

Multivariate time series

Similar to continuous and categorical features, time series is yet another data type that is prevalent in several applications [WSH06, WWW07, YK09]. A time series is a sequence of numerical values indexed by time. That is, the order of a sequence should not be altered during correlation analysis. For example, Figure 1.2 shows the ECG signals representing two different classes [CKH+15, YK09]. Each of them have a subsequence of the time series highlighted in black. These subsequences are characteristic or unique for that particular class. Correlation analysis on time series data involves identifying these subsequences that are relevant to target and encoding them into features for performing predictions. In a multivariate time series, subsequences from multiple dimensions interact to be discriminative for the prediction task. In other words, they exhibit higher-order interactions, i.e., a single subsequence form a dimension may be less relevant in comparison to a collection of subsequences from multiple dimensions. For example, in stock market data, subsequences from multiple time series can be used collectively to predict bullish or bearish markets [KvD10]. In such applications, it is crucial to evaluate the co-occurrences of subsequences from different companies for better predictions [AEG14]. Hence, our goal is to evaluate such higher-order interactions in a multivariate time series dataset.

Redundancy

As discussed above, feature selection evaluates the feature vs. target correlations. However, two relevant features can also be redundant to each other. For example, in bio-informatics applications, thousands of features pertaining to genetic information are used to classify cancer tissues from normal tissues. Several features in such high-dimensional feature space render similar information, i.e., they exhibit correlation amongst themselves [DP05, SBS+17]. These redundant features affect the efficiency of the prediction model, i.e., larger training time [MBN02, YL03]. In the KDD process depicted in Figure 1.1, the transformation step precedes the classification or regression task. Hence, it is ideal to eliminate the redundancy prior to training a machine learning model. This dissertation will aim to address the problem of information redundancy for feature selection and extraction to enhance the generalization ability of the prediction model [PLD05, DP05].

Understanding the correlations

Feature selection involves evaluation of a large number of feature combinations. The complex statistical tests and large number of evaluations deter the user's understanding of feature selection. Hence, the domain experts look at feature selection as a black-box technique [KRT+17]. Nevertheless, the domain experts perform the first and most important step of defining the analytical task in the KDD process (c.f. Figure 1.1) chain. Hence, it is necessary for the experts to understand the reason for a feature's relevance to determine whether a correlation is merely a statistical coincidence or a general dependency that influences the target. For example, in automotive applications few features can exhibit high correlation to the target due to the measurement technique applied. That is, they are merely statistical coincidences that can happen due to the measurement system and is not founded by the physics of an automotive device. As a step towards enhancement of feature selection transparency, we aim to visualize all correlations in a dataset, i.e., feature-target relevance and feature-feature redundancy. Secondly, we aim to guide the user in understanding these multivariate correlations. In Figure 1.3 we illustrate a glimpse of all the goals that we will address in this work.

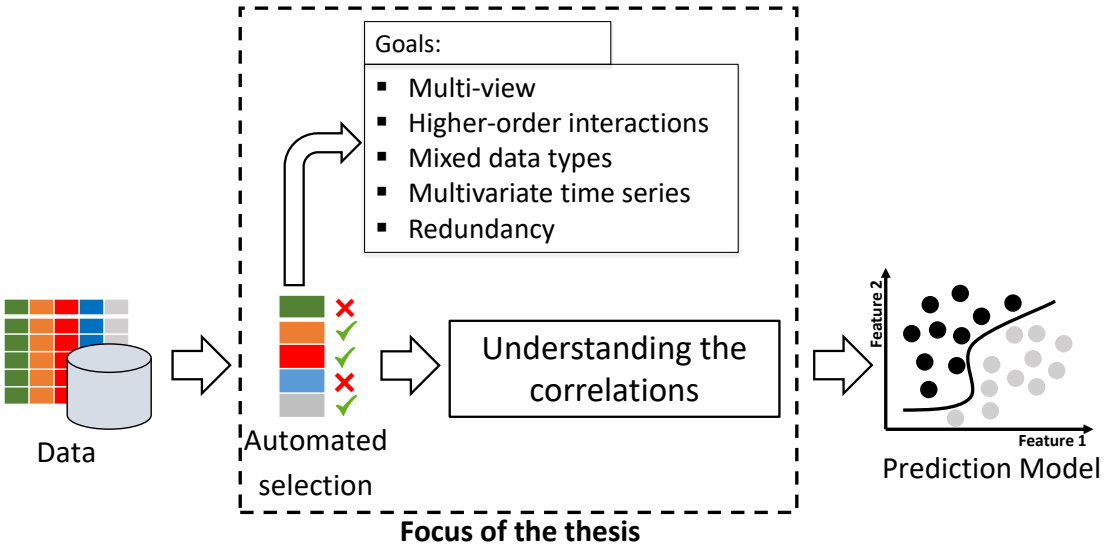


Figure 1.3: Focus of the dissertation

1.3 Challenges

In Section 1.2, we discussed our goals such as multiple-views of data, higher-order interactions, mixed data types, multivariate time series feature extraction, redundancy and user understandable correlation analysis that we will address in this dissertation. However, in the process of achieving these goals, we are confronted with several challenges and in this section we briefly discuss them.

Challenge 1: Exponential Search Space

The estimation of multivariate correlations involve analyzing multiple feature subset combinations. For a dataset with 200 features, the number of possible feature subset combinations escalates to 2^{200}. In real world applications, the dimensionality can be much larger and analyzing the relevance of every possible combination is computationally inefficient. Traditional approaches handle this problem by designing efficient search organization techniques [DV11, BALD14]. However, the challenge is not only exploring the exponential search space but also to infer maximum knowledge of the higher-order interactions while evaluating the subsets. Hence, it requires an ideal combination of a search organization and correlation inferring technique.

Challenge 2: Multi-view

Considering multiple views (subsets) of the high-dimensional feature space for the evaluation of correlations may enhance the prediction quality by capturing the interactions between different dimensions. Traditional selection algorithms [KJ97, MBN02, YL03, LMD+12] aim to select a single feature subset by evaluating the correlation based on a single statistical property. However, evaluating multiple statistical properties in a dataset can enhance the prediction quality. Two major challenges exists in the process of selecting multiple views of a dataset. The first challenge is to select the views that are relevant for a given task. Secondly, multiple views that capture the same dependencies will not provide any novel information for the prediction task. Hence, it is necessary to ensure that the multiple views are complementary to each other. Overall, it is still an open research question on how to exploit the heterogeneity of the feature dependencies in multiple feature subsets for improving the prediction quality.

Challenge 3: Higher-order interactions

In complex systems, an individual feature exhibiting higher-order interactions influences the target differently when combined with different feature combinations [SBS+17, KRT+17]. In such cases, assigning a feature's importance without evaluating its higher-order interactions is misleading. Hence, it is necessary to assess the role of a feature by analyzing its interactions in multiple subsets. However, to include the higher-order interactions in high-dimensional datasets, it is not efficient to perform an exhaustive search [JBB15]. Traditional approaches [PLD05, Hal00] are based on pairwise correlation analysis and fails to capture interactions between more than two features. Thus, the challenge is to efficiently obtain a reasonable estimate of the feature relevance by evaluating its interactions in multiple subsets.

Challenge 4: Redundancy

In addition to estimating a feature's correlation to the target, it is also necessary to evaluate its non-redundancy with respect to other features in a subset. This ensures that each feature is not only relevant but also provides new information for a classification or regression algorithm. Experimental and empirical analysis from several feature selection literatures prove that removal of redundant features enhance the speed and accuracy of classification and regression algorithms

[Hal00, KJ97, YL03, PLD05, DP05]. As redundancy does not only imply that two features are identical, the challenge is to quantify the magnitude of novelty that a feature contributes for the prediction task without actually training a prediction model.

Challenge 5: Mixed datasets

Modern datasets usually contain a mixture of continuous and categorical data [SBS+17]. Traditional approaches transform the dataset with mixed data types into homogeneous data type by discretization of the continuous features [Hal99]. Such transformation techniques avoid the necessity to treat each data type differently. However, it leads to information loss and the effectiveness of the selected features is strongly influenced by the discretization method employed [JS02]. The other naïve preprocessing step is to encode the categories with numerical values. Such encodings are not meaningful because the categories represent a qualitative property and assigning random numbers can be misleading, i.e., computing distances between two differently coded categorical features versus the target can show different results [HLY08]. Hence, the major challenge is to perform multivariate relevance and redundancy estimation of large datasets without the need for such additional preprocessing (i.e., discretization and encoding) techniques.

Challenge 6: Multivariate time series correlation

The transformation of time series into static features is a prevalent concept in the literature. However, several existing approaches [NAM01, Mör03, WWW07, FJ14] do not address the multivariate nature of the time series. That is, by performing univariate transformations on multiple dimensions they fail to encode the multivariate interactions of the time series into the resulting features. As explained in Section 1.2, in lengthy time series, it is only a subsequence that is informative for the prediction task. In a multivariate time series, by including the co-occurrences of subsequences from multiple dimensions, there exists an exponentially growing number of subsequence combinations to evaluate. However, not all of them are relevant for the target and non-redundant to the already selected subsequences. In such a scenario, traditional approaches perform unsupervised feature transformation [NAM01, WSH06, WWW07, LKL12, Kat16] and may lead to generation of irrelevant and redundant features. Hence, the challenge lies in identifying the

relevance of time series subsequences by evaluating its multivariate nature and non-redundancy.

Challenge 7: Efficiency

Both feature selection and extraction involve analysis of an exponentially growing number of feature subset or subsequence combinations. Upon this, several application domains incorporate new sensors and collect more data from existing sensors [LP03]. This directly leads to increasing dimensionality and database size. Regardless of the increasing volume of data, it is necessary to perform the correlation analysis efficiently. For this reason several feature selection and extraction algorithms focus on the algorithmic efficiency in addition to quality [MBN02, Fle04, YL03, BPZL12]. The computational efficiency of the correlation function and the search organization technique together influence the runtime of the feature selection algorithms. Hence, the challenge is to compute the feature dependencies with efficient correlation functions and search space exploration techniques that are scalable.

Challenge 8: Understanding multivariate correlations

To inspect the results of a feature selection algorithm, it is necessary for domain-experts to understand the algorithm. Conventional selection algorithms [KJ97, MBN02, YL03, LMD+12, DP05, PLD05] provide only a set of highly correlated features and do not show a summary of all correlations in a high-dimensional dataset. Though a few approaches aid in the understanding of bivariate correlations, difficulty arises when trying to comprehend the dependencies between more than two features. Overall, the dimensionality and the complex dependencies in the data impair an expert's understanding of the correlations. Hence, making multivariate correlation analysis as a transparent process is still an unresolved problem. In high-dimensional datasets, there are both feature-to-target (i.e., relevance) and feature-to-feature dependencies (i.e., redundancy). For a dataset with hundreds of features, the major challenge is to visualize both the dependencies and explain the detected multivariate correlations to domain-experts.

1.4 Contributions

This dissertation presents four major contributions to address the goals described in Section 1.2.

Diverse selection of feature subsets: In Chapter 3, we introduce a novel algorithmic framework to identify multiple diverse views of the high-dimensional feature space. Section 1.2 explains the necessity and Section 1.3 provides a glimpse of the challenges involved while capturing such views of datasets with features that are acquired from different sensor sources (c.f. Challenge 1 and 2). Our framework tackles these challenges using multiple correlation measures to evaluate different types of dependencies in the data. Additionally, based on a diversity criterion, we enhance the diversity of multiple views to ensure complementary information in each view. Conceptually, the proposed solution falls into the category of hybrid selection, i.e., a mix of filter and wrapper paradigms.

Relevance and redundancy ranking: In Chapter 4, we propose the first feature ranking framework to compute a single score that quantifies the feature relevance by considering the higher-order interactions between features and redundancy in mixed datasets. In comparison to the previous contribution, which is limited to continuous features, we broaden the concept of multiple views to mixed datasets as well. In addition, to enhance the efficiency in comparison to the previous contribution, we adhere to the filter-based paradigm for the feature ranking framework. We accomplish an efficient methodology of feature scoring by considering the feature's influence in multiple data views and novelty. Hence, the top ranked features are characterized by maximum relevance and non-redundancy (c.f. Challenge 1, 3 to 5 and 7).

Relevant and non-redundant feature extraction for time series: This dissertation tenders the first feature extraction framework that encodes the multivariate nature of ordinality in the time series into static features. As a supervised approach, our extraction scheme in Chapter 5 concurrently extracts and evaluates the correlations, i.e., relevance and redundancy of the extracted feature. Hence,

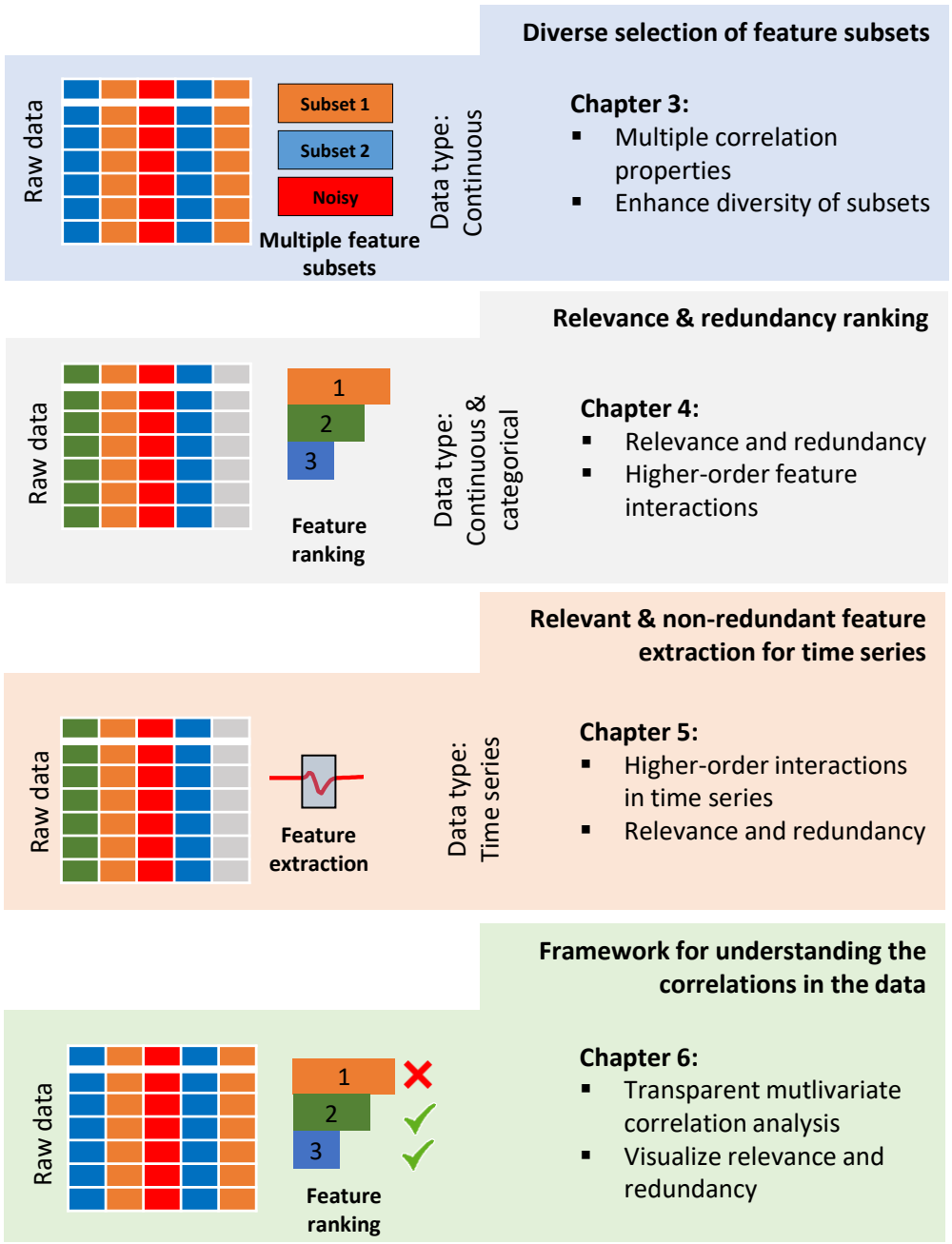


Figure 1.4: Overview of thesis contributions

the extracted features are not only relevant but also hold novel information. Thus, we further augment the concept of higher-order interactions and redundancy to feature extraction in time series applications (c.f. Challenge 6). By performing the evaluations in linear time complexity, our relevance scoring function is efficient (c.f. Challenge 7).

Understanding the correlations in the data: In Chapter 6, this dissertation proposes an interactive software framework for exploring and understanding multivariate correlations. With this framework we bolster our previous contributions by aiding the user to understand feature selection. The interactive framework aims to enhance the transparency of feature selection by presenting the complex correlation analysis calculations in a user-understandable way. To gain novel insights into the data, we provide a summary of all correlations in the feature space. Additionally, using various statistical visualization techniques, we guide the user in comprehending multivariate correlations (c.f. Challenge 8).

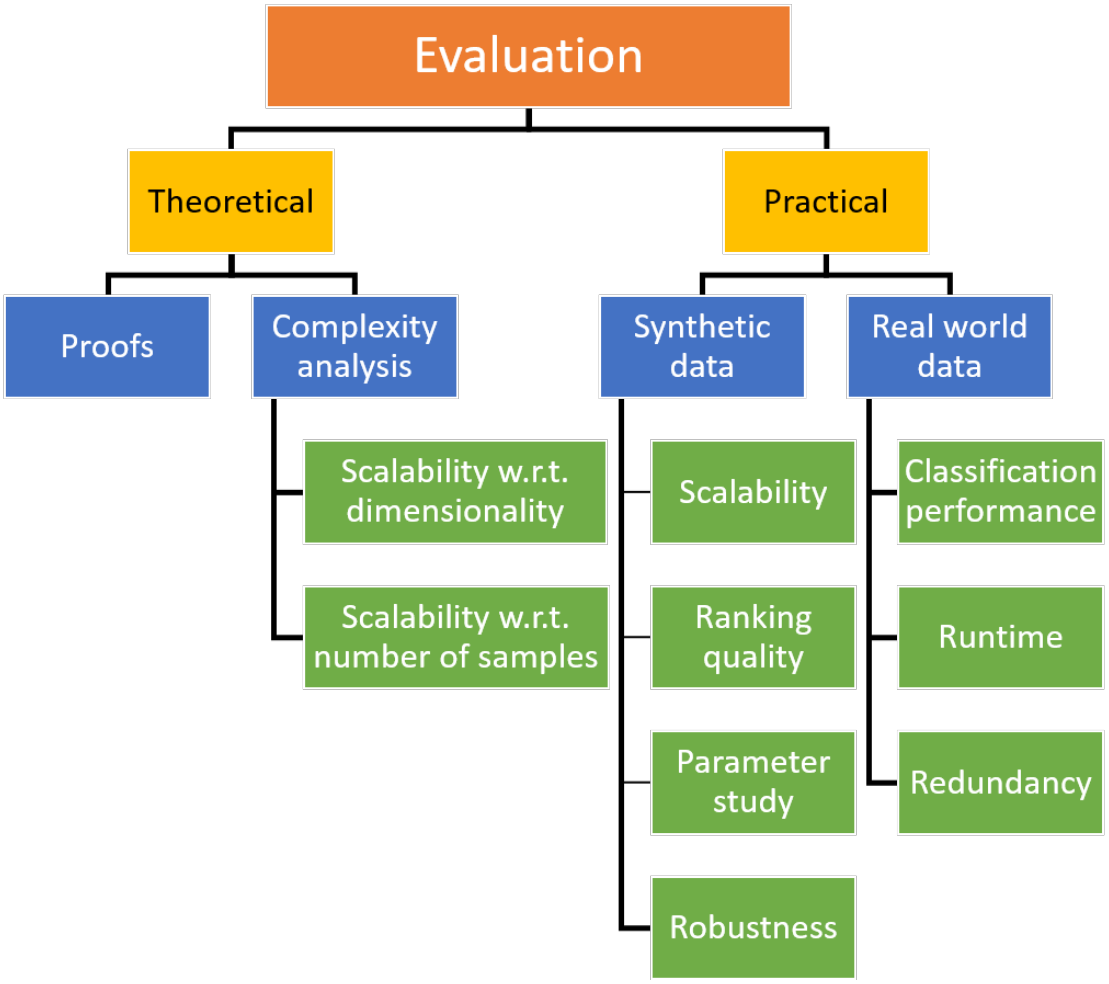


Figure 1.5: Evaluation strategy followed for all algorithmic contributions in this dissertation

In Figure 1.4, we summarize the major contributions of this dissertation. To emphasize the qualitative and quantitative improvements, our algorithmic con-

tributions are evaluated based on theoretical time complexity and practically on synthetic and real world datasets. Our contributions are substantiated with theoretical proofs and foundations whenever necessary. Using synthetically generated datasets, we analyze the scalability, ranking quality, parameters and robustness of our approach. As we do not know the ground truth in real world datasets, we evaluate the relevance and redundancy based on the classifier accuracy. In addition, we also tabulate the runtime our algorithms on real world dataset. All practical evaluation results are compared with a variety of state-of-the-art techniques from different paradigms. A consolidated overview of the evaluation strategy we follow in this dissertation is shown in Figure 1.5. To obtain finer details about each of the aforementioned contributions, it is necessary to revisit the preliminary concepts and the existing works on correlation analysis. Hence, in Chapter 2, we begin with the basic concepts that lay the foundation for multivariate correlation analysis. This also aims to ensure that the contents of the dissertation are self-contained. Furthermore, we also identify and discuss the novelty of our work with respect to the state-of-the-art approaches.

Chapter 2

Fundamentals and literature overview

2.1 Fundamentals

2.1.1 Feature selection

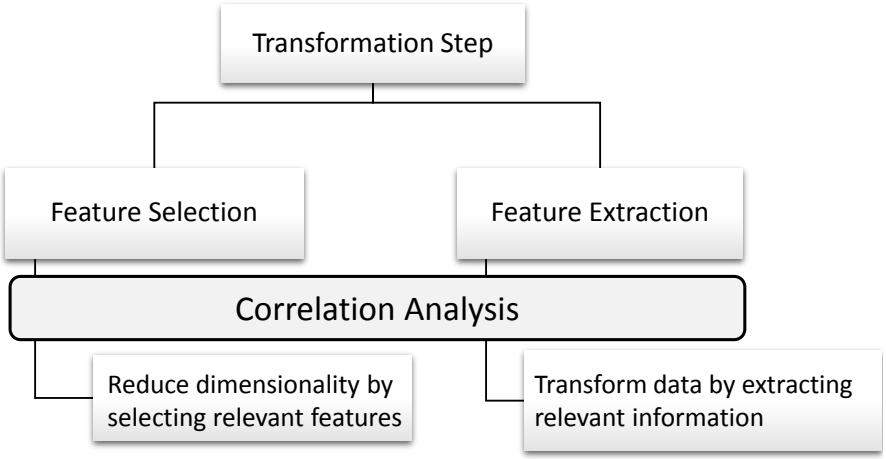


Figure 2.1: The transformation step of the KDD process

Feature selection and feature extraction are the two crucial components of the transformation step in the KDD process (c.f. Figure 2.1). Performing data mining tasks on high-dimensional data hampers the quality and efficiency of the prediction task due to the *curse-of-dimensionality* [Pow07, KMB12, Agg15]. As represented in Figure 2.2, with a growing number of dimensions, the prediction accuracy increases up to a maximum. Beyond this point, the accuracy of a prediction model declines due to the curse-of-dimensionality [JGDE08].

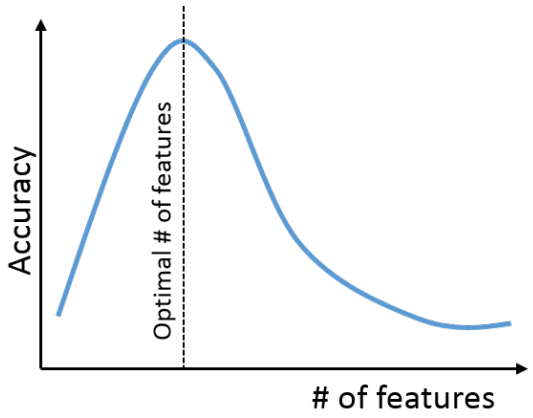


Figure 2.2: With increasing dimensionality of data, the prediction accuracy drops after a threshold number of features[JGDE08]

In other words, using a large number of features does not guarantee the best prediction accuracy. Consequently, feature selection is necessary to eliminate features that are non-contributing for a defined analytical problem. For better understanding, we explain the *curse-of-dimensionality* with a simple example [Spr14].

Example 2.1. Let us consider the task of classifying cars from trucks based on a three-dimensional feature space. Using a linear classifier, in Figure 2.3, we show that a single feature cannot perfectly classify all the cars from trucks.

Figure 2.3: Visual representation of the two classes, i.e., car and truck, over a single feature

Hence, we add an additional dimension (feature 2) in Figure 2.4. Addition of second dimension still does not allow for perfect classification of all samples in the data. However, by classifying 80% of the data correctly, it performs better than using a single feature.

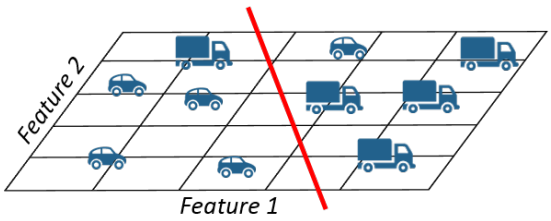


Figure 2.4: Classification task with two features, where the red line denotes the classification boundary

To improve the classification accuracy further, we add a third dimension in Figure 2.5 and show a linear plane that perfectly separates all samples of trucks from cars. That is, a linear combination of features 1, 2 and 3 is able to classify all training samples without error. From the first look, it is alluring to conclude that addition of more features lead to better accuracy. However, this can be deceptive and we explain the underlying reason by computing the density of the data.

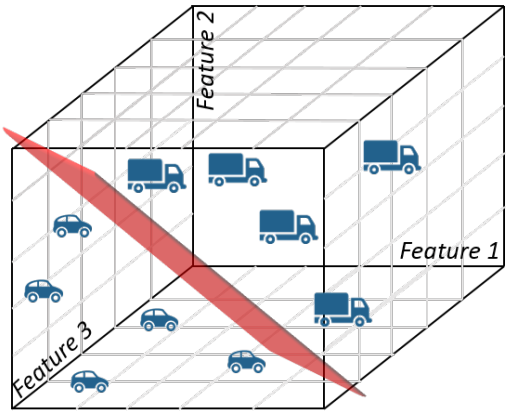


Figure 2.5: Classification task with 3 features, where the red plane denotes the classification boundary

In the one-dimensional case (c.f. Figure 2.3), ten samples are spread across an interval of 5 units of feature 1. Therefore, the density of samples per unit distance is $\frac{10}{5} = 2$ samples/unit. Analogously, the density of samples for the two and three-dimensional cases are $\frac{10}{5^2} = 0.4$ samples/unit and $\frac{10}{5^3} = 0.08$ samples/unit respectively. From the above example, we show that density of samples reduces (i.e., data gets sparser) with increasing number of dimensions. Eventually, the task of identifying a separable plane on the sparse feature space is easier (see Figure 2.6, where the 3-dimensional feature space is projected onto a 2-dimensional feature space). Hence, the classification algorithm memorizes the training data and this problem directly leads to overfitting.

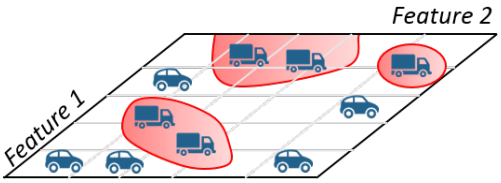
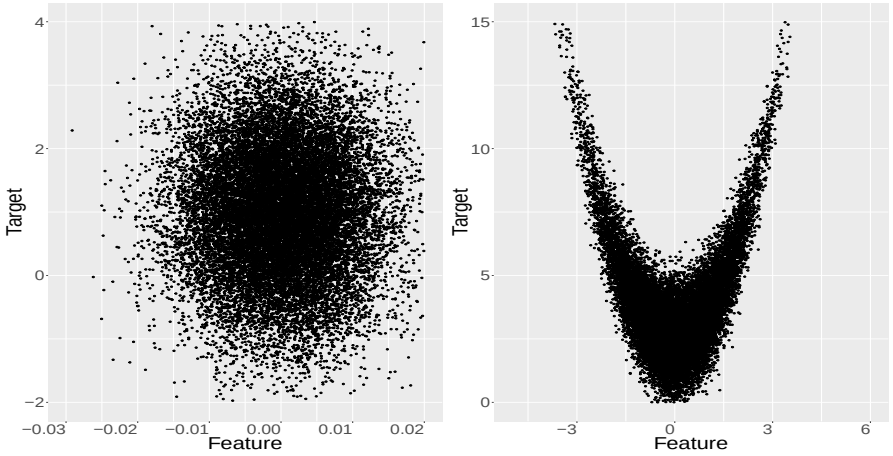


Figure 2.6: Projection of the three-dimensional feature space to two dimensions, where the red line denotes the classification boundary

Testing the over-fitted classification model on unseen real world data yields a poor prediction quality. This can also occur on using a complex non-linear classifier on a simple low-dimensional feature space. Hence, in Example 2.1, it is ideal to use a combination of feature 1 and 2 as it can generalize better for unseen real world data. A possible solution to resolve this problem is by identifying the relevance of each feature based on its potential to discriminate the two vehicle

categories, prior to training a prediction model. This process is called feature selection.

> **Quote**: "Features are relevant if their values vary systematically with the target" [GLF89].



(a) An irrelevant feature (b) A relevant feature

Figure 2.7: Correlation of the features to the target [KMB12]

Feature selection is an expedient step to select the relevant features prior to training supervised machine learning algorithms such as classification and regression. Given a prediction target (e.g., cars and trucks in Example 2.1), an irrelevant feature corroborates no correlation with the target (c.f. Figure 2.7a) and distorts the accuracy of the prediction model [GE03]. On contrary, relevant features exhibit predictive relationships (correlation) with the target and improve the prediction quality (c.f. Figure 2.7b). In addition, a subset of relevant features can also be redundant among themselves and they do not provide any novel information for the data mining task [DP05, SBS+17]. This means, in addition to relevance, it is necessary to ensure that the information we provide to the data mining algorithm is complementary. A subset of novel and relevant features are those which poses high feature-target correlations and minimum feature-feature correlations [Hal99].

2.1.2 Feature extraction

The second elementary component of the transformation step is feature extraction. It is the task of extracting parts of relevant information from a time series. The

unique property of time series data is their adherence to inherent ordering with respect to time, i.e., the relevance estimation in time series data is to be done without distorting the order of the data. For example, Figure 1.2 (in Chapter 1) shows the Electrocardiography (ECG) time series data representing two classes [CKH+15]. Out of the entire series, only a subsequence (highlighted in black) is discriminative for distinguishing the two classes. The other parts of the time series (highlighted in red) exhibit similar behavior and cannot be used to discriminate the two series. In time series applications, such regions of interest are extracted and transformed into numerical features [NAM01, Mör03, WWW07, FJ14]. The advantage of feature extraction is the possibility to provide a compact and informative set of features for a given prediction task [GE06]. The extraction is often performed based on a transformation function that evaluates specific properties of the time series and encodes them into features. For example, frequency and amplitude of the time series data are captured using Fast Fourier Transform (FFT) as a transformation function. That is, unlike feature selection, the feature extraction applies transformations on the raw data to encode the information from a region of interest into numerical features. However, similar to feature selection, it is necessary to evaluate the relevance and novelty of the transformed feature for the prediction.

From a high-dimensional dataset, feature selection identifies relevant and non-redundant features for the prediction model. On the other hand, feature extraction aims to extract relevant and non-redundant subsequences from the data and transform them into features. However, relevance and redundancy estimation in both feature selection and extraction is performed by evaluating the correlation between the features and the target. Hence, as depicted in Figure 2.1, *both feature selection and extraction are established upon the concepts of correlation analysis.*

2.1.3 Correlation Analysis

Correlation analysis is the task of evaluating the statistical dependency between a dependent and independent variables. A dependent variable is also denoted as a target and the task of correlation analysis aims to evaluate the influence of different independent features on a dependent feature. On the other hand, an independent variable is called a feature. Based on the number of features involved in the

analysis, it can be classified into bivariate or multivariate correlation analysis. Bivariate correlation analysis is limited to evaluation of relationship between a feature and the target. On contrary, the benefit of multivariate analysis is the possibility to analyze the influence of interactions between multiple features on a target. The choice of analysis can be made based on the type of data [Agg15]:

- **Non-dependency-oriented data**: Being the simplest form of data, the features in the non-dependency-oriented data do not exhibit any dependencies among themselves. Thus, it requires analysis of a single feature's correlation to the target prediction or bivariate correlation analysis.

- **Dependency-oriented data**: In this case, multiple features in the dataset may have certain dependencies or interactions among themselves. Thus, it requires inclusion of these interactions for the target prediction, i.e., multivariate correlation analysis.

Feature selection on high-dimensional dependency-oriented datasets involves evaluation of complex interactions between multiple features. That is, for dependency-oriented datasets, there can be several features that change simultaneously to influence the target prediction. Including these interactions between the features for correlation analysis provide novel insights from the data. Such datasets are common in various application domains such as automotive [SBS+17], aerospace [RSA+18] and bio-informatics [DP05]. Hence, it is essential to estimate the correlation between a set of features and the target. This motivates the importance of using multivariate correlation analysis for the selection process. In this work we focus on the task of multivariate correlation analysis for feature selection and extraction.

Existence of correlation between two variables implies that the change in one variable influences the other as well (c.f. Figure 2.7b). Based on the nature of change, correlations can be classified as linear and non-linear. The correlation between the feature and target in Figure 2.7b is an example of non-linear correlation. Real world applications are predominantly non-linear, hence we focus on non-linear correlations in this work.

2.1.4 Basic Notions

Feature selection: Feature selection and extraction demands correlation or dependency analysis between the features and a target. The term feature is used interchangeably with attribute, dimension and variable in various literatures [NAM01, GE03, RŠK03, WSH06, KMB12, Qui14]. As discussed in Section 1.2, a dataset can have features with different data types. Hence, we begin with the formal definition of a feature and different data types in Definition 2.1.

Definition. 2.1: Feature

- A continuous feature $f = (x_1, \cdots, x_N) \mid x_i \in \mathbb{R}$ of N samples is a vector of real numbers with an infinite number of possible values.
- A categorical feature of N samples is a vector with a fixed number of possible categories $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_{max}\}$. Each value in a categorical feature $f = (x_1, \cdots, x_N) \mid x_i \in \mathcal{C}$ denotes a category based on a qualitative property.

Instance #	Dataset $\mathcal{D}$				
	Feature space $\mathcal{F}$				Target
N	f_1	f_2	f_3	f_4	Y
1	1.2	100	10	True	1
2	2.5	150	20	False	1
3	3.4	275	30	False	1
4	6.99	50	55	True	2
5	2	110	22.5	False	2
6	1.6	25	66	False	2

Table 2.1: Representation of a dataset with mixed data types, where, $\mathcal{F}_C = \{f_1, \cdots, f_3\}$ are continuous features and $\mathcal{F}_N = \{f_4\}$ is a categorical feature

Given a d-dimensional feature space $\mathcal{F} = \{f_1, \cdots, f_d\}$ of N samples, a dataset is a collection of the feature space and a target Y, i.e., $\mathcal{D} = \{\mathcal{F}, Y\}$. A feature space with mixed data types is defined by a set $\mathcal{F}_C \subseteq \mathcal{F}$ of continuous and set $\mathcal{F}_N \subseteq \mathcal{F}$ of categorical features, i.e., $\mathcal{F} = \mathcal{F}_C \cup \mathcal{F}_N$. Table 2.1 shows an example

of a dataset with mixed data types, i.e., continuous and categorical, and a discrete target, i.e., classification task. For a regression task, the target Y will be a column of continuous values.

As discussed in Chapter 1, the goal of feature selection is to eliminate features that are irrelevant for the target prediction by evaluating the magnitude of correlation between the features and the target. Quantifying the correlation is performed using a cost function that evaluates the feature-target dependency. We formally define the task of feature selection in Definition 2.2.

Definition. 2.2: Feature Selection Task

Given a dataset $\mathcal{D}$ such that it contains a feature $f_{rel} \in \mathcal{D}$ relevant to the target Y and an irrelevant feature $f_{irr} \in \mathcal{D}$. Based on the number of features that a cost function can handle, it can be classified into bivariate and multivariate.

- A bivariate correlation measure $corr : (f \in \mathcal{F}) \mapsto \mathbb{R}$ computes the relevance of f such that, $corr(f_{rel}) \gg corr(f_{irr})$.
- A multivariate correlation measure $corr : (S \subseteq \mathcal{F}) \mapsto \mathbb{R}$ computes the relevance of a set of multiple features by including the higher-order interactions between them.

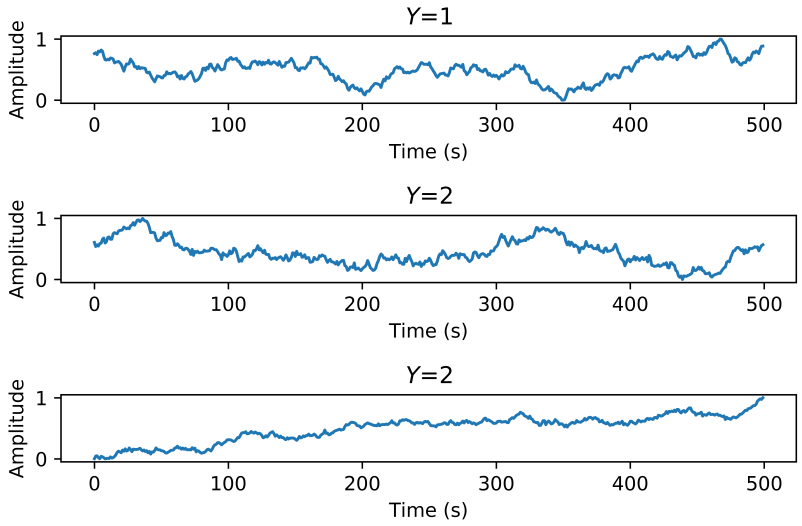
The task of feature selection aims to select a set of features that are relevant to the target such that $corr(S \mid f_{rel} \in S) \gg corr(S \setminus f_{rel})$.

Feature extraction: For a time series dataset, we aim to perform extraction of subsequences that are discriminative for the target prediction. In Definition 2.3, we formally define a univariate time series.

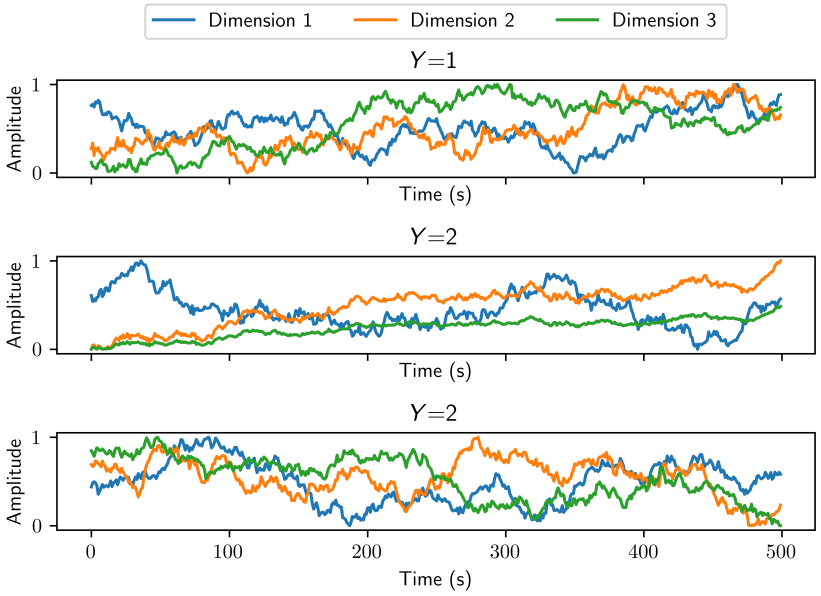
Definition. 2.3: Time Series

A time series X of length l is a collection of real numbers that are indexed based on time $t = \{1, \cdots, l\}$, i.e., $X = (x_1, \cdots, x_l) \mid x_i \in \mathbb{R}$.

As we aim to perform supervised correlation analysis on the time series data to identify relevant subsequences, each univariate time series X is provided with a target instance. For example, the univariate time series dataset in Figure 2.8a



(a) A univariate time series dataset of 3 time series samples



(b) A multivariate time series dataset of 3 time series samples and 3 dimensions

Figure 2.8: Time series dataset for a supervised feature extraction based on a discrete target Y

is a collection of three time series samples of length (l) 500 from healthy ($Y = 1$) and defective ($Y = 2$) sensors. That is, an entire series of length l is assigned a target value. Likewise, a multivariate time series sample is a collection of multiple univariate time series and each of them represents a dimension. This collection of univariate samples is assigned a target value. For example, Figure 2.8b shows a three-dimensional time series dataset with three samples and two classes.

Definition. 2.4: Feature Extraction Task

For a given time series $X = (x_1, \cdots, x_l) \mid x_i \in \mathbb{R}$ indexed by $t = \{1, \cdots, l\}$, the task of feature extraction aims to extract one or more relevant subsequence $X_{rel} = (x_a, \cdots, x_b) \mid a \neq b, a < l$ and $b > 1$ for the prediction of target Y. The subsequence is mapped to numeric features based on a defined property, i.e., $\mathcal{T} : X_{rel} \mapsto \mathbb{R}$.

Feature extraction aims to map the dynamic properties of the time series dataset into features based on a transformation function $\mathcal{T}$ (c.f. Definition 2.4). The transformation function defines the property of the series to encode in the feature. For example, $skew : X \mapsto \mathbb{R}$ transforms the time series X into a static feature that characterizes the degree of asymmetry of values around the mean value [NAM01].

2.2 Overview of feature selection literature

A feature selection algorithm should address the following points [Hal99]:

- **Search organization:** An exhaustive search in the high-dimensional feature space leads to high computation time. Hence, search strategies or heuristics that systematically traverse through the search space are applied. A search organization technique requires definition of a right starting point and a stopping criterion.
  - **Starting point:** A feature selection algorithm should have a defined starting point in the search space to begin exploration and evaluation.
  - **Stopping criterion:** The feature selection algorithm can be provided with a termination condition after which the search is stopped. For example, the user has an option to preset the maximum number of features to select and the algorithm stops further execution once this threshold is reached.
- **Evaluation strategy:** In order to judge if a feature is of any relevance for the target prediction, the algorithm needs a well defined quality evaluation criterion, i.e., the cost function $corr(f)$ in Definition 2.2.

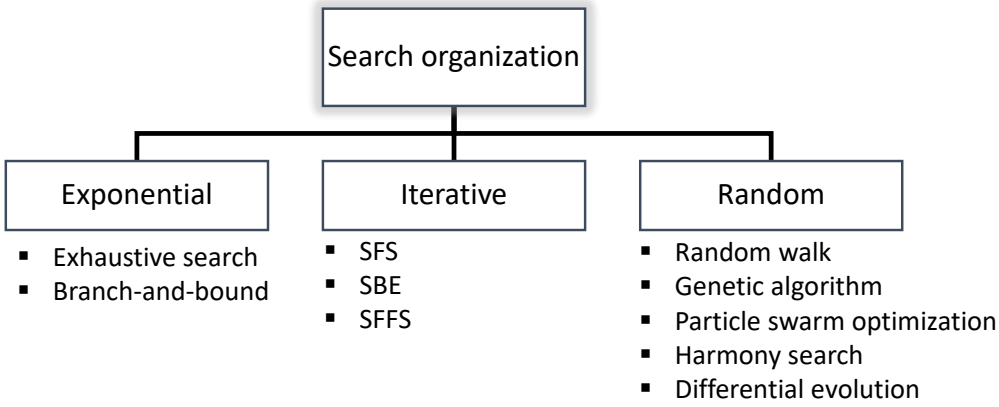


Figure 2.9: Different search organization approaches for feature selection [JBB15]

2.2.1 Search organization

To identify multivariate correlations in a d-dimensional feature space, there are 2^d feature subset combinations to evaluate. A search organization technique facilitates generation of the subset combinations to explore the search space. The generated subset combinations are evaluated for relevance based on the cost function, i.e., *corr* in Definition 2.2. The search organization is classified into exponential, iterative and random search (c.f. Figure 2.9).

Exponential

The exponential search technique such as exhaustive search generates all possible subset combinations for evaluation. In Figure 2.10, we show the search space for a dataset of four features and the 15 (excluding the null set) different feature combinations. For high-dimensional datasets (e.g., 100 features), exhaustive evaluation of all feature combinations (i.e., $2^{100} - 1$) is highly time consuming and not preferred. However, exhaustive search can achieve high accuracy in comparison to the other strategies and is preferable for dataset with low dimensionality (e.g., $d \leq 20$ features).

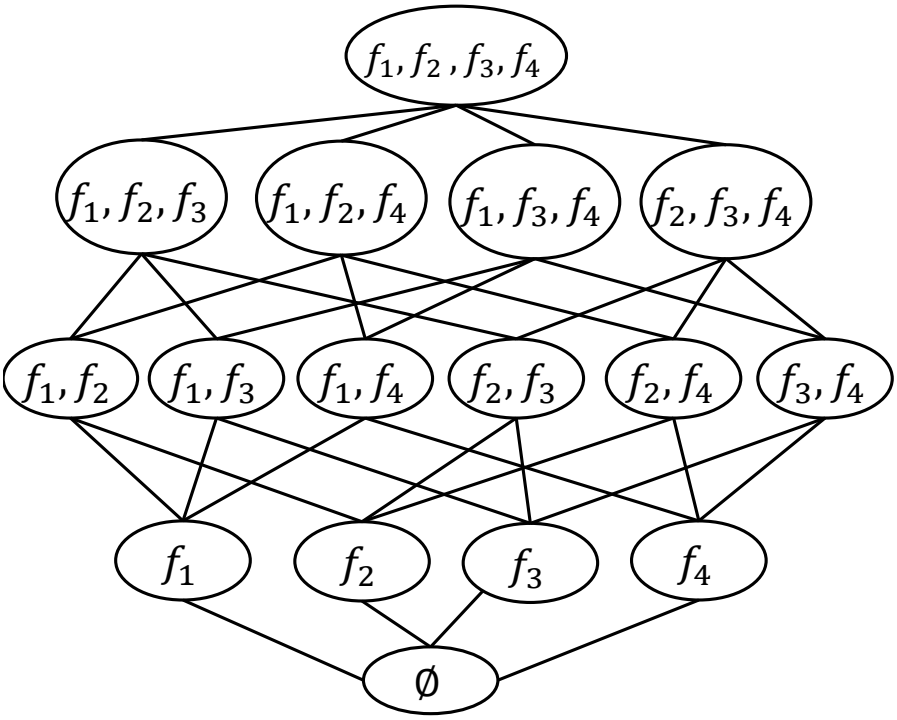
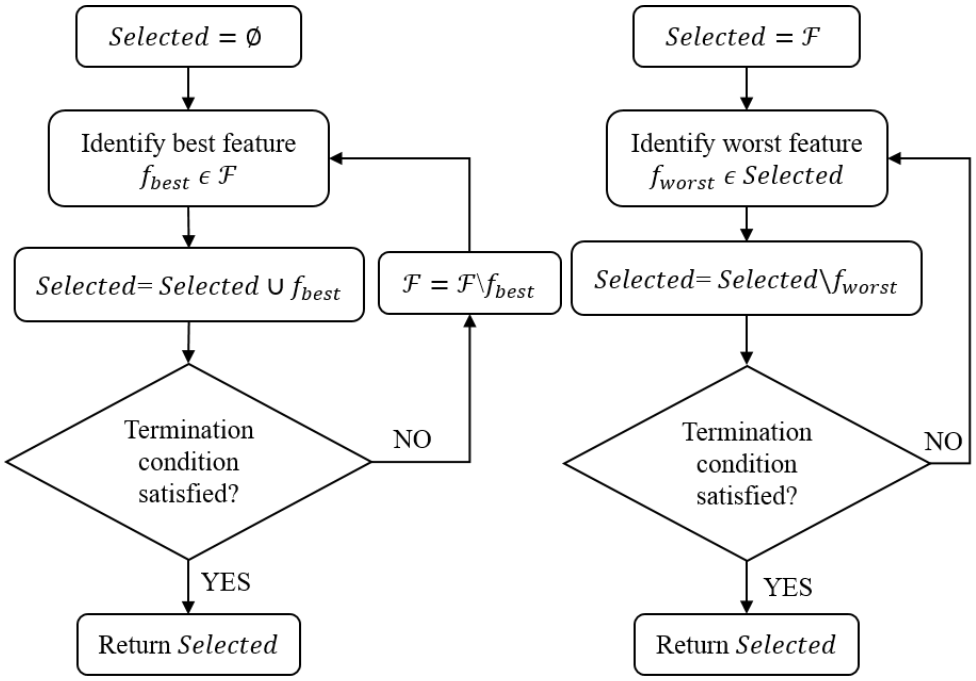


Figure 2.10: Search space for a given feature space $\mathcal{F} = \{f_1, f_2, f_3, f_4\}$

Iterative

Sequential Forward Selection (SFS), Sequential Backward Elimination (SBE) and Sequential Forward Floating Selection (SFFS) are the prevalent iterative search strategies [TPKC10]. Both SFS and SBE recursively add or eliminate features in each iteration respectively. As shown in Figure 2.11, the fundamental difference between them is the start condition. That is, SFS begins with an empty set and features that fulfill a criterion function (e.g., $corr(f) \geq \ threshold$) are recursively added to it. On contrary, in SBE, non-contributing features are recursively eliminated starting from the full-dimensional feature space until a termination condition is fulfilled. The drawback of SFS is its inability to evaluate a feature's higher-order interactions. On the other hand, SBE is unable to add features that were removed in the past iteration. SFFS was introduced to allow bidirectional search, i.e., addition and elimination of features. Though iterative approaches are comparatively efficient than exhaustive search, it is still not the most efficient technique to use for datasets with hundreds of features because it involves estimation of the criterion



(a) Forward selection (b) Backward elimination

Figure 2.11: Iterative search organization techniques

function several times, e.g., on a d-dimensional dataset, SFS evaluates the criterion function $\frac{d(d+1)}{2} - 1$ times.

Random Search

Random search, as the name suggests, involves randomness or probability in its search organization. They are preferred in several applications because they do not require any prior user inputs such as gradient information of the criterion function [JBB15]. For example, evolutionary algorithms such as Genetic Algorithms (GA) [BALD14] and Particle swarm Optimization (PSO) [SISB11] start from a randomly drawn subset combination, i.e., the start condition is decided randomly. In further iterations, new subset combinations are systematically generated under certain degree of randomness involved. The generated subset is evaluated using the criterion function in each iteration. Unlike the iterative search, by including randomness, the random search procedure avoids local optima [JBB15]. Although random search does not guarantee the most optimal solution, the approach allows a

Method	Advantage	Disadvantage	Complexity
Exhaustive search	Best accuracy	High run times	$\mathcal{O}(2^d)$
Iterative search	Efficient w.r.t. exhaustive search	Cannot handle higher-order interactions	$\mathcal{O}(d^2)$
Random search	Trade-off between runtime and accuracy	Does not guarantee the best accuracy	$\mathcal{O}(d \cdot logd)$

Table 2.2: Comparison of search organization techniques [JBB15, DSL15]

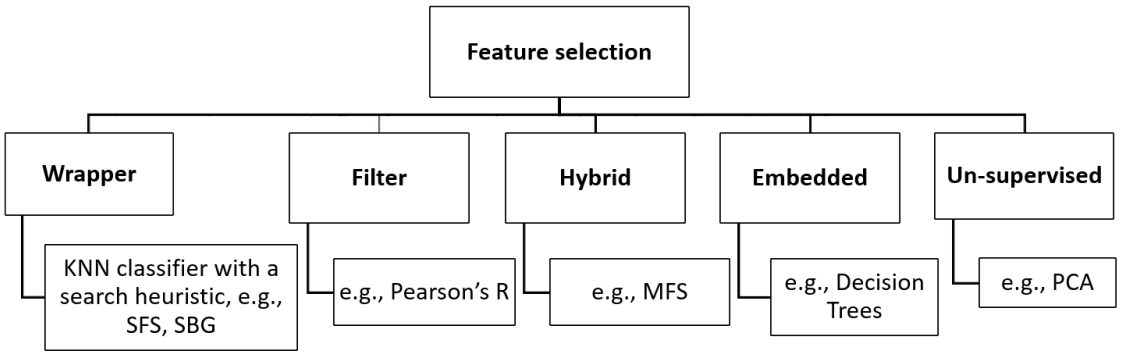


Figure 2.12: Feature selection paradigms

user to decide the trade-off between runtime and accuracy. Finally, we summarize the pros and cons of the various search organization techniques in Table 2.2.

2.2.2 Feature Selection Paradigms

Based on the evaluation strategy used, feature selection is classified in to different paradigms: wrapper, filter, hybrid, embedded and un-supervised [Qui14, GE03, RŠK03, KMB12] (c.f. Figure 2.12). Feature selection algorithms can return feature weights, rankings or subsets as output. It is the discretion of a user to choose the output type based on the desired application. Below, we briefly discuss the principles of each paradigm.

Wrapper paradigm

Wrapper-based approaches quantify the relevance of a feature by estimating the prediction error. That is, the cost function (c.f. Definition 2.2) used for the

task of feature selection is a classification or regression algorithm (e.g., K-Nearest Neighbors [KGG85]) itself. Although this approach often outperforms filter-based approaches w.r.t. the quality of predictions, the wrapper-based approaches are computationally inefficient [MBN02, SBS+17, SSM17]. In addition, the wrapper methods are prone to overfitting and have poor generalization capability [CS14], i.e., lower prediction accuracy when the set of selected features is tested with a different classification or regression algorithm. As discussed in Section 2.2.1, applying SFS on a d-dimensional dataset requires at most $\frac{d(d+1)}{2} - 1$ models to be trained. Therefore, wrappers are unrealistic for high-dimensional datasets.

Filter paradigm

Filter-based approaches quantify the relevance of a feature for the target prediction by evaluating certain statistical properties. That is, the cost function evaluates the correlation between a feature and the target to score its relevance. Some of the renowned filter-based correlation functions are Pearson's R [HK11], Mutual Information (MI) [CT12] and Distance Correlation (dCor) [SRB07]. The filter-based paradigms are known to be computationally efficient as they are independent of a classification or regression learning algorithm. Hence, they achieve better generalization in comparison to wrapper approaches [CS14, MBN02]. The existing surveys [MBN02, CS14, JBB15, UMC+18] on the topic of feature selection do not provide a consolidated summary of time complexities for the filter-based feature selection techniques. Hence, in Table 2.3 we tabulate the time complexities and various other properties of the state-of-the-art correlation measures and discuss them briefly below.

Pearson's R is a measure of linear relationship between a feature and the target. A strong negative or positive correlation between the two is denoted by a Pearson's R of -1 or 1 respectively. Similarly, Pearson's R of zero denotes no correlation. The advantage of the measure is its computational efficiency ($\mathcal{O}(N)$) and the drawback is its inability to handle non-linear dependencies and its sensitivity to outliers [XHHZ10, HK11]. The Pearson's R between a feature $f = (x_1, \cdots, x_N)$ (c.f Definition 2.1) and a continuous target $Y = (y_1, \cdots, y_N) \mid y_i \in \mathbb{R}$ is computed

as,

$$R(f,Y) = \frac{\sum_{i=1}^{N}(x_i - \bar{f})(y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{f})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{Y})^2}}, \tag{2.1}$$

where $\bar{f} = \frac{1}{N}\sum_{i=1}^{N} x_i$ and $\bar{Y} = \frac{1}{N}\sum_{i=1}^{N} y_i$.

Spearmans's ρ is a non-parametric measure of monotonicity between two features. Principally, it is Pearson's R on the ranked values of the feature. Hence, it inherits most of the advantages and drawbacks we mentioned for Pearson's R. However, the time complexity of ρ is considerably different from R as it requires sorting of the feature values, i.e., the time complexity of Spearman's ρ is $\mathcal{O}(N \cdot log(N))$ [XHHZ10].

Distance correlation (dCor) is a measure of correlation between two variables based on the distance between them. In contrast to Pearson' R, the distance correlation (dCor) is capable of identifying non-linear dependencies. Mathematically, it is represented using the distance covariance $dCov$ and the variances $dVar$ [SRB07].

$$dCor(f,Y) = \frac{dCov(f,Y)}{\sqrt{dVar(f)\ dVar(Y)}} \tag{2.2}$$

Estimating the distance correlation by applying Equation 2.2 involves a time complexity of $\mathcal{O}(N^2)$ [HS16].

Mutual Information (MI) is a non-linear and bivariate correlation measure that is founded on the principles of information theory. Mutual information, which is also known as Information Gain (IG), denotes the magnitude of information shared between the dependent and the independent features. It is computed using the Shannon entropy (H) [CT12], i.e.,

$$MI(f,Y) = H(f) - H(f \mid Y). \tag{2.3}$$

Theoretically, MI quantifies the reduction in uncertainty of feature f, given the information about Y or vice-versa (i.e., MI is a symmetric measure). While Equation 2.3 estimates mutual information based on the definitions of entropy, $MI(f,Y)$ can

also be estimated based on the probabilistic density functions [Kel15, DP05].

$$MI(f, Y) = \int \int p(f, Y)\ log \frac{p(f, Y)}{p(f)p(Y)} df dY, \tag{2.4}$$

where, $p(f, Y)$ is the joint probability density function and $p(f)$ and $p(Y)$ are marginal densities. The computational complexity for estimating mutual information relies on the methodology used. By computing the conditional probability in Equation 2.3 using Bayesian rule, the MI estimations require a time complexity of $\mathcal{O}(N^2 \cdot d)$ [KC02]. A wide range of correlation measures based on the theory of MI was introduced and we explain a few of them below.

Maximal Information Criterion (MIC) is a bivariate correlation function to analyze non-linear dependencies. MIC is a normalized version of MI that ranges between an interval of [0,1], where zero denotes statistical independence and one denotes dependence. It relies on the rudimentary estimation of MI on multiple grids of the feature [RRF+11]. The maximal MI of all the grids is estimated as the MIC value. There is no explicit definition of the computational complexity of MIC, however, it is estimated to be a polynomial of N and number of unique values m in the variable [ZZX13]. Multivariate Maximal Correlation Analysis (MAC) is a multivariate extension of MIC with an improvement in its binning strategy. The time complexity for estimation of a MAC score is $\mathcal{O}(d^2 \cdot N^{1.5})$ [NMV+14].

Maximum relevance minimum redundancy (mRmR) also applies the principle of MI to compute the relevance and redundancy of features. For a given subset of features, mRmR is based on the pairwise feature-target and feature-feature mutual information estimations. To find a subset of relevant features S from a d-dimensional feature space, the computational complexity for the incremental search method used in mRmR is $\mathcal{O}(|S| \cdot d)$ [DP05, PLD05]. This idea of pairwise independence estimation was also experimented using a non-symmetric measure called monotone dependence ($\mathcal{M}_d$) [SP10, CM14] with $\mathcal{O}(N \cdot log(N))$ time complexity and Joint Mutual Information (JMI) [BPZL12] as the criterion function. JMI for a feature $f_i \in S \mid S \subseteq \mathcal{F}$ and $k = |S|$,

$$JMI(f_i, Y) = \sum_{f_j \in S} MI(f_i f_j, Y), \tag{2.5}$$

is the information between the target and the joint random variables $f_i f_j$. Hence, its computation involves $\mathcal{O}(k)$ and $\mathcal{O}(k^2)$ evaluations of MI and conditional MI's respectively [YM00].

Fast Correlation-Based Filter (FCBF) is based on the principles of mutual information. It uses Symmetric Uncertainty (SU) to compute the feature relevance and redundancy based on a threshold parameter [YL03].

$$SU(f, Y) = 2\left[\frac{MI(f, Y)}{H(f) + H(Y)}\right] \tag{2.6}$$

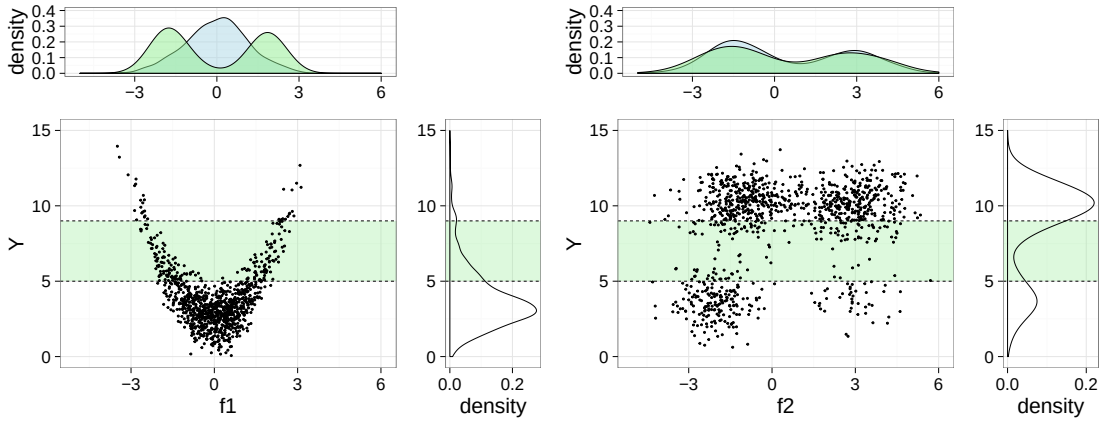
The algorithmic framework of FCBF using SU (c.f. Equation 2.6) has an overall time complexity of $\mathcal{O}(N \cdot d \cdot log(d))$. A similar approach based on Conditional Mutual Information Maximization (CMIM) criterion was also proposed for classification tasks with binary features [Fle04]. For a N_c class problem, the time complexity of CMIM is $\mathcal{O}(N_c^3)$.

Correlation-based feature selector (CFS) is a multivariate correlation measure to quantify the magnitude of non-linear dependency between a feature subset and the target. Similar to mRmR and FCBF, CFS also scores the redundancy of features in a subset. For a given k-dimensional feature subset S,

$$CFS(S, Y) = \frac{k\ r_{tf}}{\sqrt{k + k(k-1)\ r_{ff}}}, \tag{2.7}$$

where r_{tf} is the mean feature-target correlation and r_{ff} is the average feature-feature correlations. CFS uses Symmetrical Uncertainty (SU) as a measure of correlation and it is applicable for categorical features only [SM11]. Its computation complexity is $\mathcal{O}(N \cdot \frac{k^2-k}{2})$ [Hal99].

High Contrast Subspaces (HiCS) with significant divergence between the conditional and marginal distribution along the dimensions are used for the task of outlier detection [KMB12]. The magnitude of divergence between the distributions is called contrast. The contrast score is also employed as a correlation measure for



(a) A feature f_1 correlated to the target has large difference between the conditional and marginal distributions

(b) A feature f_2 not correlated to the target has low difference between the conditional and marginal distributions

Figure 2.13: Contrast as a measure of statistical dependence

relationship analysis on subspaces with continuous features [Kel15].

$$contrast(f, Y) = divergence\,(p(Y \mid f) \mid\mid p(Y)) \tag{2.8}$$

As shown in Figure 2.13, a relevant feature f_1 exhibits a high divergence between marginal and conditional distributions, i.e., high contrast. On contrary, an irrelevant feature f_2 exhibits a low divergence between the two distributions. Estimation of the contrast score between a feature subset $S \subseteq \mathcal{F}$ and the target Y is performed with a computational complexity of $\mathcal{O}(|S| \cdot N)$. The principle of quantifying the dependency between the dimensions of a subspace for outlier detection was enhanced by evaluating the cumulative distributions in the work of Cumulative Mutual Information (CMI) [NMV+13].

ReliefF is a multivariate feature relevance scoring scheme based on the nearest neighbors [Kon94]. Using a sample of instances from the data, its nearest neighbors in the same and opposite class are identified based on a distance function. The rationale is that a useful attribute should differentiate between instances from different classes and have the same value for instances from the same class [CM14]. The time complexity for the computation of ReliefF rankings are $\mathcal{O}(N^2 \cdot d)$ [UMC+18].

Filter-based approaches	Multivariate	Selection type	Relevance/Redundancy	Time Complexity
Pearson's	✗	Feature ranking	✓/✗	$\mathcal{O}(N)$
Spearmans-ρ	✗	Feature ranking	✓/✗	$\mathcal{O}(N \cdot log(N))$
dCor	✗	Feature ranking	✓/✗	$\mathcal{O}(N^2)$
Mutual Information	✗	Feature ranking	✓/✗	$\mathcal{O}(N^2 \cdot d)$
MIC	✗	Feature ranking	✓/✗	$\mathcal{O}(polynomial(N, m))$
MAC	✓	Subset ranking	✓/✗	$\mathcal{O}(d^2 \cdot N^{1.5})$
mRmR	✓	Feature ranking	✓/✓	$\mathcal{O}(\|S\| \cdot d)$
$\mathcal{M}_d$	✓	Feature ranking	✓/✓	$\mathcal{O}(N \cdot log(N))$
JMI	✓	Feature ranking	✓/✓	$\mathcal{O}(k^2)$
FCBF	✓	Subset selector	✓/✓	$\mathcal{O}(N \cdot d \cdot log(d)))$
CMIM	✓	Feature ranking	✓/✓	$\mathcal{O}(N_c^3)$
CFS	✓	Subset selector	✓/✓	$\mathcal{O}(N \cdot \frac{k^2-k}{2})$
HiCS	✓	Subset ranking	✓/✗	$\mathcal{O}(\|S\| \cdot N)$
ReliefF	✓	Feature ranking	✓/✗	$\mathcal{O}(N^2 \cdot d)$
[FKZ15]	✓	Subset ranking	✓/✓	$\mathcal{O}(MI \cdot H \cdot mRW)$
[OTN99]	✓	Subset selector	✓/✓	$\mathcal{O}(L \cdot N_c)$
[SAVdP08]	✓	Feature ranking	✓/✗	Based on the correlation measure used
[SM11]	✓	Feature ranking	✓/✓	Based on the correlation measure used

Table 2.3: Summary of filter-based approaches, where the last four literatures are ensemble methods

Ensemble methods are also one of the prevalent ideas in the topic of correlation analysis and are proven to enhance the robustness of feature selection [SAVdP08, FKZ15]. The work of [SAVdP08] introduced the idea of applying bootstrap aggregation to generate bags of data. For each bag, the feature-target correlations are estimated and are finally combined by weighted voting. An extension of this work, using the idea of pairwise correlations from mRmR and CFS was introduced to improve the efficiency and handle feature redundancy [SM11]. Similarly, the work of [FKZ15] aggregates the results of mutual information, entropy and modified relief weight (mRW) by weighted geometric mean. The time complexity of the aforementioned approaches is dependent on the choice of correlation measures used. By selecting multiple feature subsets, another ensemble feature selection approach for classification task was proposed in the work of [OTN99]. An ensemble average of the prediction models trained for each class in the target Y is used as final predictions. In addition, they promote the reduction of correlation between classifiers to achieve diversity amongst them. As it involves training a meta-learning algorithm of time complexity $\mathcal{O}(L)$ for N_c classes, the total complexity is represented as $\mathcal{O}(L \cdot N_c)$.

Hybrid paradigm

Hybrid feature selection approaches were introduced to surpass the computational inefficiency of the wrapper paradigm and exploit the generalization capability of filter paradigm. First, a preliminary filtering of the noisy features is performed using a correlation function. Then a wrapper-based approach is performed on the reduced search space to improve the computational efficiency. Several hybrid approaches such as Doquire [DV11], Mixed Feature Selection (MFS) [TM07], [FH19] and [HHL11] aim to address the problem of inefficiency (in wrappers) by building fewer classifier models.

Doquire's approach is a hybrid feature selection technique for mixed datasets [DV11]. The fundamental idea involves ranking of continuous and categorical features based on their relevance to the target. This ranking is performed using MI (c.f. Equation 2.4) and mRmR [DP05] respectively. The ranked features are combined based on the classifier accuracy. In comparison to SFS (c.f. Figure 2.11a), the Doquire approach is more efficient by training the prediction model only $d-1$ times.

Mixed Feature Selection (MFS) uses Mahalanobis distance and symmetric uncertainty (c.f. Equation 2.6) for evaluating the relevance of continuous and categorical features respectively. For subsets with mixed feature types, it introduces the idea of error probability* estimation by decomposing the continuous features along different categories of the categorical features [TM07].

Other approaches such as, [HHL11] perform initial screening of the feature using F-score and MI. Likewise, [Lee09] applies F-score and Supported-Sequential Forward Search. These pre-screened features are then fine-tuned by evaluating their prediction accuracy.

Embedded paradigm

Under embedded paradigm, feature selection is performed as a part of the prediction algorithm. Embedded methods are similar to wrappers, but are computa-

*K-Nearest Neighbors was employed to estimate the posterior probability

Method	Advantage	Disadvantage
Wrapper	High accuracy	Lacks generalization
Filter	Efficient	Less accurate w.r.t. wrapper
Hybrid	Low run time[†]	Builds multiple prediction models
Embedded[‡]	Less prone to overfitting	Accuracy influenced by split criterion
Unsupervised	Efficient	Strong assumptions

Table 2.4: Summary of various feature selection paradigms

tionally less expensive and are less prone to overfitting. Decision tree learner is a renowned example of the embedded paradigm [Qui14]. Embedded approaches such as decision trees perform the selection inherently as a part of the prediction. For a given set of features, a flowchart of decisions and their consequences are analyzed. A consequence is considered as favorable or not, based on a split criterion. Hence, the correlation analysis is performed by this split criterion internally, e.g., by evaluation of Gini indices and MI. Several ensemble based extensions of decision trees include Random forest, Adaboost and Bagging [FSA99, Bre01].

Un-supervised paradigm

Un-supervised methods perform the selection of features without evaluating their relevance to the target. For example, Principal Component Analysis (PCA) selects features only based on the variance explained and does not require a target vector [Shl14], i.e., in Definition 2.2, $corr : f \mapsto \mathbb{R}$. Such a strong assumption is one major drawback of unsupervised approaches. Auto-encoders are yet another unsupervised approach for feature selection. For example, if the original feature space can be reconstructed from the latent dimension after discarding a feature, this feature is deemed to be redundant [HWZ+18]. This thesis intends to address the research gaps in supervised correlation analysis. Hence, we do not discuss the unsupervised approaches more in detail. Finally, we summarize the pros and cons of various feature selection paradigms in Table 2.4.

[†]In comparison to wrappers
[‡]Pros and cons discussed w.r.t. decision trees

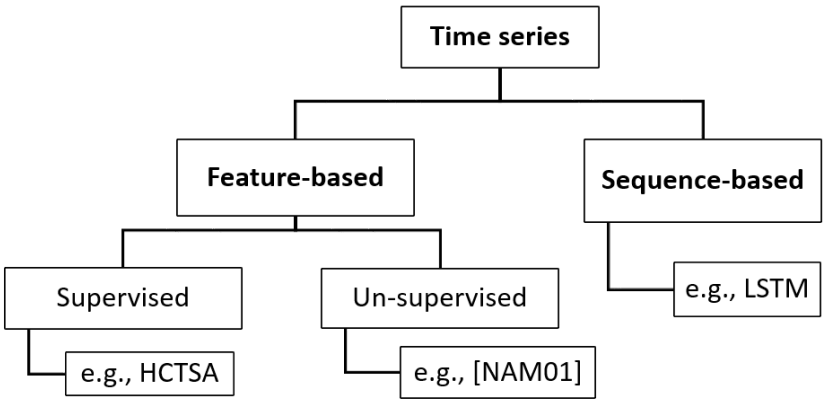


Figure 2.14: Time series approaches

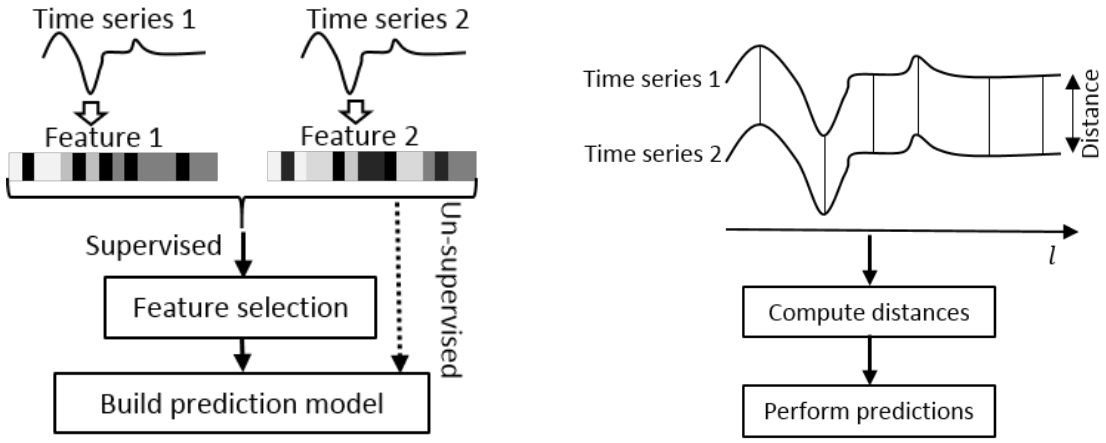
2.3 Overview of feature extraction literature

The task of correlation analysis in time series data is very similar to the task of feature selection (c.f. Definition 2.2). The additional concern is to ensure that the indices of the data values are preserved during the analysis. However, in several time series applications, only a specific subsequence or a time window is relevant for the classification or regression task [YK09, WJW+15] (c.f. Figure 1.2 in Chapter 1). In such datasets, it is necessary to analyze and extract the relevant events for a given analytical task (e.g., classification). As shown in Figure 2.14, time series correlation analysis is performed in two different ways, i.e., feature-based and sequence-based. Feature-based approaches transform the dynamic properties of the time series into static features (c.f. Figure 2.15a) and the correlations are evaluated on this transformed feature space. Sequence-based approaches perform predictions by evaluating the distances between the time series (c.f. Figure 2.15b). That is, they do not require transformation of the time series into numeric features.

2.3.1 Time series learning paradigms

Feature-based paradigm

In automotive applications, the time series data is transmitted from the vehicle to a remote location [SdSFS18]. In such cases, the transmission costs are large for lengthy and high-dimensional time series signals. Feature-based approaches handle



(a) Feature-based (b) Sequence-based

Figure 2.15: Feature extraction paradigms

this problem by transforming the lengthy time series into compact feature sets. The feature-based transformation paradigm is further classified into un-supervised and supervised approaches. In an un-supervised approach, the relevance of the transformed feature is not evaluated. The example of *skew* based transformation in the previous section (c.f. Section 2.1.4) is an un-supervised approach where the relevance of the transformed feature to the target is not evaluated.

A wide range of unsupervised feature extraction methods exist. The work of Nanopoulos [NAM01] extracts mean, standard deviation, kurtosis and skew of the original and a transformed series with reduced sampling rate. A frequency based approach was introduced in the work of Wang [WSH06, WWW07]. Symbolic aggregate approximation (SAX) based feature extraction transforms the time series into symbolic representations or words. By using a bag-of-words, the structural changes in a time series are encoded into features [LKL12]. Using Dynamic Time Warping (DTW) distance as a transformation function for feature extraction is proven to achieve high accuracy on univariate time series classification tasks [Kat16]. As a supervised approach, the work of Highly Comparative Time series Analysis (HCTSA) performs feature selection after feature extraction [FJ14].

This work will present the concept of using ordinality as a property of time series that can be mapped into static features, i.e., $ordinality : X \mapsto \mathbb{R}$ (c.f. Chapter 5). Ordinality is a property that evaluates the qualitative changes in univariate time series by identifying the relation between the values in a subsequence. It was

Method	Advantage	Disadvantage
Un-supervised feature-based technique	Efficient	Large feature space
Supervised feature-based technique	Relevant features	Inefficient
Sequence-based technique	No transformation required	Inefficient for long time series

Table 2.5: Advantage and disadvantage of various time series approaches

introduced as a complexity measure to compare univariate time series [BP02] and later extended for change detection [CTG+04, SGK12] and variability assessment in ECG signals [GGK+13]. In applications where the series contain ordered structures (e.g. Electroencephalography signals), existing entropy based measures may be inaccurate in comparison to the ordinality based complexity measure [LOR07].

Sequence-based paradigm

Sequence-based approaches evaluate the data at each instant of time in a series to infer a decision (e.g., class). Unlike feature-based approaches, the sequence-based paradigm does not transform the time series into a set of static features. That is, the representative subsequence is identified directly on the time series data (c.f. Figure 2.15b). In addition, sequence-based approaches are supervised, i.e., they require a vector of target labels Y. Shapelet technique classifies new time series based on the distance between the subsequence of a time series (shapelet) and the new time series [YK09]. The work was extended in Maximum Correlation and Minimum Redundancy (MCMR) shapelets [WJW+15], to extract non-redundant shapelets for univariate time series classification. Recurrent neural network frameworks such as Long Short-Term Memory (LSTM) are renowned for multivariate time series classification tasks [HS97]. We summarize the pros and cons of the various time series approaches in Table 2.5.

2.4 Thesis contributions in comparison to the related literature

In this section, we briefly describe the novelty of our work in comparison to the state-of-the-art approaches discussed in Section 2.2. In Chapter 3, we will present Diverse Subset Selection Strategy (DS3), a framework to select multiple diverse

subsets from a dataset with continuous features. The majority of the feature selection methods discussed in Section 2.2 from different paradigms emphasize the selection of a single projection of the high-dimensional feature space. In contrast to these approaches, DS3 selects multiple relevant projections of the feature space. In comparison to the ensemble approaches [OTN99, SAVdP08, SM11], we employ different correlation measures, because each of them estimates the importance of a feature based on different intrinsic properties. Nevertheless, we do not aggregate the results of multiple correlation measures as proposed in the work of [FKZ15]. Instead, we generate initial candidates based on multiple correlation measures by following the hybrid paradigm [TM07, Lee09, DV11, HHL11]. However, the initial candidates are later used for generation of subsets that have complementary information. In contrast to the embedded selection technique of Random Forests [Bre01], which bags the results of multiple decision trees from random subsets, our approach selects subsets by augmenting diversity between features and considering multiple intrinsic relationships between the variable and target.

Contribution 1: Selection of multiple relevant feature subsets based on different intrinsic properties they exhibit with the target and enhancing the complementary information between them are the novel propositions of DS3.

In Chapter 4 we extend the multi-view approach for correlation analysis by including the higher-order interactions and redundancies in mixed datasets, i.e., continuous and categorical features. Wrapper approaches with Sequential Forward Selection (SFS) can handle redundancy, but they are not capable of evaluating feature interactions [SBS+17]. Using Sequential Backward Elimination (SBE) addresses the problem of higher-order interactions [TPKC10]. However, the major problem of this paradigm is efficiency because the selection always depends on training the classifier numerous times. Though hybrid approaches are computationally efficient in comparison to wrappers, they also involve training of classifiers multiple times. Hence, in contrast to DS3 and other hybrid paradigms discussed in Section 2.2.2, our novel feature ranking framework Relevance and Redundancy (RaR) follows the filter-based paradigm and does not require training of multiple prediction models. In Table 2.3, various filter-based approaches are listed, which perform multivariate feature ranking by evaluating relevance and redundancy, e.g., CFS [Hal99], FCBF [YL03], mRmR [PLD05, DP05]. Tree-based embedded tech-

niques are also well-known for handling mixed data and redundancy [Qui14, Bre01]. However, the aforementioned works do not address higher-order interactions between more than two features. That is, they always focus on pairwise analysis of relevance and redundancy. Similarly, CMIM [Fle04] and JMI [BPZL12] evaluate feature relevance and redundancy. However, CMIM is limited to boolean features and both have limitations for computing higher-order interactions between more than two features. Unsupervised subspace search techniques [KMB12, NMV+13] consider higher-order interactions. However, these approaches focus on providing a score for the entire subspace. In contrast, we intend to rank individual features by including their interactions with other features and the target. Moreover, the above discussed subspace methods are incapable of redundancy elimination. From game-theoretical concepts, the shapley value identifies the individual feature importance in terms of R^2 value from linear regression [PHHN16]. Though, it can be transferred for non-linear problems by using a non-linear regressor, this will decrease the efficiency of the approach. Our approach being filter-based does not require learning of additional classification or regression models.

Contribution 2: Including higher-order interactions between more than two features for efficient estimation of relevance and redundancy in mixed data are the novel contributions of RaR.

In Chapter 5 we propose Ordinal feature extraction (*ordex*), a supervised multivariate correlation analysis on time series for feature extraction. Exploiting ordinality as a property for feature extraction in multivariate time series is yet unexplored. In contrast to the existing works on ordinality [BP02, SGK12, GGK+13], *ordex* is the first work to employ ordinality as a property for feature-based transformation of multivariate time series. Traditional feature-based approaches [NAM01, WSH06, WWW07, LKL12, Kat16] discussed in Section 2.3.1 perform feature extraction without considering the relevance and redundancy of the extracted features. Contrasting HCTSA [FJ14], we do not perform feature selection after extracting a high-dimensional feature space from the time series dataset. *Ordex* is a feature extraction methodology for multivariate time series that simultaneously generates and evaluates the features for their relevance and redundancy without additional post-processing such as feature selection. In contrast to the univariate shapelet techniques [YK09, WJW+15], our approach extracts relevant and novel features

based on co-occurrence of time series events in multiple dimensions, i.e., multivariate. In comparison to multivariate LSTM [HS97], our approach scales better w.r.t. run times. That is, we efficiently generate features based on relevant multivariate correlations in the time series dataset by also evaluating the redundancy.

Contribution 3: Defining the multivariate nature of ordinality in time series and an efficient methodology for simultaneously extracting and evaluating its relevance and redundancy are the novel contributions of ordex.

One of the major goals of this dissertation is to make correlation analysis understandable to users. Recent trends largely demand explainable Artificial Intelligence (AI) systems and several literatures focus on making AI understandable to users [SWM17, ZZ18, RSG16]. This dissertation presents explainable multivariate correlation analysis as a sub-field of explainable AI. In Chapter 6, we introduce a graph-based Framework for Exploring and Understanding Multivariate Correlations (FEXUM) that enhance the user's understanding of all correlations in the dataset. All feature selection techniques discussed in Section 2.2 provide a relevant feature subset, feature ranking or feature weights. In contrast, our contribution provides a visualization framework to support the user in understanding multivariate correlations in the dataset. In addition, we provide a consolidated visualization of all correlations (i.e., relevance and redundancy) in the dataset. This allows users to understand which groups of features are redundant to each other and the magnitude of their redundancy.

Contribution 4: A software framework to visualize all correlations (i.e., relevance and redundancy) in a high-dimensional dataset and to enhance the transparency of multivariate correlation analysis are the novel contributions of FEXUM.

Chapter 3

Diverse Selection of Feature Subsets

3.1 Motivation

Regression models for predicting sensor values assist engineers to test the system response before stepping into production phase. For example, multiple information sources such as process variables, other sensor values and driving characteristics are used as predictors to predict the target values of a sensor in the automotive industry.

The challenge arises when the predictor variables stem from multiple sources and exhibit complex relationship amidst them. Due to the heterogeneity of the sources, such features show different properties between itself and the target. For instance, let us consider the task of predicting the values of a temperature sensor Y in an automobile. A subset of predictor variables S_1 representing the air system of the vehicle is related to the target by a function $F : S_1 \mapsto Y$. However, we observe another subset of predictor variables S_2 representing the fuel system of an automobile which is related to Y by a different function $G : S_2 \mapsto Y$. In a real world scenario, several such interactions are hidden in the dataset. Conventional feature selection techniques [MBN02, LMD+12, YL03, KJ97] consider only a single projection of the feature space. Hence, the effect of these intricate local interactions cannot be captured.

For a high-dimensional feature space, several feature combinations are possible. An exhaustive search of every possible combination is inefficient. Additionally, the selected subsets have to be diverse and non-repetitive in nature. Diverse subsets contain new knowledge from which the regression algorithm can harness the local interactions. Repetitive subsets containing redundant information are undesirable as considering similar projections multiple times does not contribute to discover any new underlying patterns. This calls for an efficient strategy to generate diverse and non-redundant subsets.

Our work Diverse Subset Selection Strategy (DS3)* provides a framework for multiple subsets selection and involves two key components.

(1) *A technique for feature selection based on multiple correlation properties.*

(2) *A search strategy for the selection of multiple diverse feature subsets for en-*

*Adapted by permission from Springer Nature: Diverse Selection of Feature Subsets for Ensemble Regression in the proceedings of the International Conference on Big Data Analytics and Knowledge Discovery (DaWaK), 2017 [SSM17]

Paradigm	Approach	Relevancy	Multiple subsets	Diversity	Multiple intrinsic properties	Efficiency
Filter	e.g., FCBF [YL03]	✓	✗	✗	✗	✓
Wrapper	e.g., GA/SVM [BALD14]	✓	✗	✗	✗	✗
Un-supervised	PCA [Shl14]	✗	✗	✗	✗	✓
Ensemble	[OTN99]	✓	✓	✓	✗	✓
	[FKZ15]	✓	✗	✗	✓	✓
Hybrid	**DS3**	✓	✓	✓	✓	✓

Table 3.1: Comparison of DS3 with other relevant literatures from different feature selection paradigms

hancing ensemble regression.

Our strategy prunes the non-essential variables and generate multiple projections of the feature space. Due to heterogeneity of the interactions, each projection of the feature space has different influence on the target prediction. Hence, each of them is to be evaluated based on multiple properties they exhibit with the target. To address this, the first component of our approach extracts initial candidate sets based on multiple correlation measures following the filter-based paradigm. The second component aims to generate novel dissimilar subsets that also contribute for the prediction quality. Hence, each subset provides not only complementary but also essential information for the target prediction. Finally, an ensemble regression model for each subset is trained to obtain a composite hypothesis that maps each diverse subset to the target. The final predictions are based on the unified results of each individual composite hypothesis.

In our experiments, we compare our approach to several existing feature selection algorithms and regression models on synthetic and real world data sets. The results of DS3 show better scalability and an improvement of the prediction quality.

3.2 Comparison to Related Work

In Chapter 2 we briefly described various state-of-the-art approaches. Several feature selection methods such as [YL03, Shl14, BALD14, PLD05] select a single projection of the feature space. Ensemble approaches such as [FKZ15, OTN99] do not consider both, i.e., multiple correlation measures and diversity between the feature subsets. Hence, selection of multiple relevant views of the feature space by capturing different intrinsic properties in the dataset and enhancing complemen-

tary information between them based on a diversity criterion are the unique value proposition of our work. In Table 3.1, we compare the existing techniques from different paradigms to DS3 based on these novel propositions.

3.3 Problem Definition

In this section, we formally define the problem that we aim to solve. Given a d-dimensional feature space $\mathcal{F} = \{f_1, f_2, \ldots, f_d\}$ containing N instances and a target Y. Both the feature space and target are defined as continuous values (c.f. Definition 2.1. We aim to identify a family of m feature subsets, such that:

$$\mathcal{P} = \{S_1, S_2, \ldots, S_m\}.$$

Each set $S_i \subset \mathcal{F} \mid i = 1, \ldots, m$ is selected under the fulfillment of the following constraints:

(1) Only relevant features are selected in S_i based on a function space $C = \{corr_1, \ldots, corr_k\}$ of k correlation measures:

$$\{corr_1(f, Y) \ldots corr_k(f, Y)\},$$

such that each correlation measure $corr_i : (f, Y) \mapsto \mathbb{R}$ and for an unsupervised measure (e.g. PCA) $corr_i : (f) \mapsto \mathbb{R}$.

(2) To avoid subsets with similar features, we define diversity of subsets based on a difference criterion:

$$diff(S_i, S_j), \forall (S_i, S_j),$$

where the $diff(S_i, S_j)$ function returns a diversity enhanced feature set. The choice of the difference function will be elaborated in the forthcoming sections. As a hybrid approach, to evaluate the quality of the subsets, we need a regression algorithm Reg. A regression algorithm approximates a function between the feature set and the target $Reg : S \mapsto \widehat{Y}$, where $\widehat{Y}$ is the predictions of Y. The $error : S \mapsto e$ function quantifies the fit errors $e \in \mathbb{R}$ of the regression model

using S. For a family of m subsets $\mathcal{P}$, the collection of fit errors of each subset is represented as an m-tuple, i.e., $\epsilon = \{e_1, e_2, \ldots, e_m\}$. Hence, a subset $S_i \in \mathcal{P}$ corresponds to the fit error $e_i \in \epsilon$.

For the selection of relevant variables in each S_i, we need multiple correlation functions that quantifies the importance of each feature for prediction of Y. After this step, diverse projections of relevant features that have both high difference $diff(S_i, S_j)$ in the feature sets and contribution for target predictions based on the fit errors $error(S_i)$ are to be efficiently identified.

3.4 Relevance Based Generation of Initial Candidates

DS3 has two major phases in the selection of subsets. The first phase prunes the non-contributing features from the feature space and create subsets which exemplify different properties. To achieve this, each feature is evaluated using a set of correlation functions. We compute the following $d \times k$ matrix,

$$\mathbf{M} = \begin{bmatrix} corr_1(f_1, Y) & corr_2(f_1, Y) & \ldots & corr_k(f_1, Y) \\ corr_1(f_2, Y) & corr_2(f_2, Y) & \ldots & corr_k(f_2, Y) \\ \vdots & \vdots & \ddots & \vdots \\ corr_1(f_d, Y) & corr_2(f_d, Y) & \ldots & corr_k(f_d, Y) \end{bmatrix}.$$

The matrix depicts the relevance of each feature based on several correlation functions. However, one of our aim is to reduce dimensionality by neglecting irrelevant features. In order to prune them, we calculate a threshold value. This value defines the magnitude of correlation below which the features have to be neglected. We calculate thresholds for each correlation function by multiplying the user defined parameter $\alpha \in [0, 1]$ with the maximum value in each column of the matrix $\mathbf{M}$. For each correlation function, all features with a correlation magnitude greater than or equal to the threshold are selected. As a result, each correlation measure selects a subset of features. That is, for k correlation functions, we obtain $S_1, \ldots, S_k$ subsets. We represent the collection of subsets as $\mathcal{P}$. Each set in $\mathcal{P}$ has features that exemplifies an intrinsic property based on the correlation function.

This is the first essential step for ensuring diversity in the second component of our strategy. Following the paradigm of hybrid approach we calculate the fit error e_i for each feature subset in $\mathcal{P}$ using a regression algorithm, i.e., $e_i = error(S_i)$ and the fit errors are updated in $\epsilon = \{e_1, e_2, \ldots, e_k\}$. After pruning, the dimensionality of each subset S_i is implicitly dependent on α. For a correlation function within the range of [0,1], a factor of $\alpha = 1$ leads to the omission of all features except those which have the maximum correlation in each column. On the other hand, a value of zero tends to retain all the values.

Algorithm 1 Choose initial candidates

Input: $\mathcal{F}, Y, \alpha, C$

1: **for** $m = 1 \rightarrow |C|$ **do**
2: **for** $p = 1 \rightarrow d$ **do**
3: $\mathbf{M}_{pm} = corr_m(f_p, Y)$
4: **end for**
5: $S_m = \{\mathcal{F} \mid \mathbf{M}_{*m} \geq (max(\mathbf{M}_{*m}) * \alpha)\}$
6: $e_m = error(S_m)$
7: **end for**

return $\mathcal{P}$ and ϵ

By extracting feature subsets using multiple correlation functions, we address the first requirement defined in Section 3.3. This is a necessary step for obtaining diverse subsets with complementary information. However, this is not a necessary condition for obtaining the best prediction quality. Therefore, we need a strategy for subset generation that increases diversity and contributes for prediction quality in parallel.

3.5 Multiple Feature Sets based on Difference and Quality

A preliminary selection of subsets based on different correlation functions decreases the dimensionality. Nonetheless, there are several other combinations that were not considered and they may increase the prediction quality. Therefore, we have to efficiently search for new subsets that satisfy the difference criterion (c.f. Section 3.3). The difference criterion aims at generation of subsets with complementary or new information to the regression model. Considering a difference criterion based

on complex measures such as information gain requires a higher time complexity w.r.t. database size. As these computations have to be evaluated for a large number of subsets, it is necessary to choose a criterion that performs efficiently. Hence, we choose to apply concepts from set theory. The different operations for combining sets are, Union, Intersection and Symmetric difference. Let's assume two sets of features: $S_1 = \{f_3, f_9, f_{14}\}$ and $S_2 = \{f_1, f_2, f_3\}$. By performing union operation between two sets, we obtain $S_1 \cup S_2 = \{f_3, f_9, f_{14}, f_1, f_2\}$. The union operation generates a larger set which does not capture the local interactions and ends in a full-dimensional feature set over several iterations. Performing intersection operation $S_1 \cap S_2 = \{f_3\}$ does not enhance diversity. It creates a subset based on a feature whose role has been captured by multiple feature combinations. Symmetric difference ($\triangle$) between two sets returns the objects that belong to one of the sets but not to their intersections, i.e., $S_1 \triangle S_2 = \{f_1, f_2, f_9, f_{14}\}$. The new subset generated by applying $\triangle$ operation has led to elimination of feature that exists in both sets. Thus, it partially contributes for dimensionality reduction. Secondly, it enhances diversity by eliminating features whose contribution has already been captured in both sets. The core idea of our approach is on learning new information from diverse feature combinations. To achieve this, we select non-intersecting elements and avoid generation of subsets with redundant features (w.r.t. initial candidates).

Symmetric difference is associative and commutative [Ols86], i.e., $S_1 \triangle S_2 = S_2 \triangle S_1$. Hence, a family of m sets generate $\binom{m}{2}$ number of new offspring subsets by applying symmetric difference between itself and each of the other candidates. For example, the initial candidate subsets $\mathcal{P} = \{S_1, S_2, S_3\}$ generates 3 offspring subsets, i.e., $(S_1 \triangle S_2), (S_1 \triangle S_3), (S_2 \triangle S_3)$. For large values of m, using all the $\binom{m}{2}$ offsprings is inefficient and not all offsprings may contribute for the prediction quality. Thus, we follow a wrapper scheme and quantify the significance of an offspring based on its prediction quality. As shown in Algorithm 2, the fit errors of each new offspring subset $S_{new} \subset \mathcal{F}$ is estimated using a regression algorithm, i.e., $error : S_{new} \mapsto e_{new}$. If an offspring subset outperforms the quality (fit errors) of an initial candidate subset S_i, i.e., $e_{new} < e_i \in \epsilon$, the particular offspring S_{new} replaces the worst performing set from $\mathcal{P}$. The corresponding fit error is updated as $\epsilon = \{\epsilon \setminus e_i\} \bigcup \{e_{new}\}$. The process of symmetric difference and quality check is repeated with the updated candidates (Lines 5-9). In parallel, $\mathcal{P}_{past}$ keeps track

of subsets of the previous iteration. The process iterates until none of offspring subsets outperforms the quality of subsets from previous iteration ($\mathcal{P}_{past}$). In this case, the Boolean variable *improvement* switches to $false$ and the algorithm ends.

Algorithm 2 Generation of quality constrained diverse subsets

Input: $\mathcal{P}, Y,$ and ϵ
1: Initialize $improvement = true$
2: **while** $improvement{=}true$ **do** ▶Iterate for every updated family of subsets
3: Set $\mathcal{P}_{past} = \mathcal{P}$
4: **for each** tuple$(S_d, S_e) \in \mathcal{P}$ **do**
5: $S_{new} = \{S_d\} \triangle \{S_e\}$
6: $e_{new} = error(S_{new})$
7: **if** $((\exists e_i \in \epsilon) > e_{new})$ **then** ▶If offspring outperforms initial subsets in $\mathcal{P}$
8: $\mathcal{P} = \{\mathcal{P} \setminus S_i\} \bigcup \{S_{new}\}$ such that, $error(S_i) = max(\epsilon)$
9: $\epsilon = \{\epsilon \setminus e_i\} \bigcup \{e_{new}\}$ ▶Replace the corresponding fit errors in ϵ
10: **end if**
11: **end for each**
12: **if** $(\mathcal{P}_{past} = \mathcal{P})$ **then** ▶If no improvement by offspring subset
13: $improvement = false$ ▶Reset bit to exit while loop
14: **end if**
15: **end while**
16: **Return:** $\mathcal{P}$

3.6 Unifying Multiple Subsets By Ensemble Regression

After having described our strategy for the selection of multiple subsets, our goal is to unify the information from each subset. The unified decision of DS3 is computed by following the idea of incremental learning [LP03]. Nevertheless, our framework enables the use of other techniques for unifying the individual hypothesis (e.g. using Dempster-Schafer Technique [FKZ15]). Incremental learning aims to learn from newly available data of the same source. For example, in weather forecasting applications, new data from temperature sensors is used to update the existing prediction models. Thus, the source of data, i.e., temperature sensor remains the same. For each of the newly available data from the temperature sensor, an ensemble is trained. The final predictions are formed by the combined decision of the

ensembles. Hence, the concept is based on building ensemble of ensembles [LP03]. We employ the same principle to combine the predictions of multiple subsets. For each $S_i \in \mathcal{P}$ we train an ensemble regression model. From each ensemble model, we obtain a set of predicted values of the target. The final predictions are the weighted average based on the relative errors (on training data) of each ensemble.

3.7 Time Complexity

To analyze the time complexity, we begin with the analysis of Algorithm 1. The algorithm starts with the computation of a $d \times k$ matrix. The run times for this step depends on the number and types of correlation measures used. The features are pruned based on the parameter α. In the worst-case, a user sets the parameter α to zero. This means, that Algorithm 1 does not prune the features. This leads to k identical sets with full dimensionality d. The fit errors of each subset has to be computed by a regression algorithm. In our experiments we use OLS which has a time complexity of $\mathcal{O}(d^2 \cdot N)$ [TSKK13]. Thus, the total complexity of Algorithm 1 is given by the worst time complexity of k correlation measures and the time complexity for estimating fit errors of k subsets, i.e., $\mathcal{O}(k \cdot d^2 \cdot N)$.

In Algorithm 2, the symmetric difference between two sets have a linear time complexity $\mathcal{O}(d)$. The symmetric difference between k subsets generate $\binom{k}{2}$ offspring subsets. The fit errors are computed for each of these newly generated offspring subset. Thus, the total complexity of Algorithm 2 is represented as $\mathcal{O}(\binom{k}{2} \cdot d^2 \cdot N)$. However, the worst-case scenario of $\alpha = 0$ will not generate any new subsets in Algorithm 2. In Section 3.8, experiments show that our algorithm ends in a fewer iterations with the best quality.

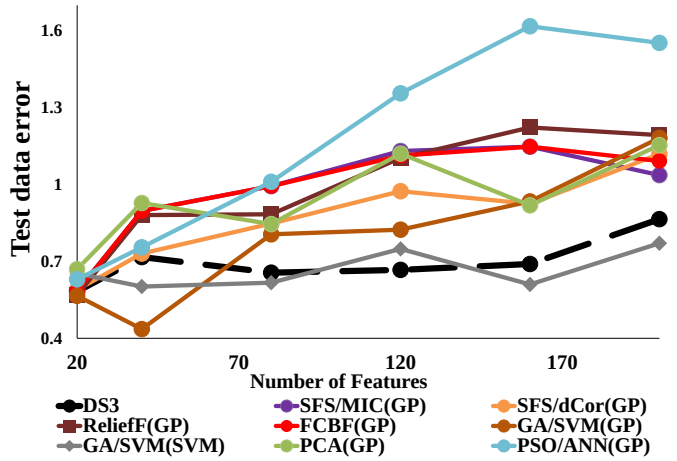
3.8 Experimental Evaluation

In this section we compare the quality and run times of our approach with several existing techniques using synthetic and real wold datasets. We consider different selection paradigms as competitors: filter-based approaches using forward selection (SFS) [MBN02] with the following cost functions: MIC [LMD+12] and dCor [SRB07], FCBF with symmetric uncertainty as cost function [YL03], ReliefF

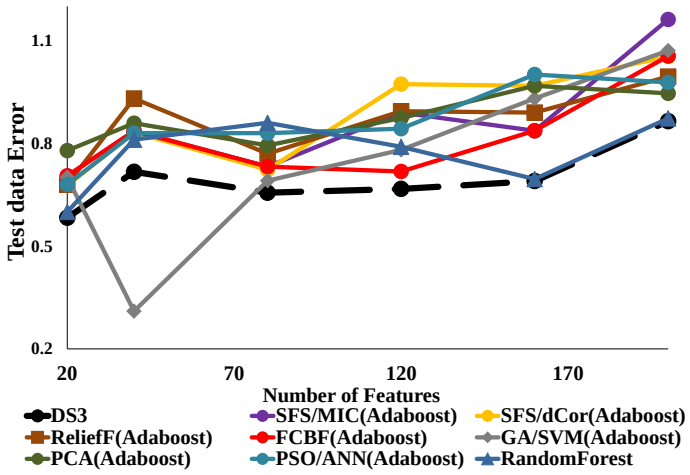
weights [RŠK03], wrapper techniques based on Genetic Algorithms (GA) [BALD14] with Support Vector Machines (SVM) [SV97] as cost function, a wrapper method with a more directed search strategy called Particle Swarm Optimization (PSO) [SISB11] with Artificial Neural Networks (ANN) as cost function and finally PCA which is a renowned technique for dimensionality reduction [Jol02]. After each feature selection algorithm, we apply two type of regression techniques: a single learner (sparse Gaussian processes [LG10]) and an ensemble learner (AdaBoost [FSA99] of Decision trees). As a techniques based on the paradigm of multiple projections, we consider the well-established Random Forest [Bre01].

Some of the aforementioned baselines require a user defined threshold value. For a fair comparison, we performed a grid search of the parameter for baselines and DS3 and only best results are used for comparison. Each data set is split such that 60% of the samples are used for selection/training and 40% for testing. As quality measure, we use the Root Mean Square Error normalized by the variance (σ^2) of the target instances. $NRMSE$ value of zero indicates that the predicted values follow the recorded values accurately with no deviations. The run times of experiments denote the feature selection summed up with training time of the regression algorithm.

Our approach requires a set of correlation functions and a wrapper method (Reg) (c.f. Section 3.4 and 3.5). In this work, we have considered the following four correlation functions: MI [CT12], dCor [SRB07], PCA [YYS05, Jol02] and ReliefF weights [RŠK03]. For faster computations, we chose Ordinary Least Squares (OLS) as the wrapper technique (Reg) for our hybrid approach. The OLS is used only as a wrapper within DS3 and not for estimating the final predictions. Unification the diverse subsets (c.f. Section 3.6) is to be done with weak learner capable of handling non-linear dependencies. Since the competitor approaches have been tested using boosted decision trees, we also employ them for unifying the results of multiple subsets as described in Section 3.6. A boosted decision tree is trained for each S_i and predictions are unified by weighted average of their errors [LP03].



(a) Single Learners vs. DS3



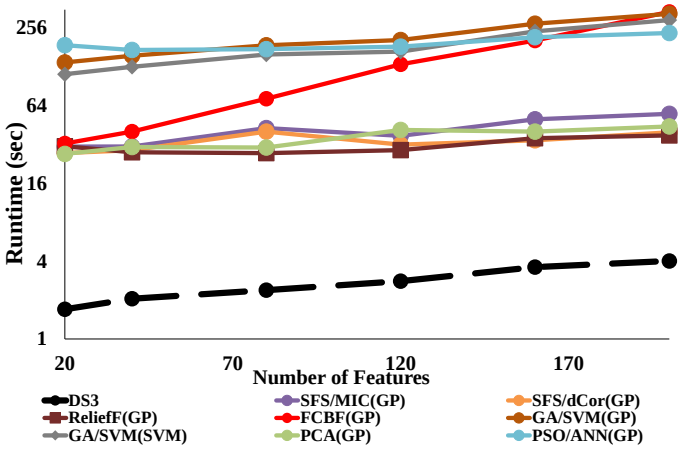
(b) Ensemble Learners vs. DS3

Figure 3.1: Quality (NRMSE) comparison of the competitor approaches vs. DS3 with increasing dimensionality (20, 40, 80, 120, 160, 200) and fixed database size of 1000 samples

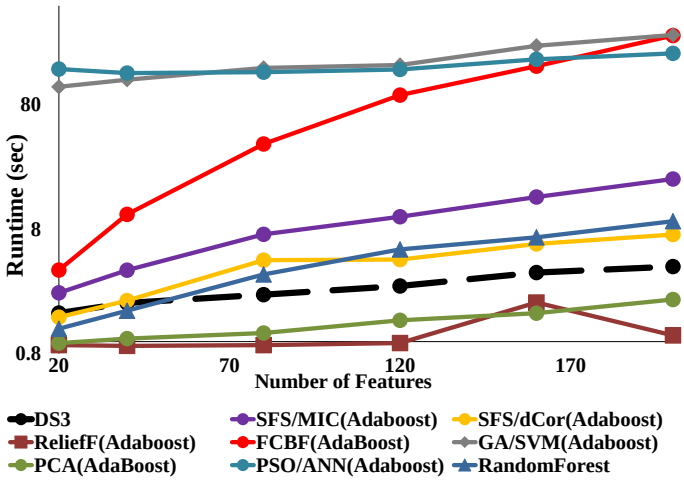
3.8.1 Synthetic Data sets

We analyze the efficiency of our approach w.r.t. the database size and dimensionality. The analysis have been performed on synthetic datasets. The synthetic data generation program of NIPS [NIP01] is employed to generate continuous feature sets with normal distribution and in any proportion of relevant features.

Figure 3.1 compares the quality between DS3 and existing competitors increasing the dimensionality, i.e., increasing the number of irrelevant variables. In these



(a) Single Learners vs. DS3

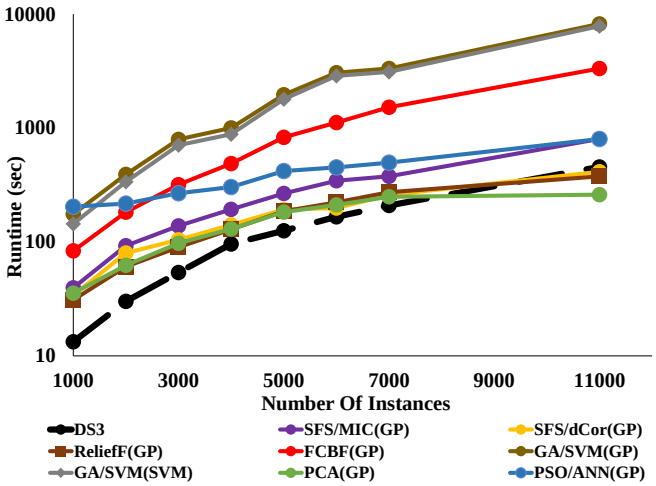


(b) Ensemble Learners vs. DS3

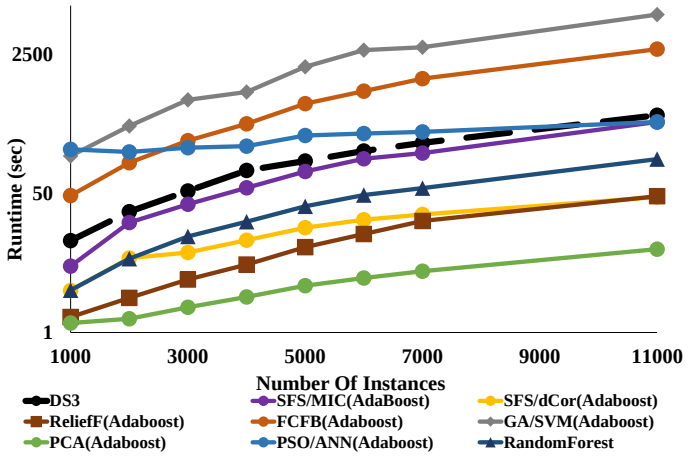
Figure 3.2: Run time (selection + training times) comparison of the competitor approaches vs. DS3 with increasing dimensionality (20, 40, 80, 120, 160, 200) and fixed database size of 1000 samples

experiments, DS3 achieves better results than traditional filter-based approaches and random forests that randomly select different projections of the data. Wrappers schemes (GA with SVM) obtain similar quality results as DS3, but the run times are significantly large in comparison to DS3. In particular, DS3 has been approximately 60 times faster in comparison to GA/SVM(SVM) (c.f. Figure 3.2).

Figures 3.2 and 3.3 show the run times of DS3 regarding increasing database size and dimensionality. Without significantly trading off the quality, DS3 shows to be



(a) Single Learners vs. DS3



(b) Ensemble Learners vs. DS3

Figure 3.3: Run time (selection + training times) comparison with increasing database size (1000, 2000, 4000, 6000, 8000, 12000) and fixed dimensionality of 80

efficient. PCA, ReliefF, SFS/dCor, SFS/MIC and RandomForest show better run times with higher prediction errors w.r.t. DS3. These approaches do not address diversity in the subset selection process. However, DS3 by applying symmetric difference aims to generate multiple diverse subsets and enhance diversity. Thus, DS3 consistently achieves better prediction accuracy (c.f. Figure 3.1) in comparison to these approaches.

Dataset	# features	# samples
Relative humidity [ZMRBRP14]	17	4137
Social media data [KDCGD13]	77	13000
Ailerons [Cam97]	40	14050
Stock exchange [Net11]	159	1813
Bosch	179	14537

Table 3.2: Properties of real world datasets used for experimental comparison of DS3 and other competitor approaches

3.8.2 Real world Data sets

The proposed selection strategy is tested on five different real world data sets from different areas of application (c.f. Table 3.2). The automotive data from Bosch is obtained from multiple sensor sources of a car and the objective is to predict a particular sensor's value. For confidentiality reasons, we do not provide explicit information about the Bosch data.

Quality Table 3.3 shows the results w.r.t. the prediction errors by each model. In comparison to the use of the entire feature space, we observe that the application of feature selection algorithms improves the quality. However, existing feature selection techniques do not show significant improvements in the prediction quality for the scenario of stock data. DS3, by exploiting the hidden local interactions of the variables has the best prediction accuracy. In the other application scenarios, feature selection algorithms increase the quality of the model, but the best results are obtained by our approach.

The wrapper scheme GA/SVM was adept in identifying the non-linearities of the data in comparison to other filter-based techniques. Thereby, it obtains the best prediction accuracies amidst the competitors in the social media, Bosch and ailerons data sets. For the humidity data set, we observe that SFS/dCor and DS3 have the same prediction accuracy, showing that the information from multiple views is not significant for such lower dimensional data sets. Overall, enhancing the diversity of the subsets and using them for a combined final hypothesis contributes for higher prediction quality.

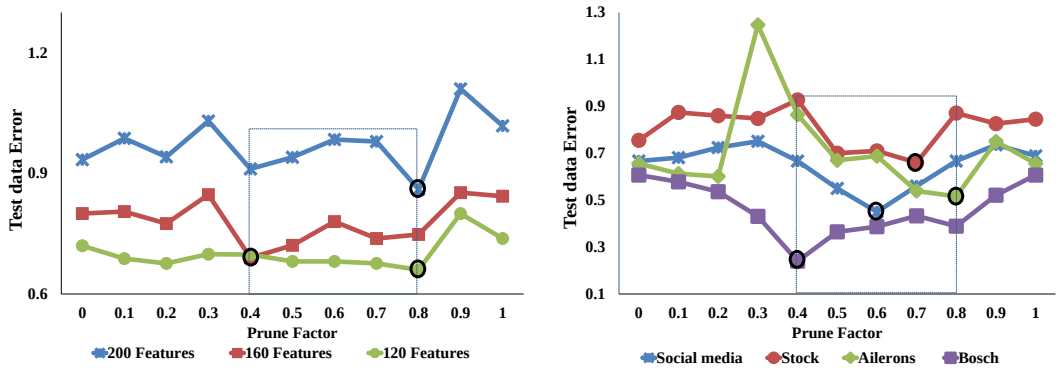
Feature Selection	Regression	Stock data	Social media data	Bosch data	Humidity data	Ailerons
Full Dimensional	GP	0.94	1.42	0.86	1.06	1.06
	Adaboost	0.83	0.66	0.86	0.61	0.78
SFS/MIC	GP	0.86	1.09	2.07	0.51	0.71
	Adaboost	0.85	0.66	0.40	0.55	0.73
SFS/dCor	GP	0.88	1.09	1.05	0.69	0.91
	Adaboost	0.84	0.68	0.39	0.44	0.82
Relieff	GP	0.88	0.93	4.49	0.83	0.85
	Adaboost	0.86	0.65	0.42	0.62	0.68
FCBF	GP	0.89	0.85	1.24	1.33	1.16
	Adaboost	0.86	0.65	1.22	0.60	1.17
GA/SVM	GP	0.87	0.97	0.64	0.83	0.69
	SVM	1.15	2.53	0.83	0.45	0.51
	Adaboost	0.87	0.51	0.3	0.65	0.65
PCA	GP	0.85	0.93	1.16	1.19	1.12
	Adaboost	0.86	0.65	0.95	0.89	0.98
PSO/ANN	GP	0.83	0.92	2.41	0.99	1.25
	Adaboost	0.85	0.76	1.08	0.54	0.98
Random Forest		0.86	0.52	0.74	0.44	0.62
DS3/OLS	Decision Tree	**0.66**	**0.45**	**0.24**	**0.44**	**0.51**

Table 3.3: Comparison of prediction errors (NRMSE) of competitor techniques versus DS3 on real-world datasets on test data

	Run time in seconds					
Feature Selection	Regression	Stock data	Social media data	Bosch data	Humidity data	Ailerons
Full Set	GP	87.83	822.76	956.32	151.38	386.45
	Adaboost	4.21	6.07	26.27	1.06	2.83
SFS/MIC	GP	91.38	852.45	7392.92	363.05	361.8
	Adaboost	29.11	260.32	6992.58	235.13	7.27
SFS/dCor	GP	149.98	1111.47	293.42	238.49	665.37
	Adaboost	27.49	157.49	242.02	20.75	50.27
Relieff	GP	107.41	527	879.96	144.93	423.02
	Adaboost	10.68	136.37	347.73	6.24	43.59
FCBF	GP	459.37	762.43	576.17	149.25	434.1
	Adaboost	375.35	78.68	99.24	11.11	5.92
GA/SVM	GP	175.7	874.35	4765.87	728.31	2051.69
	SVM	62.28	380.53	4276.52	589.98	1666.75
	Adaboost	65.89	383.58	4279.02	590.78	1667.69
PCA	GP	117.32	598.88	549.01	141.55	434.55
	Adaboost	2.15	19.25	6.49	1.13	5.26
PSO/ANN	GP	1492.16	5949.19	5898.85	4025.28	3598.13
	Adaboost	1421.86	5467.89	5454.19	3882.26	3138.18
Random Forest		4.7	67.9	117.66	2.21	17.40
DS3/OLS	Decision Tree	**41.11**	**385.03**	**824.31**	**22.13**	**65.66**

Table 3.4: Comparison of run times of competitor techniques versus DS3 on real-world datasets

Run times Table 3.4 shows that DS3 is highly efficient than the conventional iterative and evolutionary search strategies. We observe that PCA achieves the best run times. However, the prediction errors of PCA are relatively higher in comparison to DS3. PCA prunes the feature based on the postulate that principal



(a) Artificial Data

(b) Real world data sets

Figure 3.4: Analysis of the influence of α on test data with different dimensionality. The circled points denote the minimal test data error

components with larger variance represent intriguing structure, while those with lower variances are not significant. Such a strong assumption cannot not be applicable for all datasets [Shl14]. Overall, DS3 prediction model achieves efficient results w.r.t. both quality and run time because of breaking down the complexities of the feature space with the multiple diverse subsets.

Parameter Analysis

DS3 has one user defined parameter, i.e., the prune factor α. We experimentally analyze the influence of α on prediction errors on artificial and real world datasets. As discussed in Section 3, α=1 will choose only the features with maximum correlation as initial candidates. With such small subsets, Algorithm 2 has fewer tuples of feature subsets (for computing symmetric difference) that cannot generate new diverse offspring subsets. On the contrary, α=0 tends to choose the full-dimensional feature set which is inefficient. Figure 3.4 shows that choosing only features with best correlation does not ensure best prediction accuracies ($\alpha = 1$). Likewise, choosing all the features as initial candidates is also not contributing to improved prediction quality ($\alpha = 0$). However, performing grid search over all values of α is tedious. From experiments on artificial datasets (c.f. the highlighted window of Figure 3.4a), we observe that α values in the range [0.4,0.8] has comparatively lower test errors.

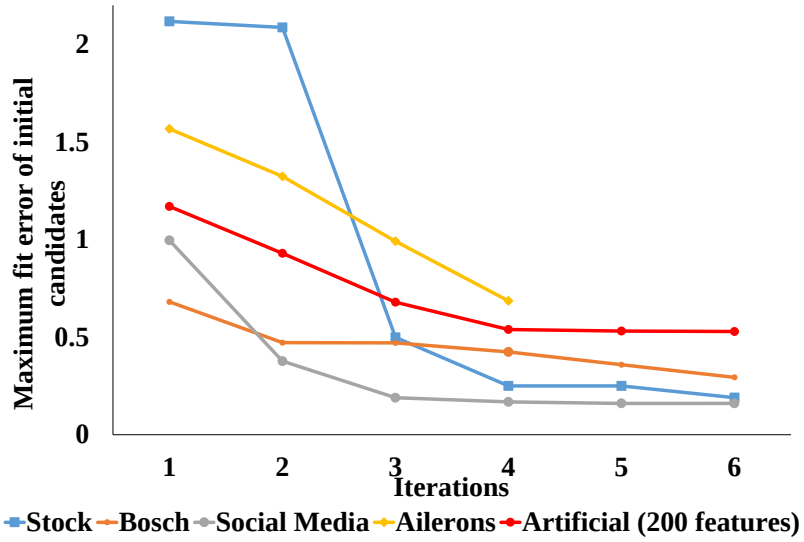


Figure 3.5: Maximum fit error (using OLS) of initial candidates in each iteration of the symmetric difference search space: α= 0.9

In Figure 3.4b, we show that the range is also practically applicable for real world datasets. A factor less than 0.4 includes many noisy features as initial candidates. On the other hand, a factor greater than 0.8 generates small subsets with highly relevant features as initial candidates. Thus, the symmetric difference search space is also minimal. $\alpha \in [0.4, 0.8]$ gives reasonable initial candidates for which enhancing the diversity improves the prediction quality.

Iterations

DS3 prunes the feature space by adherence to multiple correlation measures and evaluate the symmetric difference search space. Figure 3.5 shows maximum fit error (i.e. $max(\epsilon)$) of the initial candidates for each iteration. In each iteration, the subsets are replaced by new offspring subset that have lower fit errors (Line 5-9 of Algorithm 2). When none of the new subsets improve the prediction quality, the algorithm ends. Figure 3.5 shows the maximal iteration up to which the offsprings were outperforming the initial candidates. We observe that DS3 converges in a fewer iterations and contributes for efficiency.

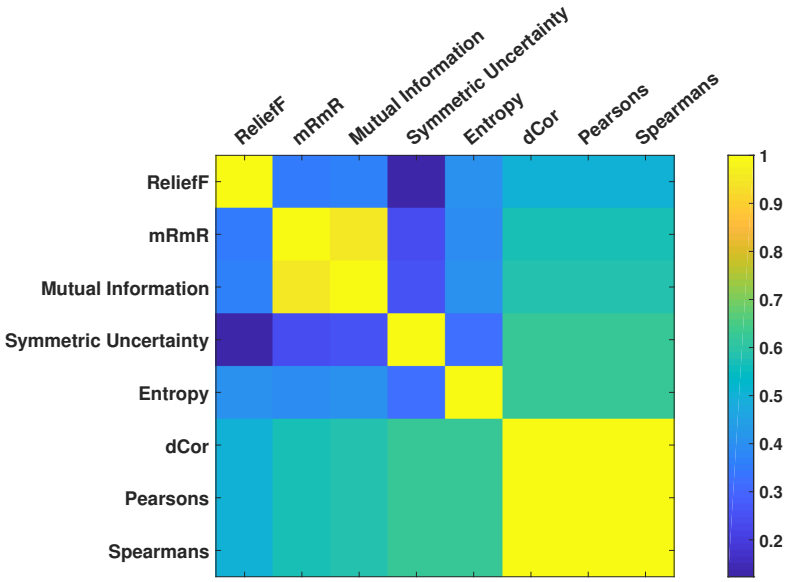


Figure 3.6: Feature ranking of different correlation measures. 1 denotes that the ranking is exactly the same and 0 denotes extreme dissimilarity in feature ranking

Differences of multiple correlation measures

Our proposed approach Diverse Selection of Feature Subsets for Ensemble Regression (DS3) in this chapter relies on enhancing the diversity (c.f. Algorithm 2) of feature subsets based on the initial candidates chosen in Algorithm 1. To enhance the subset diversity it is necessary to choose correlation measures that are capturing different intrinsic properties. In this section we aim to show how different are the features ranked based on different correlation measures. In our experiment we chose various correlation measures such as, ReliefF, mRmR, Mutual Information, Symmetric Uncertainty, Enropy, dCor, Pearsons and Spearmans. All features are ranked using these different correlation measures. The concordance of feature ranks are visualized in Figure 3.6. For instance, the ranking of features using dCor and Pearsons are identical (denoted by yellow color code). On contrary, the feature rankings of ReliefF and Symmetric Uncertainty are highly different. We exploit this property of the correlation measures in our work to select initial candidates (c.f. Algorithm 1) and latter enhance the subset diversity using symmetric difference (c.f. Algorithm 2).

3.9 Summary

In this chapter, we proposed a novel heuristic called DS3 for the selection of multiple relevant views of the feature space. DS3 exploits multiple correlation measures to capture different intrinsic properties in a high-dimensional feature space. Using the feature space pruned by different correlation measures, we enhance the algorithm to evaluate multivariate correlations by generating multiple feature combinations. The generation process is subjected to our simple and efficient diversity criterion. The diversity criterion is aimed at capturing not only relevant but also complementary views of the feature space.

From experimental evaluation on synthetic datasets, we show that DS3 scales better in comparison to wrappers and several filter-based feature selection techniques. We also showed that using a single view of the feature space selected by one correlation measure do not always have the best prediction accuracies. This corroborates the importance of diversity enhancing multiple subsets search strategy. Considering such multiple subsets enhance the prediction model by combining the underlying patterns hidden in multiple views of the high-dimensional feature space. Our experiments on real world data show that the proposed heuristic is efficient and has improved prediction quality in comparison to several state of the art techniques. Hence, we aim to retain the idea multiple views and include further enhancements such as,

- Improved efficiency by adhering to filter-based paradigm.
- Handle correlation analysis for mixed datasets, i.e., continuous and categorical features.
- Identify multivariate interactions, i.e., higher-order interactions, between more than two features.
- Estimate the feature redundancy during the correlation analysis.

In the forthcoming chapter, we will address all the four aforementioned enhancements.

Chapter 4

Multivariate Relevance and Redundancy Scoring in Mixed dataset

4.1 Motivation

In automotive applications, the data from several sensors (continuous values), status bits, gear-position (categorical values) and calculations forms a mixed dataset with a large number of features. In such a feature space, a set of features interact amongst themselves and these interactions are strongly correlated to the target class. For example, engine-temperature and fuel quality are two essential features required to predict engine-performance. On analyzing its individual correlations to the target, each feature is weakly correlated to the engine's performance. However, engine-performance is a combined outcome of engine-temperature and fuel quality. That is, their interactions contribute to the target predictions when used together. In such cases, assigning low relevance scores based on individual correlations is misleading. Hence, to draw conclusions on the relevance of engine temperature, it is necessary to assess its role in multiple subspaces. In addition to the multi-feature interactions, some features may have redundant information. Following our automotive example, certain signals are measured or calculated multiple times in a vehicle for safety reasons. These redundant signals provide similar information, but are not necessarily identical. In such a scenario, two redundant features have the same magnitude of relevance to the target class. However, using both features for a prediction model is unnecessary as they provide similar information. Elimination of redundant features reduces the computational load and enhances the generalization ability of the classifier [PLD05]. All aforementioned problems are motivated with examples from our application, but they exist in several other domains such as Bio-informatics [DP05] and Media [CL15].

The first challenge lies in estimating the feature relevance based on interactions between the features and the target. Evaluating all possible feature combinations for these interactions results in an exponential runtime w.r.t. the total number of features. Thus, it is necessary to perform the evaluations in an efficient way. The second major challenge lies in measuring the redundancy of each feature while still acknowledging its relevance w.r.t. the target class. A final challenge is to evaluate relevance and redundancy in mixed feature space. To emphasize its pertinence, Figure 4.1 shows that a large number of datasets in UCI repository have mixed data types. Nevertheless, existing filter-based feature selection methods [GE03,

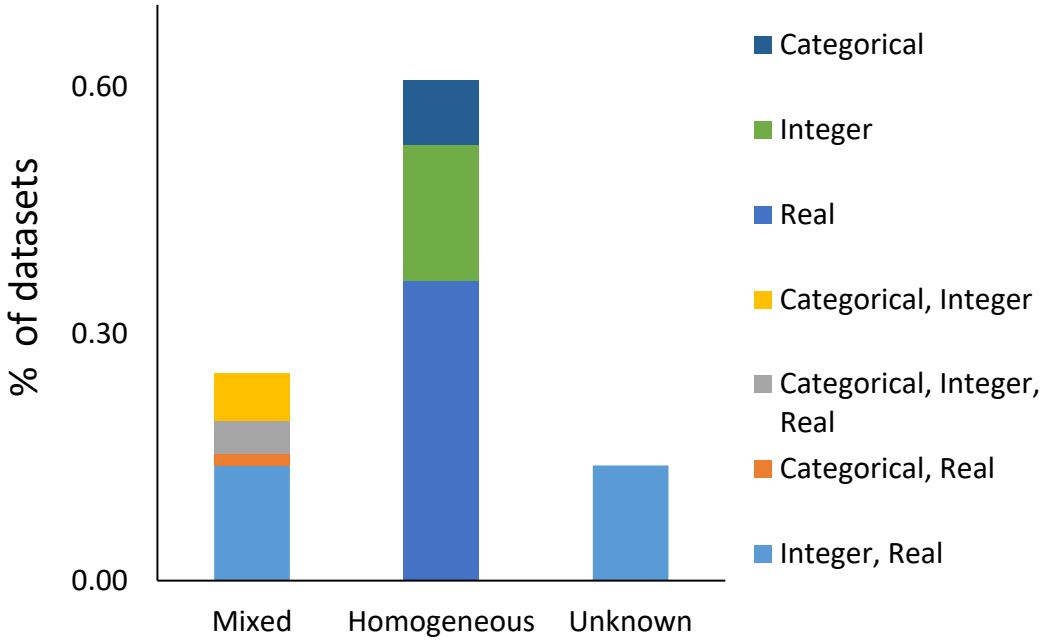


Figure 4.1: Proportion of datasets (in percentage) with different data types in UCI repository as on October 2019. Where total number of datasets were 488 and 68 datasets did not have the data type defined in the repository summary

PLD05, Hal00, TM07] do not focus on considering all three challenges together: relevance based on multi-feature interactions, redundance and mixed data.

In this chapter, we propose a feature ranking framework (RaR)* to address all three challenges. We begin with computing relevance scores of multiple subspaces. These subspace relevance scores are decomposed to evaluate the individual feature contributions. In order to include the multi-feature interactions, the relevance of a feature is computed based on these individual contributions to multiple subspace relevance scores. The relevance estimation is followed by the redundance calculation. The relevance and redundancy scores are unified such that the relevance of a feature is penalized based on its redundancy. The major contributions of the paper are as follows:

*Adapted by permission from Springer Nature: Including Multi-feature Interactions and Redundancy for Feature Ranking in Mixed Datasets in the proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), 2017 [SBS+17]

(1) A feature relevance score, that considers the multi-feature interactions.
(2) A measure of redundancy to evaluate the novelty of a feature w.r.t. a subset.
(3) Experimental studies on both synthetic and real world datasets to show that several state-of-the-art approaches underestimate the importance of such interacting features.
Our extensive experiments show that our approach has better ranking quality and lower run times in comparison to several existing approaches.

4.2 Comparison to Related Work

Paradigm	Approach	Mixed data	Redundancy	Feature interactions	Efficiency
Wrapper	SFS [TPKC10]	✓	✓	✗	✗
	Recursive elimination [TPKC10]	✓	✓	✓	✗
Hybrid	MFS [TM07]	✓	✗	✗	✗
	Doquire [DV11]	✓	✓	✗	✗
Subspace Ranking	HiCs [KMB12]	✗	✗	✓	✓
Embedded	C4.5 [Qui14]	✓	✓	✗	✓
Filter	mRmR [PLD05]	✓	✓	✗	✓
	CFS [Hal00]	✓	✓	✗	✓
	RaR	✓	✓	✓	✓

Table 4.1: Comparison of RaR with other relevant literatures from different feature selection paradigms

We discuss feature selection in (1) mixed data, (2) by including higher-order interactions and (3) redundancy. In Table 4.1 we group literatures from various paradigms such as filters, wrappers, hybrid, embedded and unsupervised subspace ranking (c.f. Chapter 2). Our RaR algorithm follows the filter-based feature selection paradigm. By performing pairwise feature-feature and feature-target correlation analysis, most of the traditional approaches such as [Hal99, Bre01, YL03, DP05, SM11] fails to address the problem of feature interactions between more than two features. Although recursive elimination can handle higher-order interactions, as discussed in Section 2.2, they demand high computation times. As a filter-based framework, RaR is highly efficient in comparison to iterative techniques. Unlike the work of [TM07], we perform feature ranking by also evaluating the redundancy of information. Overall, efficient ranking of features based on its relevance by including its higher-order interactions and redundancy in mixed data are the value addition that we provide in this chapter.

The idea of decomposing the subspace scores into individual feature scores resonates with the idea of Shapley value (SV) regression. Given a feature space $\mathcal{F}$, SV of a single feature $f \in \mathcal{F}$ is the average usefulness of all possible linear models built using the subsets that contain the feature. The usefulness of the model is measured by the R^2 value [LC01, PHHN16]. However, the two ideas aim to solve different problems with different approaches. SV aims to identify the relative importance of individual features in a linear regression problem with a collinearity setting. Moreover, SV applies for a linear regression models and not for non-linear cases [LL17]. Recent works have proposed the extension of SV to non-linear models [Jos19]. However, replacing the subspace relevance score $rel(S)$ with SV (with the linear or non-linear variant) decomposition is not ideal as the latter requires to fit multiple regression models for each iteration [ŠKŠ09]. To avoid training of multiple models, one paper proposes the calculation of SV on the conditional expectations of the model [ŠK14]. However, they assume that the features are distributed uniformly and independently [SN19]. On the other hand RaR aims to solve the problem of feature ranking for datasets with higher-order interactions. For this we propose a heuristic that provides an estimate of a feature's importance based on its role in multiple subspaces. In addition, as a filter-based approach RaR is independent of classification or regression algorithms for feature ranking.

4.3 Problem Definition

In this section, we define the problem that we aim to solve. Having formalized the definitions of a categorical $f \in \mathcal{F}_N$ and continuous feature $f \in \mathcal{F}_C$ in Chapter 2, let $\mathcal{F}$ be a d-dimensional mixed dataset. Such that, $f_j \in \mathcal{F} \mid j = 1, \cdots, d$ and $\mathcal{F} = \{\mathcal{F}_C \cup \mathcal{F}_N\}$ with N instances. As a supervised learning process, the target Y is a collection of discrete classes.

In the following, we denote $error : S \mapsto \mathbb{R}$ as the error function of the classifier, trained using a subset of features $S \subseteq \mathcal{F}$. For the given mixed dataset, we aim to: (1) Compute feature relevance by including their interactions with other features. (2) Evaluate the redundancy score of each feature.

Evaluation of feature interactions requires a multivariate correlation measure, that quantifies the relevance of S to Y. Given such a subspace relevance score $rel : S \mapsto \mathbb{R}$, which is a function of individual feature relevancies, i.e.,

$$rel(S) \;=\; \phi(\{r(f_j) \mid \forall f_j \in S\}),$$

where ϕ is an unknown function such that $\phi : \mathbb{R}^{|S|} \mapsto \mathbb{R}$. To infer the individual feature relevancies $r : f_j \mapsto \mathbb{R}$, the first challenge is to decompose the subspace scores into individual feature scores. However, individual feature relevance cannot be inferred from a single feature subset because of possible interactions of f_j in other subspaces. To include the multi-feature interactions, it is necessary to evaluate M different subspaces. Thus, we aim to deduce a valid relevance score of a feature $r(f_j)$, based on the contribution of f_j to M different subspace scores.

Additionally, we aim to estimate the redundance of information a feature has, w.r.t. a subspace, i.e., $red : (f_j, S) \mapsto \mathbb{R}$. Given a feature $f_i \in S$ that is non-redundant to $S \setminus f_i$ and $f_j \in S \mid i \neq j$ with redundant information to $S \setminus f_j$, we intend to quantify a redundance score such that $red(f_j, S) > red(f_i, S)$. Addition of redundant feature information to a classifier does not contribute to the prediction quality, i.e., $error(S) \approx error(S \setminus f_j)$ [PLD05]. A major challenge for filter-based feature selection approaches is to evaluate this efficiently without training a classifier. Finally, the features are ranked based on the unification of two scores.

4.4 Subspace relevance

In the following, we introduce the definition of subspace relevance and a method to calculate it. To estimate the relevance of a subspace to the target, we use the concept of conditional independence. For an uncorrelated subspace, the law of statistical independence is not violated. The degree of violation is quantified by measuring the difference between the conditional and marginal distributions [NPBT07, KMB12].

Definition. 4.1: Subspace Relevance

Given a subspace $S \subseteq \mathcal{F}, |S| = k$ and a divergence function div, the subspace relevance score $rel(S)$ to the target Y is defined as:

$$rel(S) = E_S\Big[div\Big(p(Y \mid S \in [s_1, \cdots, s_k]) \mid\mid p(Y)\Big)\Big].$$

For a set of discrete target classes Y, the marginal of the target is compared to its conditional distribution. This definition enables the measuring of multivariate and non-linear correlations [KMB12] in mixed datasets. For $f_j \in \mathcal{F}_C$, the conditional is estimated based on a slice of continuous instances drawn from f_j. Similarly, for a $f_j \in \mathcal{F}_N$, the conditional is based on a slice of instances that have a particular categorical state. The magnitude of divergence between these two distributions can be estimated with Kullback–Leibler (KLD) or Jensen-Shannon divergence functions [Lin91]. As a non-symmetric measure $KLD(p(Y \mid S) \mid\mid p(Y))$ is not equal to $KLD(p(Y) \mid\mid p(Y \mid S))$. We instantiate RaR with $KLD(p(Y \mid S) \mid\mid p(Y))$ based on Lemma 4.1.

Lemma 4.1. *Instantiating the KL-Divergence function as $KLD(p(Y \mid S) \mid\mid p(Y))$ is equivalent to mutual information and $KLD(p(Y) \mid\mid p(Y \mid S))$ is not.*

Proof: Given a subspace S and a target Y, $KLD(P(Y|S)||P(Y))$ converges to mutual information. As a measure of statistical dependence between S and Y, we estimate the divergence between the distribution of Y and that of Y under certain conditional slice of S. We follow the adaptive slicing methodology where the final relevance score is based on the expected divergence between the marginal and several conditional slices [KMB12]. That is,

$$rel(S) = E_S[KLD(P(Y|S) \mid\mid P(Y))].$$

Hence, it is sufficient to prove that $E_S[KLD(P(Y|S) \mid\mid P(Y))]$ converges to $MI(S, Y)$.

$$E_c[KLD(P(Y|S) \mid\mid P(Y))]$$
$$= \sum_S P(S) \sum_Y P(Y|S) log\left(\frac{P(Y|S)}{P(Y)}\right)$$

$$= \sum_S \sum_Y P(S)P(Y|S)log\left(\frac{P(Y|S)}{P(Y)}\right)$$

Rewrite $P(Y|S)$ $P(S)$ as $P(S,Y)$

$$\implies \sum_S \sum_Y P(S)P(Y|S)log\left(\frac{P(Y|S)}{P(Y)}\right) = \sum_S \sum_Y P(S,Y)log\left(\frac{P(Y|S)}{P(Y)}\right)$$

Multiplying and dividing the above equation by $P(S)$,

$$\sum_S \sum_Y P(S,Y)log\left(\frac{P(Y|S)P(S)}{P(Y)P(S)}\right).$$

Rewrite $P(Y|S)$ $P(S)$ as $P(S,Y)$

$$\sum_S \sum_Y P(S,Y)log\left(\frac{P(S,Y)}{P(Y)P(S)}\right)$$

$$= MI(S,Y)$$

□

4.5 Decomposition For Feature Relevance Estimation

A simple solution to estimate the relevance of f_j using Definition 4.1 is by computing $rel(\{f_j\})$. Such individual feature relevance scores lacks information about feature interactions. The aim of our approach is to evaluate feature relevance $r(f_j)$ by including its interactions with other features and not to compute subspace scores $rel(S)$. The subspace relevance score represents the contribution of all features present in the subspace. Hence, the subspace score can be seen as a function of individual feature relevancies. We estimate the feature relevance $r(f_j)$ by decomposing the subspace score, which is the result of individual feature relevancies.

Example 4.1. Assume a dataset $\mathcal{F} = \{f_1, f_2, f_3, f_4\}$, such that there exists multi-feature interactions between $\{f_1, f_2, f_3\}$. Hence, relevance of a subset with all

interacting features ($rel(S_1) \mid S_1 = \{f_1, f_2, f_3\}$) is greater than the relevance of a subset ($rel(S_2) \mid S_2 = \{f_1, f_2, f_4\}$) with an incomplete interactions.

A naïve decomposition is to decompose $rel(S)$ as the sum of individual feature relevancies. On applying naïve decomposition to our Example 4.1, we obtain $rel(S_1) = r(f_1) + r(f_2) + r(f_3)$ and $rel(S_2) = r(f_1) + r(f_2) + r(f_4)$. With an incomplete interaction structure, $rel(S_2)$ will underestimate the values of $r(f_1)$ and $r(f_2)$. Such underestimations are misleading as there exists another subspace where f_1 and f_2 in combination with f_3 forms a complete interaction structure to be more relevant to Y. This necessitates to rewrite the decomposition rule, such that it holds true for both cases. Hence, we define the decomposition as an upper bound of the subspace relevance.

Definition. 4.2: Feature Constraint

Let $r(f_j) \in \mathbb{R}$ be the relevance of individual features within the subspace $S \subseteq \mathcal{F}$, we define the feature constraint as:

$$rel(S) \leq \sum_{f_j \in S} r(f_j)$$

Theoretical justification for the constraint

In order to estimate the multi-feature interactions, we decompose the relevance score $rel(S)$ into contributions of each subset combination $T \in 2^S$. To perform this decomposition, let us define the contributions of the subset combinations. The contribution of T strictly denotes the influence of that particular combination of features and does not account for any other subsets of T. Hence, the contribution is formulated as,

$$c(T) = rel(T) - \sum_{Q \subset T} rel(Q). \tag{4.1}$$

Therefore, our relevance score can be decomposed using the contribution function as,

$$rel(S) = \sum_{T \in 2^S} c(T). \tag{4.2}$$

For example, for $S = \{f_1, f_2, f_3\}$,

$$rel(S) = c(\emptyset) + c(f_1) + c(f_2) + c(f_3) + c(f_1, f_2) + c(f_1, f_3) + c(f_2, f_3) + c(f_1, f_2, f_3).$$

The relevance of a set S, i.e., $rel(S)$, increases as the contribution of a specific subset combination T increases. Hence, the relevance of the set can never be greater than the imposed contribution of all its subsets. However, two features can provide redundant contributions to the relevance score. To handle information redundancy, we rewrite Equation 4.2 as,

$$rel(S) \leq \sum_{T \in 2^S} c(T) \tag{4.3}$$

and our example is reformulated accordingly,

$$rel(S) \leq c(\emptyset) + c(f_1) + c(f_2) + c(f_3) + c(f_1, f_2) + c(f_1, f_3) + c(f_2, f_3) + c(f_1, f_2, f_3).$$

The contribution $c(f_1)$ stands for the relevance of f_1. Combination of feature f_1 with other features, e.g., f_2, can be more influential for the $rel(S)$ score due to multi-feature interactions and this is captured in $c(f_1, f_2)$. As a empty subset has no influence for the relevance score $c(\emptyset) = 0$, we avoid using it in our forthcoming formulations.

Using our inequality defined in 4.3, we deduce a relevance score for a feature $f \in \mathcal{F}$ as a weighted sum of contributions where f is a part of,

$$r(f) = \sum_{T \in 2^{\mathcal{F}} \wedge f \in T} \frac{1}{\mid T \mid} \cdot c(T). \tag{4.4}$$

The weighted sum ensures that the contribution of a feature subset is equally distributed to all features in the cluster. However, we are not interested in estimating the feature contributions but the feature relevance by honoring the multi-feature interactions.

For any randomly drawn feature subset S in a Monte-Carlo iteration, our interest lies in estimating the relevance of each feature $f \in S \mid S \subseteq \mathcal{F}$. In Equation 4.5, we show that the relevance of a feature is the weighted sum of contributions of different subset combinations of S where f is a part of. In addition, it can

also have an incomplete multi-feature interactions that are not entirely captured by the subset S. These interactions that are mathematically denoted as the term highlighted in red.

$$\sum_{f\in S} r(f) = \sum_{f\in S} \sum_{T|T\in 2^S \wedge f\in T} \frac{1}{|\,T\,|}\cdot c(T) + \sum_{f\in S} \sum_{T|T\notin 2^S \wedge f\in T} \frac{1}{|\,T\,|}\cdot c(T) \tag{4.5}$$

$$\geq \sum_{f\in S} \sum_{T|T\in 2^S \wedge f\in T} \frac{1}{|\,T\,|}\cdot c(T)$$

$$\geq \sum_{T\in 2^S} \sum_{f\in T} \frac{1}{|\,T\,|}\cdot c(T)$$

$$\geq \sum_{T\in 2^S} c(T).$$

$$\implies \sum_{f\in S} r(f) \geq \sum_{T\in 2^S} c(T) \geq rel(S) \text{ (c.f. Equation 4.3)}$$

$$\implies rel(S) \leq \sum_{f\in S} r(f)$$

The defined inequality applies for a subspace with a complete or an incomplete interaction structure. The relevance of a feature f_j is to be estimated based on multiple subspaces, i.e., $S \mid S \in 2^{\mathcal{F}}$ and $f_j \in S$. Hence, a single inequality is not sufficient to estimate feature relevance based on multi-feature interactions. Moreover, a single inequality does not enable us to compute the relevance of all features $f_j \in \mathcal{F}$ in the high-dimensional feature space. However, it is computationally not feasible to deduce constraints (c.f. Definition 4.2) for all possible feature combinations. We address this challenge by running M Monte Carlo iterations. For each iteration, we select a subspace S and define a constraint based on the subspace relevance $rel(S)$ score and the features belonging to S. The constraints provide information on how a feature interacts in multiple subspaces. From these constraints, we aim to estimate the relevance of a feature $r(f_j)$.

Table 4.2 shows an illustrative example of how our idea of generating constraints works for a dataset (in Example 4.1) with multi-feature interactions . Our approach draws several random subspaces as shown in Table 4.2.

With the calculated subspace relevancies, we build 3 constraints for estimating the bounds of the individual feature relevance. The constraints of $i = 2$ and 3

i	S	$rel(S)$	Constraint
1	$\{f_1, f_2, f_3\}$	0.9	$r(f_1) + r(f_2) + r(f_3) \geq 0.9$
2	$\{f_1, f_4\}$	0.12	$r(f_1) + r(f_4) \geq 0.12$
3	$\{f_2, f_1, f_4\}$	0.15	$r(f_1) + r(f_2) + r(f_4) \geq 0.15$

Table 4.2: Illustrative example of feature constraints for 3 Monte Carlo iterations

underestimate the relevance of the individual features. However, constraint of $i = 1$ increases the boundaries of individual feature relevance. The relevance of a feature $r(f_j)$ is decided by considering multiple subspaces where f_j is a part of. Hence, our approach prevents underestimation of $r(f_1)$ and $r(f_2)$ and enable inclusion of multi-feature interactions. In addition, we also deduce that any feature subset $S \subseteq \mathcal{F}$ that contains $\{f_1, f_2, f_3\}$ will have a high relevance score $rel(S)$. This inference is deduced because the multivariate relevance function $rel(S)$ is monotone.

Lemma 4.2. *As a monotonic score, adding an irrelevant feature f to a set S of relevant features does not lead to loss of information, i.e., $rel(S) \leq rel(S \cup f)$.*

Proof: As the expectation of divergence between conditional and marginal distribution converges to mutual information (c.f. Lemma 4.1), based on the literature from information theory, we can say that $rel(S)$ is monotonic [LSLZ09]. Moreover, the work of [Ryu93] also proves that as the data information gets richer, the KL-divergence measure increase monotonically. We show a simple proof for Lemma 4.2 using a feature subset S and a target Y. Mutual information between a subset of variables and a target can be written as [CT12],

$$MI(S, Y) = KLD(P(S, Y)||P(S)P(Y)).$$

Add a noisy feature f, that is statistically independent to S.

$$\implies KLD(P(S \cup f, Y)||P(S)P(f)P(Y))$$

$$= \sum_{S,f,Y} P(S \cup f, Y)\ log\left(\frac{P(S, f, Y)}{P(S)P(f)P(Y)}\right)$$

Using independence of f,

$$\sum_{S,f,Y} P(S \cup f, Y)\ log\left(\frac{P(S,f,Y)}{P(S)P(f)P(Y)}\right) = \sum_{S,f,Y} P(S,Y)\ P(f)\ log\left(\frac{P(S,Y)\ P(f)}{P(S)P(f)P(Y)}\right)$$

$$= \sum_{S,f,Y} P(S,Y)\ P(f)\ log\left(\frac{P(S,Y)}{P(S)P(Y)}\right)$$

$$= \sum_{S,Y} P(S,Y)\ log\left(\frac{P(S,Y)}{P(S)P(Y)}\right) \sum_{f} P(f).$$

As we know that $\sum_f P(f) = 1$,

$$\sum_{S,Y} P(S,Y)\ log\left(\frac{P(S,Y)}{P(S)P(Y)}\right) \sum_{f} P(f) = \sum_{S,Y} P(S,Y)\ log\left(\frac{P(S,Y)}{P(S)P(Y)}\right)$$

$$= KLD(P(S,Y)||P(S)P(Y))$$

□

Our approach generates M inequalities for M Monte Carlo iterations. Solving the system of M inequalities does not lead to a unique value of $r(f_j)$. The inequalities provide only the boundaries for feature relevancies. We aim to deduce a reasonable estimate of the relevancies such that all constraints are satisfied. As these constraints denote the lower bounds of the feature relevancies, we aim to minimize the contributions of individual features. Therefore, we define an objective function that estimates $r(f) \mid f \in \mathcal{F}$ subject to the defined constraints,

$$\min_{r(f)} \left[\sum_{f\in\mathcal{F}} r(f) + \sum_{f\in\mathcal{F}} (r(f) - \mu)^2\right] \text{s.t. } rel(S_i) \leq \sum_{f\in S_i} r(f) \mid i = 1, \cdots, M, \tag{4.6}$$

such that, $\mu = (1/|\mathcal{F}|) \sum_{f\in\mathcal{F}} r(f)$. The first term denotes the sum of individual feature relevance. The second part of the optimization function is a standard L2-regularization term to ensure that all relevancies $r(f)$ contribute equally to the boundary. Finally, we apply quadratic programming in order to optimize Equation 4.6 subject to the M affine inequalities. The inequalities define a feasible region in which the solution to the problem must be located for the constraints to be satisfied. Thus, we obtain the relevance score for each feature. Computing the subspace relevance (c.f. Definition 4.1) for each iteration requires the estimation

Algorithm 3 Estimation of Feature Relevance

Input: $\mathcal{F}, Y, M, k$

$C = \emptyset$
for $i = 1 \rightarrow M$ **do**
    Sample $\{S_i \mid S_i \subseteq \mathcal{F} \wedge |S_i| \leq k\}$
    Compute $rel(S_i)$ using Definition 4.1
    Construct constraint (cf. Definition 4.2)
    Add constraint to set C
end for
Optimize objective function Equation 4.6 subject to C
return $r(f) \mid \forall f \in \mathcal{F}$

of conditional probability distributions. However, evaluating the empirical conditional probabilities for large $|S|$ is inaccurate. We demonstrate this by empirical evaluation in Section 4.10.1. Hence, it is necessary to restrict the size of the subspace to a maximum of k. That is, each randomly drawn $S_i \mid S_i \subseteq \mathcal{F}$ and $|S_i| \leq k$. Algorithm 3 shows the pseudo-code for feature relevance estimation.

4.6 Redundancy Estimation

The feature relevance estimation does not include the effect of redundancy. This means, two identical features are ranked the same based on its relevance scores. A major challenge lies in the detection of redundant features which do not have identical values as explained in Section 4.3. Hence, redundancy is not a binary decision. A pair of redundant features can only have a certain magnitude of information shared among them. Therefore, it is necessary to incorporate this specific information into the final score that exemplifies redundancy and relevance. The principle of redundancy estimation is similar to the relevance measurement. We use the same property of comparing marginal and conditional distributions as in Definition 4.1 to evaluate redundancy.

Definition. 4.3: Feature Redundancy

Given a set of features $R \subseteq \mathcal{F}$, a feature $f_j \mid (f_j \in \mathcal{F}$ *and* $f_j \notin R)$ is non-redundant w.r.t. R *iff*:

$$P\Big(p(f_j \mid R) = p(f_j)\Big) = 1.$$

Using Definition 4.3 we aim to compute the feature-feature redundancy and not the feature-target relevance. For this reason we do not include the target variable Y in our definition. Our feature redundancy estimation is a two step process.
Step 1: All features $f_j \in \mathcal{F}$ are ranked based on relevance $r(f_j)$ score.
Step 2: For an ordered set R_n that denotes a set of features until relevance rank n, we compute redundancy score of n^{th} ranked feature based on the redundancy it imposes on features with relevance rank 1 to $n-1$.
By following this methodology, if two redundant features have similar relevance scores, the second feature will obtain a higher redundancy score. This redundancy score is used to devalue the redundant contribution of that feature.

$$red(f_j, R) \equiv E_R\Big[div\Big(p(f_j \mid R) \mid\mid p(f_j)\Big)\Big] \tag{4.7}$$

If f_j is independent of R, the expected marginal and the conditional probability distributions will be the same. In other words, if f_j has non-redundant information w.r.t. the features $f \in R$, the deviation between the distributions in Equation 4.7 will be 0. We illustrate the steps with an example.

Example 4.2. Assume a feature space $\mathcal{F} = \{f_1, f_2, \ldots, f_5\}$ in which f_1 and f_3 are redundant features.

For the given feature space in Example 4.2, the features are sorted based on relevance scores following the step 1, i.e., $R_n = \{f_5, f_3, f_1, f_2, f_4\} \mid n = |\mathcal{F}|$. The highest relevant feature f_5 is not evaluated for redundancy as, it has no preceding ranked features to be redundant with. The redundancy that f_3 imposes on $R_1 = \{f_5\}$ is estimated by applying Equation 4.7. Therefore, we rank the features based on their relevance and use the top n-relevant features to compute the redundancy of f_{n+1}. The pseudo-code for this estimation is shown in Algorithm 4.

For estimation of feature relevance, we restricted the subspace size to k (c.f. Section 4.5). This avoids inaccurate conditional probability estimates. Algorithm 4 also involves estimation of conditional probabilities. For a large $|R_n|$, the conditional probability estimations using Equation 4.7 are not accurate. For example: for estimating the redundancy of the 100^{th} ranked feature, we need to estimate the conditional based on the 99 features ahead in the rank. Thus, for estimation of redundancy score of the n^{th} ranked feature, we sample subspaces $\forall S \subseteq R_{n-1}$.

Algorithm 4 Estimation of Redundancy

Input: $\mathcal{F}, Y$

1: R_n=Sort $\forall f_j \in \mathcal{F}$ based on $r(f_j)$ from Algorithm 3
2: **for** $n = 2 \rightarrow |\mathcal{F}|$ **do**
3: Compute $red(R_n \triangle R_{n-1}, R_{n-1})$ c.f. Equation 4.7 ▶$\triangle$ denotes symmetric difference
4: **end for**
5: **return** Calculate redundancy scores red $\forall f_j \in \mathcal{F}$

From R_{n-1}, various subspaces S of size k are sampled without replacement, i.e., $\binom{n-1}{k}$ number of subsets. The maximal imposed redundancy of the n^{th} ranked feature on the list of subspaces is the redundancy of the n^{th} feature. In Section 4.9 we provide an enhancement for redundancy approximation as a part of relevance estimation algorithm itself.

4.7 Unification of Relevance and Redundancy scores

Having estimated the relevance and redundancy of the features in Section 4.5 and 4.6, our final goal is to rank features based on a single score that combines both the properties.

Definition. 4.4: RaR score

Given the relevance $r(f_j)$ and redundancy score $red(f_j, R)$ of feature f_j, we define $RaR(f_j)$ score as,

$$RaR(f_j) = \left[\frac{2 \cdot r(f_j) \cdot (1 - red(f_j, R))}{r(f_j) + (1 - red(f_j, R))} \right].$$

$RaR(f_j)$ is the harmonic mean of relevance and redundancy scores. The harmonic mean in Definition 4.4 penalizes the relevance score with the information based on redundancy.

Example 4.3. Assume a feature space $\mathcal{F} = \{f_1, f_2, \ldots, f_5\}$ in which f_1 and f_3 are relevant and exhibit feature interactions. Additionally, f_4 and f_5 are features with redundant information.

In such a case, RaR ranks the feature based on multi-feature interactions and redundancy. Hence, RaR ensures that the non-redundant and the features with interactions, i.e., $\{f_1, f_3\}$ to be present ahead in the feature ranks.

4.8 Time Complexity

RaR consists of three major phases: subspace sampling for constraint generation (Lines 2- 7 of Algorithm 3), quadratic optimization (Line 8 of Algorithm 3) and redundancy estimation (Algorithm 4). In the following, we discuss the time complexity of each part and finally present the overall time complexity of our approach.

For each Monte Carlo iteration, we compute the subspace relevance based on the slicing method presented in [KMB12]. This requires to iterate the instances in the selected slice. In the worst case scenario, all instances are included in the slice with a time complexity of $\mathcal{O}(N)$. The selection of a slice is done for each dimension in subspace S_i. Since $|S_i| \leq k$, it leads to a complexity of $\mathcal{O}(N \cdot k)$ for calculating $rel(S_i)$ (Line 4 of Algorithm 3). The total time complexity for extracting M constraints takes $\mathcal{O}(M \cdot N \cdot k)$. The final step of estimating the relevance of each feature, requires to optimize Equation 4.6 subject to M constraints (Line 8).

A quadratic programming algorithm for a d-dimensional feature space has a time complexity $\mathcal{O}(\sqrt{d} \cdot ln\frac{1}{\epsilon})$ [Gon12]. The complexity considers that the optimizer converges to an ϵ-accurate solution. To compute the redundancy of a feature, we group subspaces of size k with all features ahead of it and compute the maximal redundancy using Equation 4.7. Thus redundancy takes a total time of $\mathcal{O}\left(d \cdot \frac{d-1}{k} \cdot N\right)$. Finally, ranking the features requires to sort the features based on their relevance and redundancy scores. This procedure requires $\mathcal{O}(d \cdot log(d))$. Considering the complexity of computing the harmonic mean of relevance and redundancy as constant, the total complexity of RaR is represented as,

$$\mathcal{O}\left(M \cdot N \cdot k + \frac{d^2}{k} \cdot N\right).$$

Instantiations for RaR

In Algorithm 3, a random subspace $S \subseteq \mathcal{F}$ is selected with maximum dimensionality k for each iteration. In order to estimate $rel(S)$, we compute the distribution of Y under some conditional slice of S. That is, we aim to obtain a slice of S which satisfies a specific set of conditions, i.e., $div(p(Y \mid S \in [c_1, \cdots, c_{|S|}]), p(Y))$. Defining explicit conditions is a tedious task. Hence, we use adaptive subspace slicing, more details can be found in [KMB12]. After calculating the subspace relevance, we extract an inequality and the set C is updated with this constraint. Finally, we obtain a set of M constraints and optimize the objective function of Equation 4.6 subject to these constraints.

RaR requires a divergence function to quantify the difference between distributions. As KLD is formulated for both continuous and discrete probability distribution, it is directly applicable for redundancy estimation (c.f. Definition 4.7) on mixed feature types. However, for continuous variables we use the KS-test because it does not make any assumptions on the sample distributions and relies only on the data samples [KMB12]. For this reason, we show Lemma 4.1 only for categorical case and not for continuous.

4.9 Algorithmic enhancement for redundancy estimation

In this we estimate the redundancy of a feature at position i based on the $i-1$ features ranked ahead of it (c.f. Algorithm 4). However, this can be computationally challenging for cases where there are thousands of features. We provide an algorithmic enhancement in Algorithm 5 to exploit the subsets sampled during the relevance estimation (c.f. Algorithm 3). The additional computations that we do in comparison to the relevance estimation are highlighted in the pseudo-code below. The enhancement includes sampling of an feature f in addition to the subset S (c.f. Line 5) and computing its redundancy with respect to the sampled subset (c.f. Line 6).

We explain the significance of our algorithm with a simple example. For a feature space $\mathcal{F} = \{f_1, \cdots, f_5\}$ and target Y, where f_1 and f_3 are redundant to

Algorithm 5 Enhancement for redundancy estimation

Input: $\mathcal{F}, Y, M, k$

$C = \emptyset$
for $i = 1 \rightarrow M$ **do**
    Sample $\{S_i \mid S_i \subseteq \mathcal{F} \wedge |S_i| \leq k\}$
    Compute $rel(S_i)$ using Definition 4.1
    Sample a feature $f \in \mathcal{F} \mid f \notin S_i$ ▶In addition to Algorithm 3
    Compute $red(f, S_i)$ (c.f. Equation 4.7) ▶In addition to Algorithm 3
    Construct constraint (cf. Definition 4.2)
    Add constraint to set C
end for
Optimize objective function Equation 4.6 subject to C
return $r(f) \mid \forall f \in \mathcal{F}$

i	S_i	f	$red(f, S_i)$
1	$\{f_2, f_3\}$	f_1	High
2	$\{f_2, f_3\}$	f_5	Low
3	$\{f_1, f_2, f_4, f_5\}$	f_3	High
4	$\{f_3, f_4\}$	f_2	Low
5	$\{f_2, f_3, f_4\}$	f_1	High
6	$\{f_2, f_4\}$	f_5	Low
7	$\{f_2, f_4, f_5\}$	f_3	Low
8	$\{f_1, f_4\}$	f_3	High

Table 4.3: Illustrative example of feature constraints for 3 Monte Carlo iterations

each other. Using 8 different Monte-Carlo iterations shown in Table 4.3, let us assume that the features are ranked as f_5, f_2, f_3, f_1, f_4, based on the relevancies. The redundancy of feature f_3 is estimated with respect to the features ranked ahead of it, i.e., $T = \{f_5, f_2\}$. The redundancy of f_3 with respect to other features, viz., f_1, f_4, are irrelevant at this point. To estimate the maximal imposed redundancy of f_3 w.r.t. T, we identify subsets S_i that have intersecting elements with T. We define such subset samples as admissible samples.

Definition. 4.5: Admissible subsets

For estimating redundancy of a feature f with respect to a set of features $T \subseteq \mathcal{F}$, the sample $red(f, S_i)$ is defined to be admissible iff,

$$S_i \cap T \neq \emptyset$$

Our Definition 4.5 states that a redundancy sample $red(f, S_i)$ will provide us insights about $red(f, T)$ only if there are intersecting elements between T and S_i. For example, to estimate the redundancy of f_3 with respect to $T = \{f_5, f_2\}$, the $i = 8$ does not provide any useful insights. This is because, there are no common elements between S_8 and T. On contrary, with the presence of common elements, $i = 3, 7$ are admissible subsets. However, the admissible subsets can largely over-estimate because we are still not aware if the redundancy score is influenced by the common elements of $S_i \cap T$ or $S_i \setminus T$. We can use our subset samples in Table 4.3 to identify this. That is, an admissible sample is said to be justified and not over-estimating if it satisfies our Definition 4.6.

Definition. 4.6: Justified subsets

An admissible redundancy sample $red(f, S_i)$ is justified with respect to a subset of features T iff,

$$\nexists S_j \subseteq \mathcal{F} : \forall x[x \in (S_i \cap T) \rightarrow x \in S_j] \text{ and}$$

$$red(f, S_j) < red(f, S_i)$$

Our admissible subset S_3 is not-justified as we have S_7 that contains all intersecting elements $S_3 \cap T$ and has a lower redundancy score $red(f_3, S_7) < red(f_3, S_3)$. On contrary our admissible subset S_7 is justified as we have S_3 that contains all intersecting elements $S_7 \cap T$ but does not have has a lower redundancy score. Hence, the redundancy of f_3 is $red(f_3, S_7)$ because the sample 3 denotes its redundancy with respect to f_1, which is irrelevant for estimating the redundancy with respect to $T = \{f_5, f_2\}$. In the event of multiple admissible and justified subsets, the redundancy of a feature f is the maximum redundancy of the justified subsets.

$$\max_{S_i | justified} red(f, S_i)$$

Though the aforementioned enhancement enriches the algorithms runtime, it can have undesirable properties. That is, in the event where no redundancy sample is admissible, we have no knowledge about the feature's redundancy. In such cases, it is recommended to follow Algorithm 4 for better estimation.

4.10 Experimental Evaluation

Experimental Setup

In this section we compare the run times and quality of our approach against several existing techniques as competitors. We consider techniques from different paradigms, i.e., filters, wrappers, embedded and hybrid techniques for mixed data as competitors. As wrappers, we test Sequential Forward Selection (SFS) [TPKC10] with K-Nearest Neighbors (KNN) [KGG85], capable of handling redundant features. As hybrid technique, we consider the heuristic of Doquire [DV11]. The scheme requires a correlation measure and a classifier, hence we employ mRmR [PLD05] and KNN with the heuristic of Doquire [DV11]. As filter approach, we test Maximal Information Criterion (MIC) [LMD$^+$12], mRmR [DP05, PLD05], ReliefF [RŠK03] and Correlation Filter Selection (CFS) [Hal00]. Finally, we test the embedded scheme of decision trees (C4.5 [Qui14]). The results of our experiments on other classifiers are also made available. Additionally, we employ Gurobi [LLC15] optimizer for the optimization of relevancies in RaR. We evaluate and compare our approach with the above mentioned competitors on synthetic and real world datasets.

Synthetic datasets were generated with varying database sizes and dimensionality. We employ the synthetic data generation program of NIPS [NIP01] to generate continuous feature sets with normal distribution in any proportion of relevant (with multi-feature interactions) and noisy features. For a generated continuous feature f and v number of states, we discretized f to form a categorical feature of v unique values. In our experiments, we generated mixed datasets with equal number of categorical and continuous features. As a measure of feature ranking quality, we use Cumulative Gain (CG) from Information Retrieval [BYRN99].

For evaluation of our feature ranking framework, we also use 6 public datasets from the UCI repository with different dimensionalities and database sizes (c.f.

Dataset	# features	# samples
NIPS [CL06]	500	2000
Ionosphere [SWHB89]	24	351
Musk2 [DJLLP94]	166	6598
Isolet [FC90]	500	2000
Semeion [Bus98]	179	14537
Advertisement [FM03]	1558	3279

Table 4.4: Properties of datasets used for experimental comparison of RaR and other competitor approaches

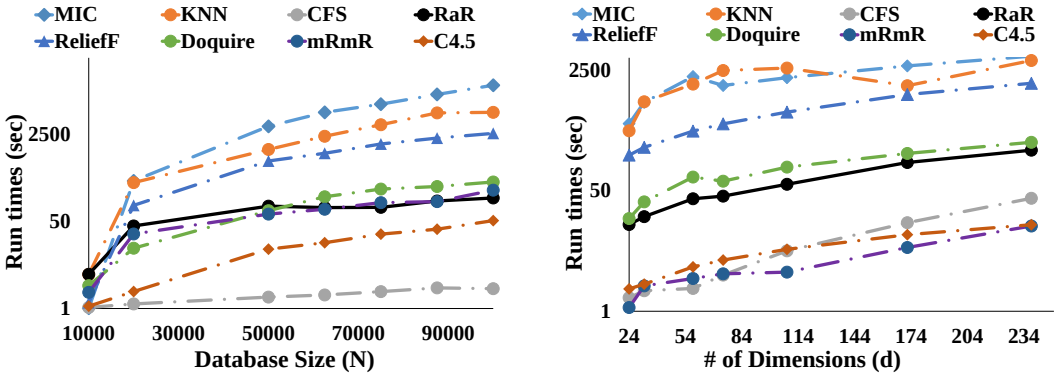
Table 4.4). The datasets contain both continuous and categorical features. Experiments that had run times more that one day are denoted as ** in Table 4.5 and 4.6.

4.10.1 Synthetic Data sets

We perform scalability analysis by evaluating the run times with increasing dimensionality and database size. Figure 4.2 shows the efficiency of RaR with increasing database size and dimensionality. In general, methods that do not evaluate for feature interactions, i.e., C4.5, mRmR and CFS, have lower run times than RaR. By evaluating these interactions, RaR has better feature ranking quality (c.f. Figure 4.3). In comparison to ReliefF, which ranks features based on multi-feature interactions, RaR has lower run times and better feature ranking quality.

Parameter Analysis

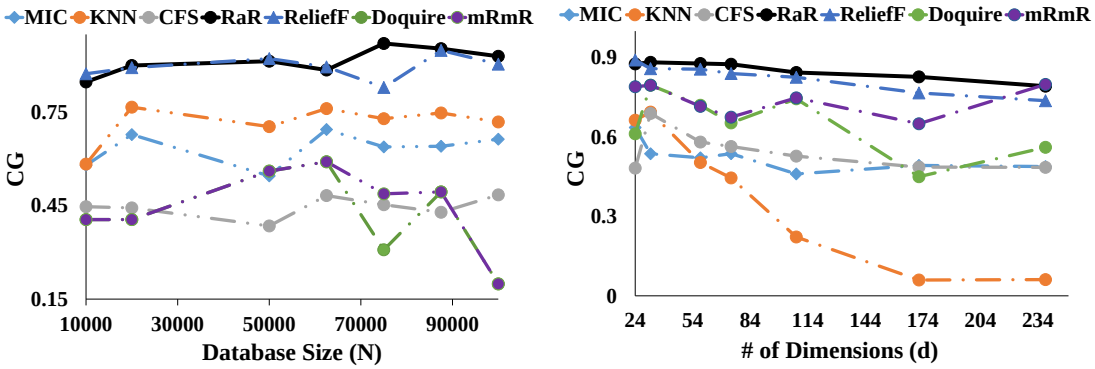
The k parameter of RaR decides the maximum size of the subset drawn for every iteration $i \mid i = 1, \cdots, M$. From our experiments (c.f. Figure 4.4a) on synthetic data, we observe that the CG decreases with increasing k. The size of the conditional slices is determined by the α parameter [KMB12]. For a dataset of $N = 1000$ and $\mid \mathcal{F} \mid = 100$, setting $\alpha = 0.1$ and a large value of k ($k = 50$) leads to a conditional slice of size $\alpha^{\frac{1}{k}} \cdot N$ [KMB12]. Hence, the conditional slice has approximately 95% of all the instances. This leads to a very similar conditional and marginal distributions and distorted feature ranking. In Figure 4.4b, we vary



(a) Fixed dimensionality (50) and increasing database size

(b) Fixed database size (20000) and increasing dimensionality

Figure 4.2: Run time Evaluation: Run times of RaR vs. competitor approaches

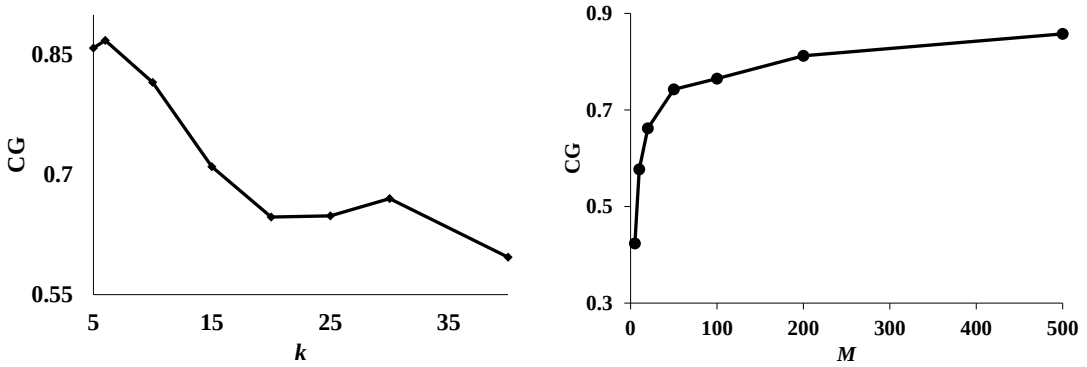


(a) Fixed dimensionality (50) and increasing database size

(b) Fixed database size (20000) and increasing dimensionality

Figure 4.3: Quality Evaluation: CG of RaR vs. Competitor techniques

M and evaluate its influence on feature ranking. The experiment shows that the ranking quality is stable for a large range of M. Thus, we recommend to restrict k to small values and increase M for better accuracy. Choosing large M affects run times of selection process. However, the task of sampling and building constraints can be distributed over multiple processor threads. Figure 4.5 shows the efficiency gained by distributed computations of RaR. Speedup denotes the number of folds of decrease in run times (w.r.t. single thread) on distributing the Monte Carlo iterations to multiple processor threads.



(a) Influence of k on feature ranking (M=100)

(b) Influence of M on feature ranking (k=5)

Figure 4.4: Parameter Study, on synthetic dataset of 50 features and 20000 instances

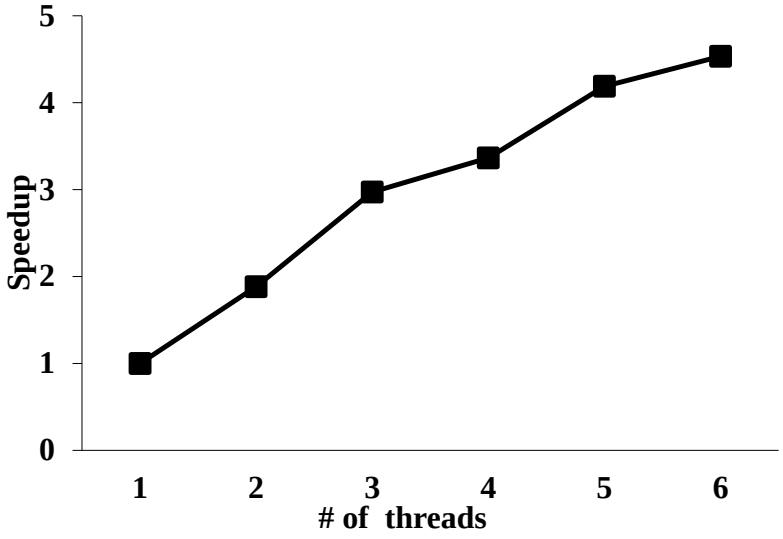


Figure 4.5: Speedup of RaR

Robustness w.r.t. erroneous labels

In several application scenarios, the target labels Y are assigned by domain experts. This manual process is prone to errors. With such datasets, it is necessary to ensure that the feature ranking is robust to erroneous target labels. To test this, we manually induced label errors in the synthetic datasets. The hybrid approach from Doquire [DV11] was able to perform well on a few cases (c.f. Figure 4.6). However, as a filter approach, RaR defines the feature relevance score based on constraints defined by multiple subsets. Thus, RaR is more robust to label errors.

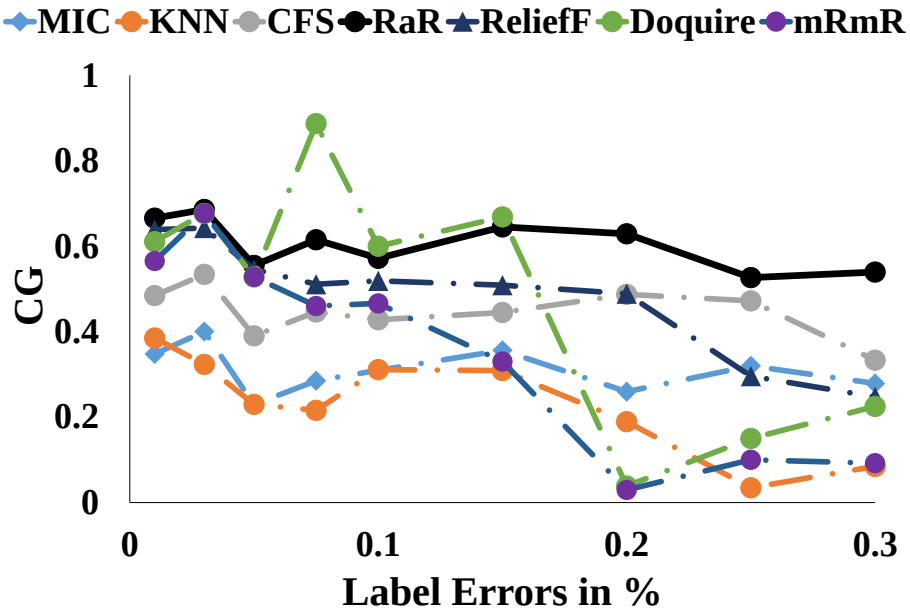


Figure 4.6: Robustness of feature ranking

Selection	NIPS	Ionosphere	Musk2	Isolet	Semeion	Advertisement
Full-dimension	0.57	0.70	0.8	0.58	0.1	0.73
C4.5	0.58	0.87	0.9	0.63	0.79	0.9
MIC	0.78	0.83	0.86	0.78	0.8	0.91
SFS(KNN)	0.84	0.85	0.91	**	**	**
CFS	0.82	0.81	0.86	0.82	0.9	0.91
ReliefF	0.87	0.79	0.84	0.82	0.87	0.87
mRmR	0.55	0.89	0.9	0.57	0.9	0.9
Doquire	0.56	0.88	0.9	0.56	0.93	0.9
RaR	0.88±0.006	0.88±0.00	0.91±0.008	0.87±0.002	0.92±0.005	0.92±0.005

Table 4.5: Average f-score of 3 fold cross-validation using KNN (K=20) classifier

Selection	NIPS	Ionosphere	Musk2	Isolet	Semeion	Advertisement
C4.5	1.2	0.5	3.1	3.8	0.21	15.58
MIC	37.7	0.47	40.79	37.25	81.2	49.35
SFS(KNN)	105741.3	6.9	14132.9	**	**	**
CFS	36.7	1.8	8.3	37.5	2.51	417.9
ReliefF	29.3	0.18	98.08	32.7	5.46	95.07
mRmR	42.3	0.5	4.5	59.27	6.1	78.81
Doquire	44.6	4.25	9.19	62.15	9.8	131.42
RaR	10.35	2.05	5.3	7.9	4.37	50.26

Table 4.6: Feature ranking run times in *sec* of RaR vs. competitor approaches

4.10.2 Real world Data sets

Table 4.5 shows the results w.r.t. the prediction quality of each feature selection technique. Overall, we observe that application of feature selection improves the quality of prediction. By evaluating the feature interactions in the dataset, RaR has the best accuracy in comparison to the competitor approaches. Especially, the existing feature selection techniques do not show improvement of f-score in the case of NIPS challenge dataset. NIPS dataset contains multi-feature interactions, noisy and large number of redundant features. As the competitor approaches do not evaluate feature interactions, they assign lower scores to such interacting features.

Table 4.6 shows that our approach is several times more efficient in comparison to the competitor filter and wrapper methods. Embedded approach C4.5 has lower run times in comparison to RaR. However, C4.5 is unable to identify feature interactions and has lower prediction quality (c.f. Table 4.5). Similar to our experiments on synthetic datasets (c.f. Figure 4.2 and 4.3), we observe that methods that have lower run times than RaR have lower f-scores as they no not evaluate feature interactions. For dataset with few features (Ionosphere data), simple bivariate correlation measures (MIC and CFS) was a better choice w.r.t. run times.

Evaluation of the ranking

To evaluate the quality of feature ranking, i.e., to experimentally show that the top ranked features of RaR are maximally relevant and non-redundant, we follow a 2 step evaluation process on real world datasets. First, we rank the features using each approach. Then, we iteratively add the features ranked by each technique to a classifier (KNN [KGG85]) in the order (best to worst) of their ranks. As shown in Figure 4.7, after including each feature, the average f-score of 3 fold cross-validation is calculated. As the top ranked features of RaR are non-redundant, we observe the best quality with the least number of features. However, other approaches do not take into account the effect of redundancy. For example, ReliefF has very similar prediction quality (c.f. Table 4.5) to RaR. By ranking the non-redundant features ahead, RaR achieves better f-score with fewer features (c.f. Figure 4.7), i.e., RaR obtains an f-score of 0.87 with 14 features and ReliefF obtains an f-score of 0.82 with 20 features. We performed the experiment on the public datasets and

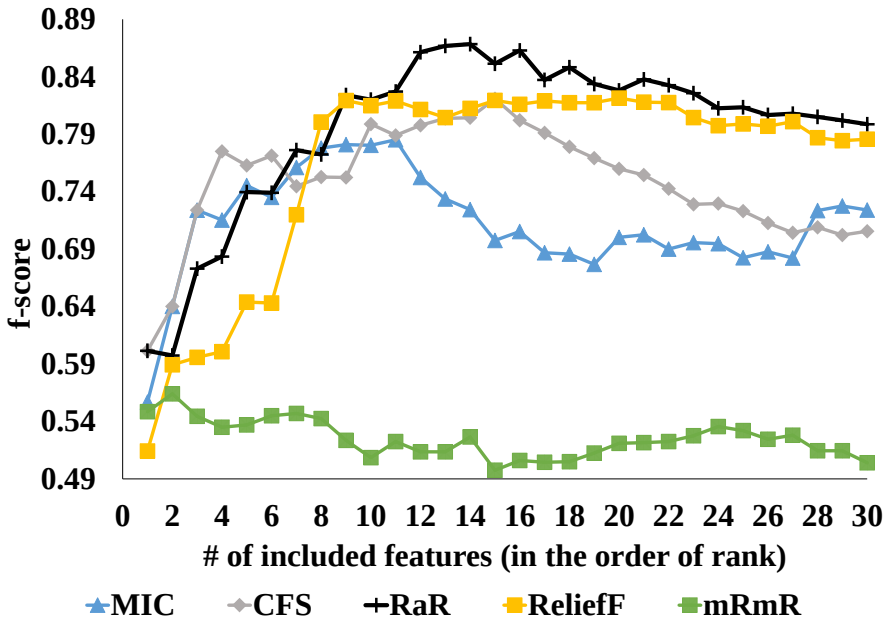


Figure 4.7: f-Scores of top 30 features on Isolet dataset

Selection	NIPS	Ionosphere	Musk2	Isolet	Semeion	Advertisement
MIC	11	2	163	11	82	14
SFS(KNN)	5	2	135	**	**	**
CFS	15	2	155	15	119	7
ReliefF	20	4	136	20	173	54
mRmR	5	5	117	2	151	13
Doquire	2	4	117	2	156	15
RaR	12	2	16	11	17	9

Table 4.7: Number of features required to obtain the quality in Table 4.5

we show the number of features (c.f. Table 4.7) at which the maximum f-score (c.f. Table 4.5) was observed. Table 4.7 shows the number of top ranked features required to obtain the quality in Table 4.5, and RaR achieves the best f-score with fewer features.

4.11 Summary

In this chapter, we presented a novel feature ranking framework that includes the effect of higher-order interactions for relevance estimation and redundancy in

datasets with continuous and categorical feature types. RaR randomly samples several feature subsets and each of their relevance to the target prediction is computed. Using this information, the relevance of each feature is estimated by evaluating the role of a feature in multiple subsets. Hence, the interactions of a feature in several feature subsets are assessed to derive an estimate of a feature's relevance. The framework is applicable for datasets with any feature type as the divergence between two discrete distributions or probability density are comparable.

The results of various state-of-the-art algorithms on the synthetic and real world datasets show that our feature ranking method is suitable for high-dimensional datasets exhibiting complex feature interactions. By ranking the non-redundant features ahead, RaR achieves better prediction quality with fewer features. Hence, we aim to retain the idea of multi-view analysis using filter-based paradigm and extend it for time-series data types. Estimating the conditional and marginal distributions in RaR require sorting the features. That is, on applying RaR on time series data, the inherent ordering of the series is distorted. It is still an open challenge:

- To extract features by preserving the multivariate interactions in time series dataset.
- To efficiently evaluate the relevance and redundancy of the extracted features.

In the forthcoming chapter, we introduce a novel feature extraction framework for multivariate time series and address these problems.

Chapter 5

Multivariate Relevance and Redundancy Scoring in Time Series

5.1 Motivation

Time series classification is predominant in several application domains such as health, astrophysics and economics [FJ14, GGK+13, YK09]. In particular, for automotive applications, the time series data is transmitted from the vehicle to a remote location. In such cases, the transmission costs are large for lengthy and high-dimensional time series signals. Feature-based approaches handle this problem by transforming the lengthy time series into compact feature sets. The transformation of time series can be done based on several properties (c.f. Section 2.3.1 in Chapter 2), e.g., frequency and amplitude properties of the time series are captured using a Fast Fourier Transform (FFT).

Fundamentals of ordinal patterns

Several time series applications need to capture the structural changes instead of the exact values at each instant of time [SGK12, YK09]. A transformation based on the ordinality of the time series effectively captures these structural changes in a dynamic system [BP02, GGK+13, SGK12]. Let us consider a simple univariate time series X that represents the behavior of a smooth driver (c.f. Figure 5.1). The series X is of length $l = 7$, where $X[t]$ denotes the value of X at time t. To evaluate the ordinality at each time step t, a window of $\mathscr{d} - 1$ (where $\mathscr{d} \geq 2$) preceding values in the time series are used [BP02]. For $\mathscr{d} = 3$, the ordinality at $t = 3$ * is $X(t) > X(t-1) > X(t-2)$, which is represented as 012. As shown in Figure 5.1, for a fixed $\mathscr{d}$, there are at most $\mathscr{d}!$ unique ordinalities that exist in a time series and we denote each of them with an unique symbol. Hence, the ordinalities of X at $t = 3, ..., 7$ are denoted as (u, u, x, w, u). Given $\mathscr{d}!$ ordinalities, an ordinal pattern is a subset of ordinalities, e.g., $\{u, x\}$ is an univariate ordinal pattern. Thus, there are at most $2^{\mathscr{d}!}$ patterns present in a univariate time series.

Similarly, we compute the ordinalities for a rash driver X_r (c.f. Figure 5.2) as, (z, z, z, w, x). On comparing the ordinal representation of the two time series (i.e., X and X_r), we see that the smooth driver has an increasing trend (u) in the time series more often ($\frac{3}{5}$ times) and a rash driver exhibits a declining trend (z) more

*as $t = 1$ and 2 have less than $\mathscr{d} - 1$ preceding values

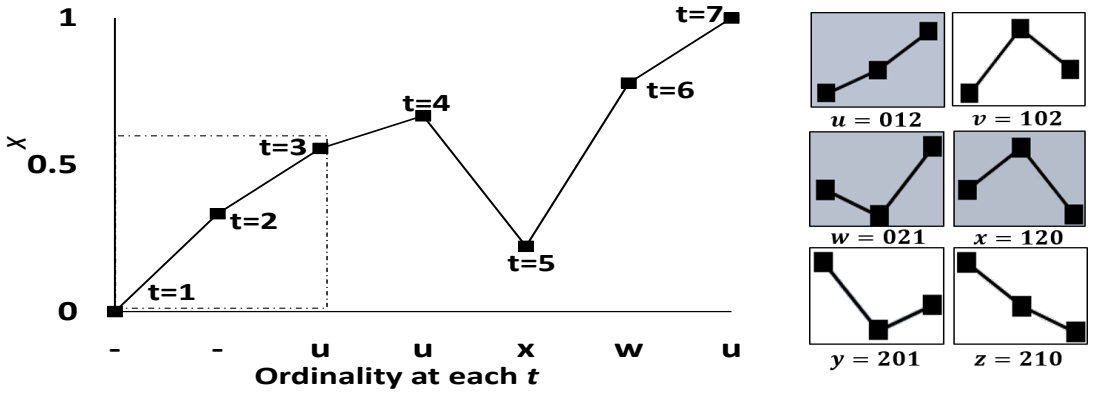


Figure 5.1: Example of univariate ordinality and the all ordinalities of $\mathscr{d} = 3$ for a smooth driver

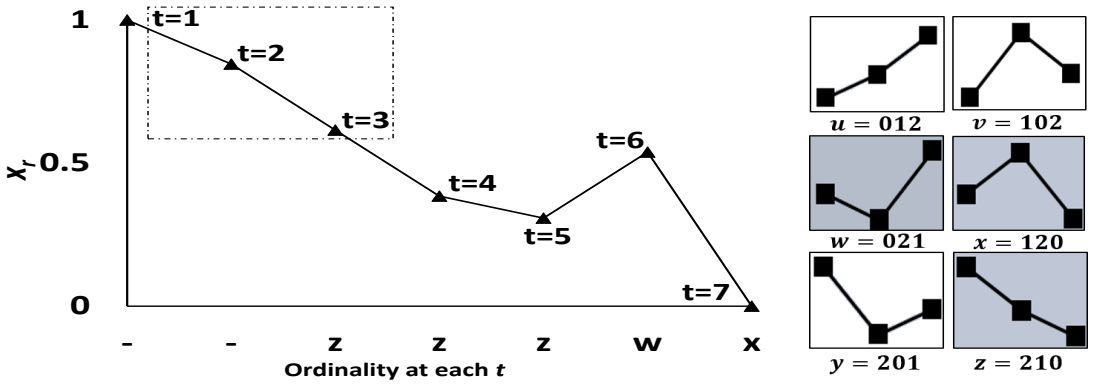


Figure 5.2: Example of univariate ordinality and the all ordinalities of $\mathscr{d} = 3$ for a rash driver

often. Secondly, Figure 5.1 and 5.2 also show that the ordinal patterns w and x occur for both time series, but this information is not sufficient to discriminate between the driver types, i.e., irrelevant ordinalities. Thus, encoding the occurrence of relevant ordinalities as a feature can be beneficial for time series classification. The above example is motivated using our automotive application domain. However, for any non-stochastic system, these hidden structural and dynamic changes are captured by a transformation based on ordinality.

In a multivariate time series classification task, there can be co-occurrence of patterns between multiple dimensions that are more relevant for the class prediction than individual patterns. For example (c.f. Figure 5.3), in automotive applications, an increasing pattern (u) of engine torque and declining (z) tem-

perature combined together indicates a specific component failure. However, the increasing torque in combination with other ordinalities (e.g., v_{temp}) is not relevant for classification. In such cases, for m dimensions, the number of possible multivariate pattern combinations scales up to $2^{(\mathscr{A}!\cdot m)}$. Following the traditional feature-based approach [FJ14] of transforming all pattern combinations into numeric features and performing feature selection to identify the relevant patterns is computationally inefficient.

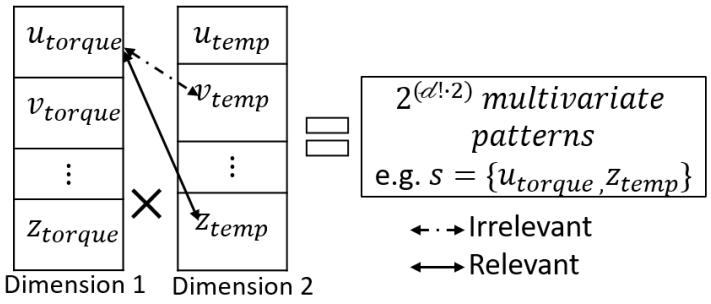


Figure 5.3: Example of multivariate pattern combination

Thus, the first challenge is to efficiently extract these multivariate patterns and estimate their relevance simultaneously. However, none of the existing works on ordinal patterns [BP02, GGK+13, SGK12] consider the influence of ordinalities in multivariate time series datasets.

Additionally, multiple patterns can have similar information (redundant) for the class prediction. For example, for a declining engine torque pattern, the engine speed also exhibits a declining pattern. This implies that both patterns provide redundant information for classification. In such cases, it is necessary to ensure that the extracted patterns have complementary information to each other. Thus, the second challenge lies in estimating the novelty of the features extracted using ordinal patterns. Nevertheless, existing feature-based transformation techniques [FJ14, Mör03, NAM01, WWW07] do not focus on considering both challenges: relevance w.r.t. classes and redundancy of the extracted features. In this work, we introduce **Ord**inal feature **ex**traction (*ordex*)*, a feature-based approach for multivariate time series classification using the property of ordinality in the time series.

*Adapted by permission from Springer: Selection of Relevant and Non-Redundant Multivariate Ordinal Patterns for Time Series Classification in the proceedings of the International Conference on Discovery Science (DS), 2018 [SPISM18]

After conversion of the raw multivariate time series dataset into its ordinal representation, we define a method to extract multivariate ordinal patterns. To estimate the relevance of these patterns, *ordex* introduces a measure. This measure estimates the recurrence of an extracted pattern in a given class and its uniqueness w.r.t. other classes. The relevance estimation is followed by the redundancy calculation. Given a set of relevant patterns, *ordex* scores the non-redundancy of each pattern based on its correlation with other relevant patterns. Finally, both scores are combined such that the unified score exemplifies relevance and non-redundancy. Experiments on real world and synthetic datasets show that our approach is beneficial for several application domains.

5.2 Comparison to Related Work

Paradigm	Approach	Multivariate	Relevance	Redundancy
Ordinal patterns	[BP02] and [SGK12]	✗	✗	✗
	[GGK+13]	✗	✓	✗
Sequence-based	Shapelet[YK09]	✗	✓	✗
	MCMR[WJW+15]	✗	✓	✓
	LSTM [HS97]	✓	✓	n.a.
Feature-based	[NAM01], DTW[Kat16], SAX [LKL12], [WSH06] and HCTSA [FJ14]	✗	✗	✗
	Ordex	✓	✓	✓

Table 5.1: Comparison of *ordex* with other relevant literatures from different time series classification paradigms

We distinguish our work on feature extraction in this chapter from the others discussed in Section 2.4 based on three characteristics: (1) capture the multivariate interactions in the time series dataset, (2) evaluate relevance and (3) redundancy of extracted features simultaneously without the need of additional post-processing such as feature selection. In Table 5.1, we group the time series literatures into two paradigms, i.e., feature-based and sequence-based. In this chapter we contribute a feature-based framework for extracting relevant and non-redundant features from

the time series based on the property of ordinality. In contrast to the existing works on ordinality and feature-based extraction techniques, we propose a multivariate feature extraction scheme in this work. By evaluating the relevance and redundancy of the extracted features *ordex* provides a smaller set of predictive features from the time series.

5.3 Problem Definition

A multivariate ordinal pattern s is a set of ordinalities from multiple dimensions, e.g., in Figure 5.3, $s = \{u_{torque}, z_{temp}\}$. In a multivariate time series dataset, a large number of pattern combinations exist and several of them are irrelevant for classification and redundant to each other. We denote $error : s \mapsto \mathbb{R}$ as the error function of the classifier trained using an ordial pattern s. The classification error using a relevant pattern s_1 is lower in comparison to that of an irrelevant pattern s_2, i.e.,

$$error(s_1) < error(s_2).$$

On the other hand, using redundant patterns for classification does not improve the prediction accuracy. That is, for a set of patterns $\mathcal{S}$, where $s_i \in \mathcal{S}$ has redundant information to other elements in $\mathcal{S}$,

$$error(\mathcal{S}) \approxeq error(\mathcal{S} \setminus s_i).$$

Irrelevant and redundant features lead to large feature space and lower prediction quality [SBS+17]. Hence, the contributions of this work are two-fold:
(1) Including and defining the multivariate nature of ordinal patterns for time series classification.
(2) A novel score for evaluating the relevance and redundancy of ordinal patterns without training a classifier.
From a pool of large number of ordinal patterns, we aim to select a set of o patterns $\mathcal{S} = \{s_1, \cdots, s_o\}$ that are relevant for classification and are non-redundant w.r.t. other elements in the set. Hence, we maximize the sum of the individual relevancies and minimize the correlation between the ordinal patterns. This requires a scoring function that can efficiently estimate the ability of a multivariate pattern

to discriminate between different classes, i.e.,

$$rel : s \in \mathcal{S} \mapsto \mathbb{R}.$$

Secondly, a redundancy scoring function to ensure that the elements in $\mathcal{S}$ have complementary information to contribute for the classifier, i.e.,

$$red : (s \in \mathcal{S}, \mathcal{S} \setminus s) \mapsto \mathbb{R}.$$

Notations: As we aim to extract and evaluate ordinal patterns from multivariate time series, we begin with the conversion of raw time series into its ordinal domain. In the work of [BP02], ordinality of degree $\mathscr{d} \geq 2 \mid \mathscr{d} \in \mathbb{N}$ at each instant of time $t \mid (\mathscr{d} - 1) < t \leq l$ for a univariate times series $X = (x_1, \cdots, x_l)$ of length l is defined as,

$$\mathbb{O}_{\mathscr{d}}(X, t) = (rank(X[t]), rank(X[t-1]), ..., rank(X[t-(\mathscr{d}-1)])), \tag{5.1}$$

where $rank(X[t])$ is the position of $X[t]$ after sorting the values of $(X[t], ..., X[t-(\mathscr{d}-1)])$. For example, in Figure 5.1,

$$\mathbb{O}_{\mathscr{d}=3}(X, t=4) = X(t) > X(t-1) > X(t-(3-1)) = 012.$$

Thus, the ordinal representation of a univariate time series X is a new series $ord_{\mathscr{d}}(X) = \mathbb{O}_{\mathscr{d}}(X, t), \cdots, \mathbb{O}_{\mathscr{d}}(X, l)$, where the ordinality $\mathbb{O}_{\mathscr{d}}(X, t)$ at each instant of time t is assigned as a symbol. The resulting series can have a maximum of $\mathscr{d}!$ distinct symbols and a length of $l' = l - (\mathscr{d} - 1)$. For example, in Figure 5.1, $ord_3(X) = (u, u, x, w, u)$ and $l' = 7 - (3 - 1) = 5$.

A m-dimensional time series sample $T^j = \langle X_{\,1}, \cdots, X_{\,m} \rangle$ is a m-tuple of univariate time series. Finally, a multivariate time series dataset $D = \{T^1, \cdots, T^n\}$ consists of n such multivariate time series samples. As a supervised approach, each sample $T^j \in D$ is assigned a class from a set of possible classes $C = \{c_1, \cdots, c_k\}$. The i^{th} dimension in the j^{th} sample of a dataset is denoted as T^j_i. The ordinal representation of a multivariate time series dataset D is a collection of the ordinal representations of all univariate time series, i.e., $ord_{\mathscr{d}}(D) = \{\langle ord_{\mathscr{d}}(T^j_1), ..., ord_{\mathscr{d}}(T^j_m) \rangle \mid j = 1, \cdots, n\}$. For the ease of notation, we use a fixed length l for all time series, but this is not a formal requirement.

5.4 Extraction of Multivariate Ordinal Patterns

Ordex is a heuristic approximation algorithm that includes evaluation of relevance and redundancy of ordinal patterns. As shown in Figure 5.4, a m-dimensional time series dataset D is converted to its ordinal representation of defined degree $\mathscr{d}$, i.e., $ord_{\mathscr{d}}(D)$ (c.f. Section 5.3). From the ordinal search space, *ordex* aims to extract multivariate ordinal patterns. Hence, we begin with the introduction of multivariate ordinal patterns. This section is followed by our relevance and non-redundancy scoring function for ordinal patterns. Finally, we elaborate on the algorithmic component of our approach.

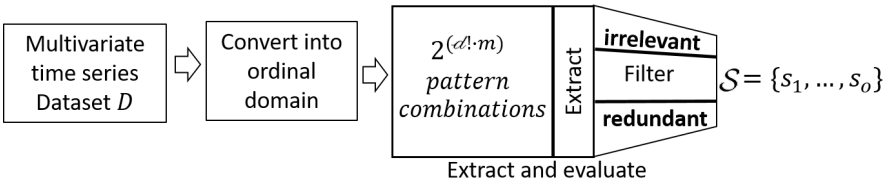


Figure 5.4: Workflow of *ordex*

As shown in Figure 5.3, a multivariate ordinal pattern is a subset of ordinalities from multiple dimensions. We introduce multivariate ordinal pattern set with our formal definition.

Definition. 5.1: Multivariate Ordinal Pattern set

Let $\mathcal{I} = \{1, \cdots, m\}$ be the set of dimensions and $\Omega_i = \bigcup_{1 \leq j \leq n} ord_d(T_i^j) \mid i \in \mathcal{I}$ is a set of ordinalities in the i^{th} dimension of all samples in D. Given the search space $\Omega = \{\Omega_i \mid \forall i \in \mathcal{I}\}$ and a subset of $m' \leq m$ dimensions, i.e., $\mathcal{I}' \subseteq \mathcal{I} \mid |\mathcal{I}'| = m'$, we define a multivariate ordinal pattern set as,

$$s = \{\Pi_i \subseteq \Omega_i \mid \forall i \in \mathcal{I}'\}.$$

As discussed in the Section 5.3, evaluating every possible pattern set is computationally inefficient. In this work, we handle this challenge by using the Monte-Carlo approach [KMB12] where a random multivariate pattern set is extracted for each iteration.

Example 5.1. Assume a time series dataset $D = \{T^1, T^2\}$ with three dimensions, (i.e., $\mathcal{I} = \{1, 2, 3\}$) and two samples (i.e., $n = 2$) of length $l = 8$.

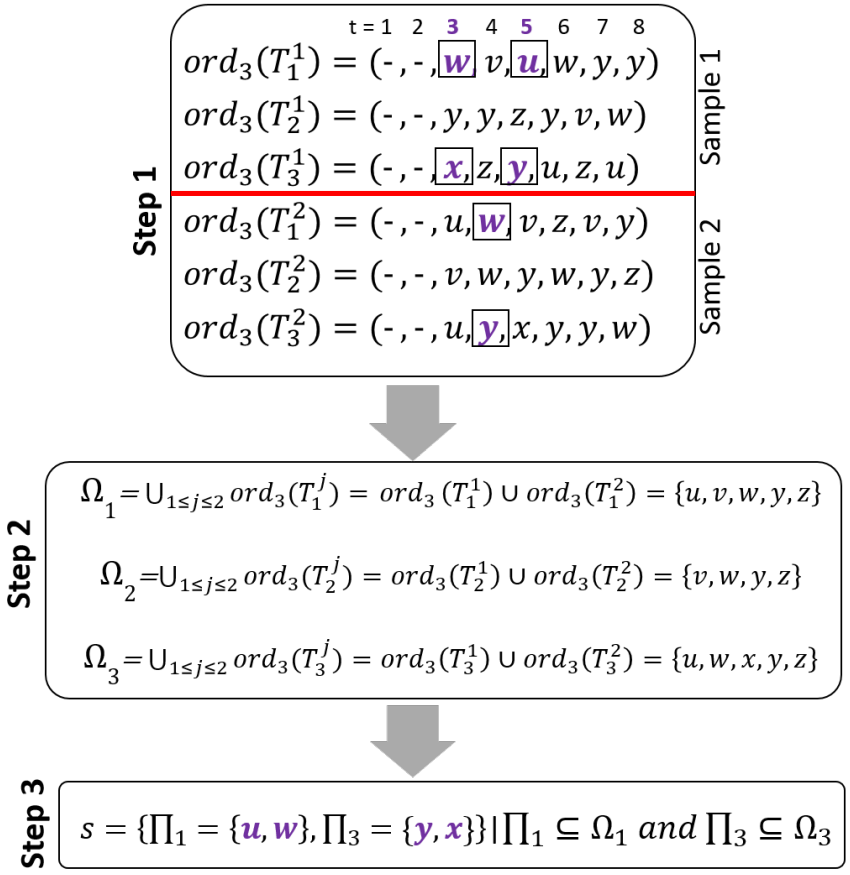


Figure 5.5: Illustration of multivariate ordinal pattern set for Example 5.1, where $\mathscr{d} = 3$

Using Figure 5.5, we show one possible multivariate ordinal pattern extracted from D by applying Definition 5.1. As the first step, the time series data is converted into its ordinal representation of $\mathscr{d} = 3$ by assigning its ordinality at each instant of time (c.f. Equation 5.1). For a set of ordinalities Ω_i in the i^{th}-dimension of all time series samples, e.g., $\Omega_1 = \bigcup_{1 \leq j \leq 2} ord_3(T_1^j)$, a multivariate ordinal pattern of size $m' = 2$ is a subset of ordinalities from m' dimensions. In our example in Figure 5.5, we select a random subset of dimensions $\mathcal{I}' = \{1, 3\}$. From each selected dimension, a subset of ordinalities are drawn to form a multivariate ordinal pattern set, i.e., $s = \{\Pi_1 \subseteq \Omega_1, \Pi_3 \subseteq \Omega_3\}$. In Figure 5.5 we show one possible multivariate ordinal pattern set s, where ordinalities u and w are drawn from Ω_1. Similarly, ordinalities y and x are drawn from Ω_3.

In order to score the relevance of the extracted multivariate ordinal pattern set for classification, we transform the multivariate symbolic representation of ordinalities into a numeric feature. As our approach uses the ordinal representation of

time series and not the actual values, it is not possible to perform transformation based on standard operations such as mean or median. Following the literature of probabilistic sequential mining [XKWMN07], we perform a transformation based on the occurrences of a pattern set. For the extracted s, we compute its probability in each time series sample $j \mid j = 1, ..., n$ based on our definition below.

Definition. 5.2: Transformation function

Let $T = (T[1], \cdots, T[l])$ be a m-dimensional time series sample of length l, i.e., $T[t] \in \mathbb{R}^m$, and $\mathcal{I}'$ is a set of dimensions from which a multivariate ordinal pattern set s is extracted. The pattern s occurs in T at time t iff

$$ord_{\mathscr{d}}(T_i[t]) \in \Pi_i, \forall i \in \mathcal{I}'.$$

The transformation function assigns the probability of s in a time series sample, i.e., $P : (s, T) \mapsto \mathbb{R}$ and we define the transformation function as,

$$P(s, T) = \frac{|\{t \mid s \text{ occurs } \ in\ T \text{ at time } t\}|}{l - (\mathscr{d} - 1)}.$$

Hence, for a time series dataset with n-samples, the defined transformation function generates a n-dimensional numeric feature vector $f = (P(s, T^1), \cdots, P(s, T^n))$.

Example 5.2. Assume we apply our transformation function (c.f. Definition 5.2) to transform the multivariate ordinal pattern set s in Figure 5.5 into a numeric feature.

The Definition 5.2 transforms a multivariate pattern into a numeric feature by evaluating the co-occurrence of ordinalities from multiple dimensions. In Figure 5.5, s occurs at $t = 3, 5$ in T^1, i.e., $ord_3(T^1_1, 3) = w \in \Pi_1, ord_3(T^1_3, 3) = x \in \Pi_3$ and $ord_3(T^1_1, 5) = u \in \Pi_1, ord_3(T^1_3, 5) = y \in \Pi_3$. Thus, the occurrence of s in T^1 is $P(s, T^1) = \frac{2}{6} = 0.33$. The pattern s occurs in T^2 once at $t = 4$, i.e., $ord_3(T^2_1, 4) = w \in \Pi_1, ord_3(T^2_3, 4) = y \in \Pi_3$. On applying the transformation function on T^2, we have $P(s, T^2) = \frac{1}{6} = 0.16$ and the generated feature vector is $f = (0.33, 0.16)$. Similarly, for a given set of o patterns $\mathcal{S} = \{s_1, \cdots, s_o\}$, the transformation generates a numeric feature space of size $\mathbb{R}^{n \times o}$. Thus, the transformation defined in Definition 5.2 efficiently converts the pattern set into numeric features for datasets with large number of dimensions and samples.

5.5 Relevance Scoring

The transformed feature is based on the pattern set s drawn by a Monte-Carlo iteration and its relevance for classification is necessary to be evaluated. With our defined transformation function, a naïve solution is to convert all patterns into numeric features and perform feature selection. As such an approach is computationally expensive, it is necessary to evaluate the relevance of an ordinal pattern set right after the transformation. By estimating the misclassification rate of a classifier trained for each transformed feature, it is possible to evaluate the feature relevance. However, we aim to efficiently score the relevance and redundancy of a transformed feature without training a classifier. Hence, we estimate the misclassification rate of a feature f by applying principles of Chebyshev's inequality [KS66].

Let us consider a simple binary classification task with classes $\{c_a, c_b\}$ and feature f generated using the pattern set s. Using the theory of Chebychev inequality [KS66], the misclassification of feature f is represented using the variance $Var[f \mid c_i]$ and expected value $E[f \mid c_i]$ as,

$$error(f) = \frac{Var[f|c_a] + Var[f|c_b]}{2 \cdot (|E[f|c_b] - E[f|c_a]|)^2}. \tag{5.2}$$

The Equation 5.2 has statistical properties similar to a two-sample t-test. Its detailed proof is provided in Section 5.5.1 and we explain the intuition behind the equation with an example.

Example 5.3. Assume two multivariate ordinal patterns s_1 and s_2, where s_1 is relevant and s_2 is irrelevant for the classification.

Each ordinal pattern set is transformed into numeric features f_1 and f_2 respectively (c.f. Definition 5.2). As a relevant pattern, s_1 has a higher discriminative power, i.e., it occurs in every time series of one class (e.g., c_a) with a high probability and never occurs for the other class. Therefore, the distributions of the transformed feature f_1 for each class, exhibits a minimal variance, i.e., $Var[f_1 \mid c_a]$ and $Var[f_1 \mid c_b]$. On contrary, an irrelevant multivariate ordinal pattern set s_2, without any discriminative power to classify, occurs in different time series randomly. Hence, the distribution of the transformed numeric feature $f_2 \mid c_a$ and

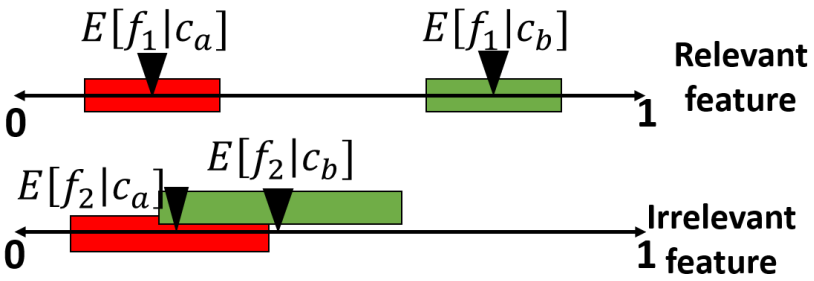


Figure 5.6: Illustration of relevant and irrelevant feature based on Equation 5.2. Where, length of the colored blocks denote the variance of distributions, inverted triangles denote the expected values of the distributions and colors denote the class

$f_2 \mid c_b$ has random peaks and lows. This leads to a larger variance in the respective distributions $Var[f_2 \mid c_a]$ and $Var[f_2 \mid c_b]$. This means, the classification error is high when the sum of the variances are large.

In real world applications, due to factors such as noise, it is possible that s_1 (which has high occurrence for class c_a) occurs in a few samples of class c_b, i.e., $Var[f_1 \mid c_b]$ is not exactly equal to zero. Hence, in addition to the variance, the distance between the expected values of the distributions is estimated, i.e., $|E[f|c_b] - E[f|c_a]|$. As we aim to extract the most distinguishing pattern set between two classes, the expected values of their distributions under each class will have a larger difference, i.e., $E[f \mid c_a] >> E[f \mid c_b]$. This large difference in the expected values helps the classification boundaries to be well-separated. This means, the classification error is large if the difference between the expected values are small.

Using Figure 5.6, we illustrate the property of a relevant feature f_1 and an irrelevant feature f_2. Our transformation function is dependent on the frequency of an ordinal pattern (c.f. Definition 5.2). Hence, for a relevant feature, the variance of its distribution under each class is smaller in comparison to that of an irrelevant feature. The variance is denoted by blocks of different lengths on the number line, whereas, the color denotes the class, i.e., red denotes $Var[f \mid c_a]$ and green denotes $Var[f \mid c_b]$. On contrary, for a relevant feature, its expected value under each class is well separated in comparison to an irrelevant feature. We denote the expected value of the distributions, i.e., $E[f \mid c_a]$ and $E[f \mid c_b]$, as inverted triangle in Figure 5.6.

Definition. 5.3: Relevance scoring

For a classification task with $C = \{c_1, \cdots, c_k\}$ classes, the lower bound of the ability to distinguish any pair of classes $c_a \in C$ and $c_b \in C$ using the transformed feature f is,

$$dis_{c_a,c_b}(f) = 1 - error(f)$$

and we define its relevance as the lowest value of all pairwise dis scores, i.e.,

$$rel(f) = min\{dis_{c_a,c_b}(f) \mid c_a \neq c_b\}.$$

Assume a classification task with classes c_a, c_b, c_c for which the $dis_{c_a,c_b}(f)$, $dis_{c_a,c_c}(f)$ and $dis_{c_b,c_c}(f)$ are computed. The three values denote the accuracy of each class. The relevance of f is defined as the minimum of the three dis scores in Definition 5.3. Intuitively, it means that feature relevance is the lowest accuracy of all pairwise scores. Hence, maximizing $rel(f)$ implies maximizing the lowest accuracy of all pairs of classes.

5.5.1 Theoretical foundations of feature relevance score based on Chebychev's inequality

Consider f as a feature extracted using a multivariate ordinal pattern set s (c.f. Definition 5.1) to classify between c_a and c_b. We denote its distribution for class c_a as $f|c_a$. The expected value and the variance of the distribution are represented as $E[f|c_a]$ and $Var[f|c_a]$ respectively. Similarly, for class c_b, we define the distribution $f|c_b$, expected value $E[f|c_b]$ and variance $Var[f|c_b]$. Without loss of generality, we assume $E[f|c_a] < E[f|c_b]$. The upper bound of mis-classification for feature f with arbitrary distribution is strongly founded by the principles of Chebyshev-inequality,

Chebychev's inequality defines the upper bound of the fraction of samples that can lie beyond a threshold $a > 0$. For any feature f, no more than $1/a^2$ of the values can be greater than a standard deviations away from the mean [KS66],

$$P(|f - E[f]| \geq a) \leq \frac{Var[f]}{a^2}, \tag{5.3}$$

where $a > 0$.

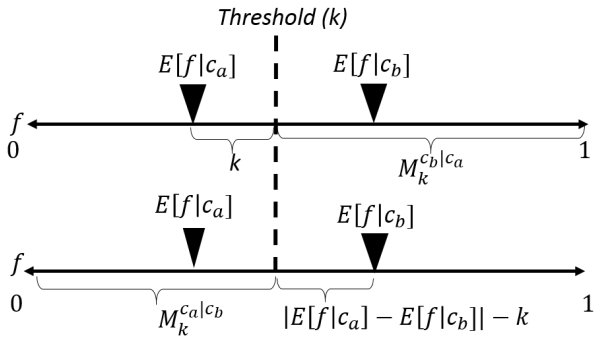


Figure 5.7: Example: A number line with limits [0,1]

Given the expected value $E[f]$ and variance $Var[f]$ of the feature, Equation 5.3 represents the probability of a sample being greater than a. The approach is commonly used for finding outliers, i.e, instances with a high probability of being greater than $E[f] + a$ are outliers. Applying the rule of Chebychev's inequality [KS66] for classification problems, an instance of feature f is classified as c_a or c_b based on the arbitrary threshold value $k \mid 0 < k < |E[f|c_b] - E[f|c_a]|$ (c.f. Figure 5.7).

We denote $P(M_k^{c_a,c_b})$ as the probability that c_a is mis-classified as c_b or c_b is misclassified as c_a. Under the assumption that $f|c_a$ and $f|c_b$ are symmetrically distributed around their expected values, a sample is classified as c_b when its expected value is greater than $E[f|c_a] + k$. Hence, to estimate $P(M_k^{c_a,c_b})$ we need to quantify the maximum number of c_a that exceed the threshold and likewise for c_b.

Lemma 5.1. *The upper bound of mis-classification is represented as,*

$$P(M_k^{c_a,c_b}) \leq \frac{Var[f|c_a]}{2k^2} + \frac{Var[f|c_b]}{2(|E[f|c_b] - E[f|c_a]| - k)^2}.$$

Proof: For a classification task, using our threshold $E[f|c_a] + k$, we aim to estimate:
$M_k^{c_b|c_a}$: a sample with class c_a is misclassified as c_b based on threshold k.
$M_k^{c_a|c_b}$: a sample with class c_b is misclassified as c_a based on threshold k.
Figure 5.7 visualizes both cases on a simple number-line, by applying the Chebychev inequality,

$$\begin{aligned} P(M_k^{c_b|c_a}) &= P((f|c_a) \geq E[f|c_a] + k) \\ &= P((f|c_a) - E[f|c_a] \geq k). \end{aligned}$$

We assumed that $f|c_a$ is distributed symmetrically around its expected value. Thus $P(M_k^{c_b|c_a})$ is half the probability that the value of $f|c_a$ has at least a distance of k to its expected value,

$$P(M_k^{c_b|c_a}) = \frac{1}{2} P\left(|(f|c_a) - E[f|c_a]| \geq k\right).$$

Using the Chebyshev-Inequality and setting $a = k$ in Equation 5.3, we can estimate an upper bound of $P(M_k^{c_b|c_a})$ as,

$$P(M_k^{c_b|c_a}) \leq \frac{Var[f|c_a]}{2k^2}. \tag{5.4}$$

On applying the same symmetric assumption on $(f|c_b)$, $M_k^{c_a|c_b}$ is half of the probability that values of $(f|c_b)$ are at least $|E[f|c_b] - E[f|c_a]| - k$ away from their expected value.

$$\begin{aligned} P(M_k^{c_a|c_b}) &= P((f|c_b) \leq E[f|c_a] + k) \\ &= P((f|c_b) - E[f|c_b] \leq E[f|c_a] - E[f|c_b] + k) \\ &= P(E[f|c_b] - (f|c_b) \geq E[f|c_b] - E[f|c_a] - k) \\ &= \frac{1}{2} P(|(f|c_b) - E[f|c_b]| \geq |E[f|c_b] - E[f|c_a]| - k) \end{aligned}$$

Comparing the above result with Equation 5.3, we derive, $a = |E[f|c_b] - E[f|c_a]| - k$. To estimate an upper bound

$$P(M_k^{c_a|c_b}) \leq \frac{Var[f|c_b]}{2(|E[f|c_b] - E[f|c_a]| - k)^2} \tag{5.5}$$

From Equation 5.4 and 5.5 this, we derive an upper bound for the total misclassification probability for c_a and c_b as,

$$\begin{aligned} P(M_k^{c_a,c_b}) &= P(M_k^{c_b|c_a} \cup M_k^{c_a|c_b}) \\ &\leq P(M_k^{c_b|c_a}) + P(M_k^{c_a|c_b}) \\ &\leq \frac{Var[f|c_a]}{2k^2} + \frac{Var[f|c_b]}{2(|E[f|c_b] - E[f|c_a]| - k)^2}. \end{aligned}$$

This means, given an optimal value of the threshold $E[f|c_a]+k$, we can calculate an upper bound for the minimal misclassification of each pair of classes. However, we cannot assume that all classifiers find such an optimal k based on the data. Moreover, finding this bound costs additional computation time. In order to be independent of the classifier and have a better efficiency, we use the fact, that $0 < k < |E[f|c_b] - E[f|c_a]|$. Due to that, the upper bound of the mis-classification grows approximately as fast as

$$\frac{Var[f|c_a] + Var[f|c_b]}{2(|E[f|c_b] - E[f|c_a]|)^2}.$$

□

5.6 Redundancy Scoring

As explained in Section 5.1, there are large number of multivariate ordinal patterns in a time series dataset. However, multiple pattern combinations can be redundant to each other, i.e., they do not provide novel information for classification. Such redundant ordinal patterns lead to lower accuracy and larger feature sets. The relevance estimation does not include the effect of redundancy. This means, two redundant patterns are scored the same based on their relevance scores.

A transformed feature f represents the probability of a particular pattern in each time series and two features are redundant if their occurrence distribution is discriminative for the same class. Assume two redundant ordinal patterns s_1 and s_2, such that their numeric transformations are f_1 and f_2 respectively (c.f. Definition 5.2). Feature f_1 signifies that the pattern s_1 occurs with a higher probability for class c_a, i.e., its values can be used to differentiate class c_a from $\{c_b, c_c\}$ (c.f. Table 5.2). On contrary, feature f_2 signifies that the pattern s_2 occurs with a lower probability for class c_a and its values can also classify c_a from the other classes. In the above example, both features are completely redundant as they are discriminative for the same class c_a. Hence, to quantify the redundancy between two features, we measure the monotonicity between them.

j	1	2	3	4	5	6	7	8	9
f_1	**0.8**	**0.88**	**0.95**	0.1	0.5	0.3	0.4	0.35	0.19
f_2	**0.2**	**0.12**	**0.05**	0.9	0.5	0.7	0.6	0.65	0.81
$class$	c_a	c_a	c_a	c_b	c_b	c_b	c_c	c_c	c_c

Table 5.2: Illustrative example of ordinal pattern redundancy

In this work, we instantiate the redundancy function with Spearmans-ρ as a measure of monotonicity [HWC13], i.e.,

$$red(f_i, f_j) = |\rho(f_i, f_j)|,$$

as it does not assume the underlying distribution of the variable. By defining the redundancy between features as an absolute value, our redundancy measure ranges between $[0, 1]$. However, other measures of monotonicity are also applicable. Information theoretic measures (e.g., Mutual Information) evaluate only the mutual dependence between the variables [CT12]. Hence, for the above example where the two features are completely redundant, $|\rho(f_1, f_2)|= 1$, but $MI(f_1, f_2) = 0.5$.

For a set of o transformed features $F = \{f_1, \cdots, f_o\}$, the redundancy of $f \in F$ against all elements in the set, i.e., $F \setminus f$, is the maximal imposed redundancy of f on the other features in the set. Hence, we compute the pairwise redundancy of f against all features in $F \setminus f$ and use its maximum. Multiple possibilities exist for combining the relevance and redundancy scores. For example, in the work of [SBS+17], the relevance of a feature is penalized for its magnitude of redundancy by computing the harmonic mean between them. Other options include subtracting the magnitude of feature redundancy from its relevance score. From experimental evaluation, we understand that both penalization techniques work well for *ordex*. Hence, we choose the latter, i.e.,

$$score(f, F) = rel(f) - red(f, F \setminus f),$$

for its simplicity. The unified *score* represents the relevance of f for classification and its redundancy w.r.t. other elements in F. Finally, the unified *score* for a set of features is the sum of all individual feature's *score*.

$$score(F) = \sum_{f \in F} score(f, F \setminus f) \tag{5.6}$$

Algorithm

From a given dataset, Algorithm 6 aims to select o relevant and non-redundant patterns by transforming them into numeric features. As mentioned in Section 5.1, it is computationally not feasible to evaluate every ordinal pattern combination. To address this computational challenge, we perform M Monte-Carlo iterations. Each Monte-Carlo iteration extracts a random ordinal pattern set s which is converted into its numeric representation using Definition 5.2 (c.f. Line 6). For the first o Monte-Carlo iterations, the algorithm draws o random pattern sets which are not scored for relevance or redundancy (c.f. Line 7). Thereon, each newly extracted pattern replaces the worst performing pattern from the set of selected patterns (c.f. Lines 11-17). The scoring of F in each iteration is performed using Equation 5.6.

For high-dimensional time series, this random pattern selection leads to the inclusion of several irrelevant (for class prediction) dimensions. Hence, in Line 5, we regulate the selection process by setting the maximum number of selected dimensions to m', i.e., $|\mathcal{I}'| \leq m'$ (c.f. Definition 5.1). The selection of s is a random process, this leads to the selection of different pattern sets in every execution. To avoid this and make the random process stable [KMB12], the overall occurrence probability of s is approximately $\alpha \in [0, 1]$. Assuming independence between dimensions, each $\Pi_i \in s$ is selected with an occurrence probability of $\alpha^{\frac{1}{|\mathcal{I}'|}}$. The influence of m' and α on the stability and prediction accuracy will be evaluated in the experimental section.

5.7 Time Complexity

Ordex begins with the conversion of the time series into ordinal representation (c.f. Algorithm 6, line 1). For an n sample, m-dimensional dataset of length l, we calculate $l-(\mathscr{d}-1)$ ordinalities for each univariate time series. As computing each ordinality involves sorting the time series values of degree $\mathscr{d}$, the total complexity for the conversion of a time series dataset into its ordinal representation is $\mathcal{O}(n \cdot m \cdot l \cdot \mathscr{d} \cdot log(\mathscr{d}))$.

Algorithm 6 Ordinal feature extraction

Input: D, o, m', α, M

1: Initialize $F = \emptyset$
2: Transform D to ordinal domain of order $\mathscr{d}$
3: **for** M Monte-Carlo iterations **do**
4: $\quad$ *Draw* $\mathcal{I}' \subseteq \mathcal{I}$ where, $\mathcal{I} = \{1, \cdots, m\}$ and $|\mathcal{I}'| \leq m'$
5: $\quad$ *Draw* $s = \{\Pi_i \subseteq \Omega_i \mid \forall i \in \mathcal{I}'\} \mid probability(\Pi_i) = \alpha^{\frac{1}{|\mathcal{I}'|}}$ (c.f. Definition 5.1)
6: $\quad$ Transform s to numeric f (c.f. Definition 5.2)
7: $\quad$ **if** $|F| < o$ **then** $F = \{F\} \cup \{f\}$
8: $\quad$ **else**
9: $\quad\quad$ $max\_score = score(F)$ (c.f. Equation 5.6)
10: $\quad\quad$ $F\_best = F$
11: $\quad\quad$ **for** $f' \in F$ **do**
12: $\quad\quad\quad$ **if** $score(\{F \setminus f'\} \cup f) > max\_score$ **then**
13: $\quad\quad\quad\quad$ $F\_best = \{F \setminus f'\} \cup \{f\}$
14: $\quad\quad\quad\quad$ $max\_score = score(F\_best)$
15: $\quad\quad\quad$ **end if**
16: $\quad\quad$ **end for**
17: $\quad\quad$ $F = F\_best$
18: $\quad$ **end if**
19: **end for**
20: **return** F

The run time of the algorithm depends on the number of iterations M. In addition, extraction of s (c.f. Algorithm 6, Line 5) depends on the maximum number of dimensions m' and maximum number of ordinalities in each dimension. Thus the complexity is represented as $\mathcal{O}(M \cdot m' \cdot \mathscr{d}!)$.

The transformation of an extracted pattern into its numeric feature involves evaluation of its probability in each time series sample with a maximum of m' dimensions. As the relevance scoring is done for each pair of classes, the complexity of our scoring function for a classification problem with k classes is represented as $\mathcal{O}(n \cdot m' \cdot l + k^2)$.

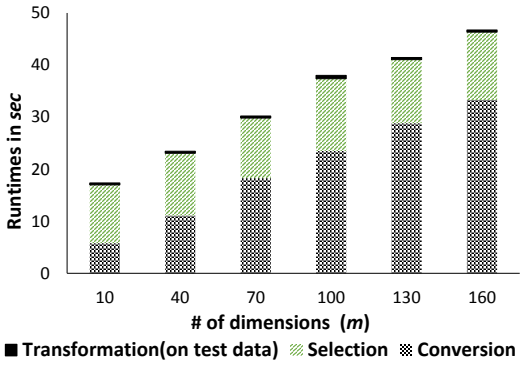
For a set of o selected features, the complexity for computing the redundancy using Spearman's correlation is represented as $\mathcal{O}(o \cdot n \cdot log(n))$ [XHHZ10]. However, as we compute the redundancy of all feature pairs, the time complexity for redundancy estimation is $\mathcal{O}(o^2 \cdot n \cdot log(n))$.

5.8 Experimental Evaluation

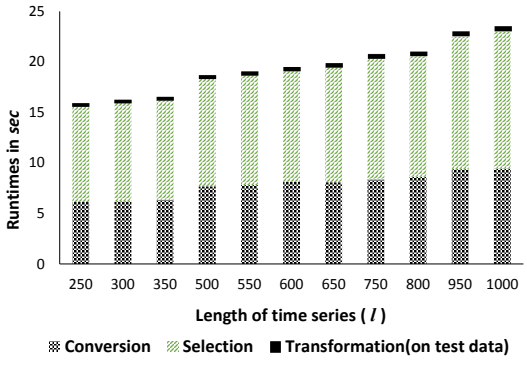
In this section we evaluate the efficiency and quality of *ordex* on multivariate synthetic, real world datasets from the UCI repository [Lic13] and a dataset from our automotive domain. Following the previous works [FJ14, NAM01, WSH06, YK09], we use accuracy on the test dataset as a quality measure. As a non-deterministic approach, we execute *ordex* five times on each dataset and plot the mean test data accuracy and run times in the experimental section below. For both synthetic and real world experiments, we use KNN (with K=5) classifier for the training and testing of the transformed features.

For generation of multivariate synthetic time series datasets, we made adaptations to the well-known cylinder-bell-funnel time series generator [Sai00]. Using the data generator, we generate separate training and test datasets. As real world datasets we use the character trajectory (3 dimensions and 20 classes), activity recognition (6 dimensions and 7 classes), indoor user movement (4 dimensions and 2 classes), occupancy detection (5 dimensions and 2 classes) and EMG Lower Limb data (5 dimensions and 2 classes) from the UCI repository [Lic13]. The EMG data was recorded with three different experimental settings, called 'pie', 'mar' and 'sen', which we treat as three different data sets. For confidentiality reasons we do not publicly provide or discuss the Bosch dataset (25 dimensions and 2 classes) we used in this work.

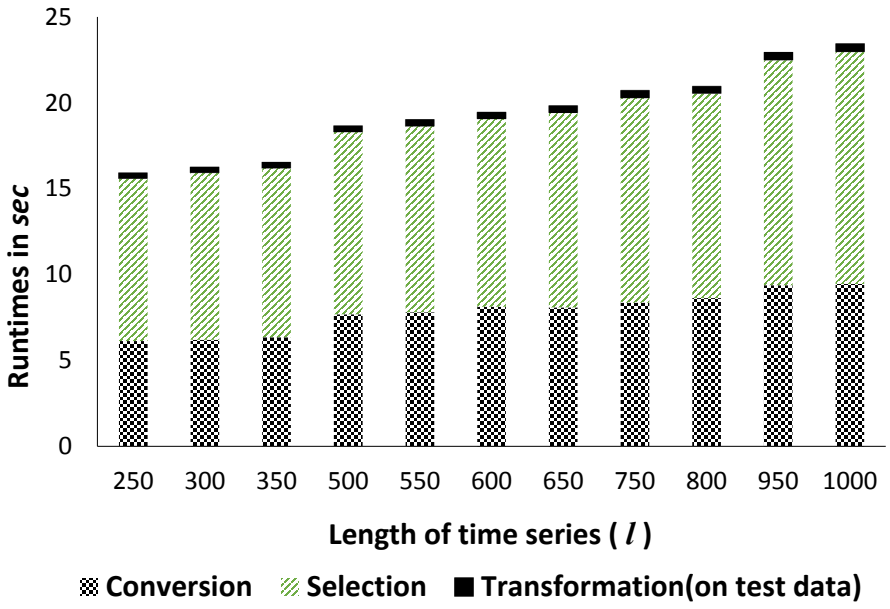
As a feature-based approach, we compare *ordex* with the various competitor techniques of the same paradigm. As competitors that extract features from the time series without evaluating its relevance to the target classes, we test Nanopoulos [NAM01], DTW [Kat16], SAX [LKL12], Wang [WSH06] and Fast Fourier Transforms. As a competitor that evaluates the feature relevance after extraction, we consider HCTSA [FJ14] approach. As a multivariate neural network based approach, we test LSTM as a competitor.



(a) Scalability w.r.t. increasing m, where $l = 600$, $n = 200$



(b) Scalability w.r.t. increasing l, where m=5, n=100, $\mathscr{A}$=5, o=10 and M=200



(c) Scalability w.r.t. increasing n, where m=5, l=600, $\mathscr{A}$=5, o=10 and M=200

Figure 5.8: Evaluation of scalability using synthetic data, where $\mathscr{A} = 5$

5.8.1 Synthetic Data sets

Scalability Experiments

We evaluate the scalability of *ordex* w.r.t. increasing dimensionality and a fixed number of time series samples. Figure 5.8 shows the breakdown analysis of time elapsed for each phase in *ordex*, i.e., conversion of training data into ordinal repre-

sentation, selection of relevant ordinal pattern sets and transformation of relevant ordinal pattern sets into numeric features on test dataset.

Our experiments in Figure 5.8a show that the run time of *ordex* scales linearly w.r.t. increasing number of dimensions. After selection of the relevant pattern sets from the training dataset, the time taken for transformation of the relevant patterns into numeric features on a test dataset is negligible. This is desirable as new samples will be transformed into static features efficiently. Scalability of *ordex* w.r.t. increasing time series length (l) and samples (n) show similar behavior (c.f. Figure 5.8b and 5.8c).

Robustness

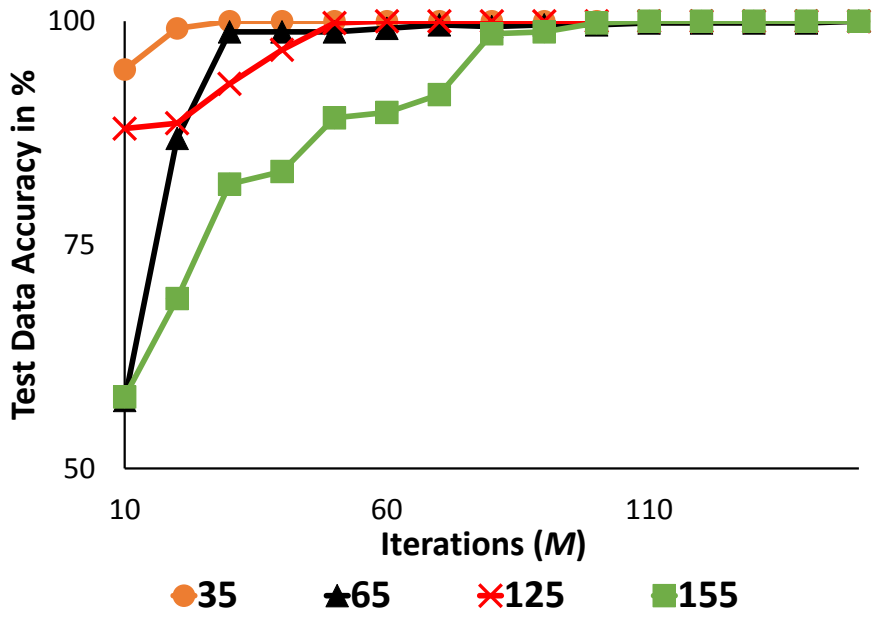
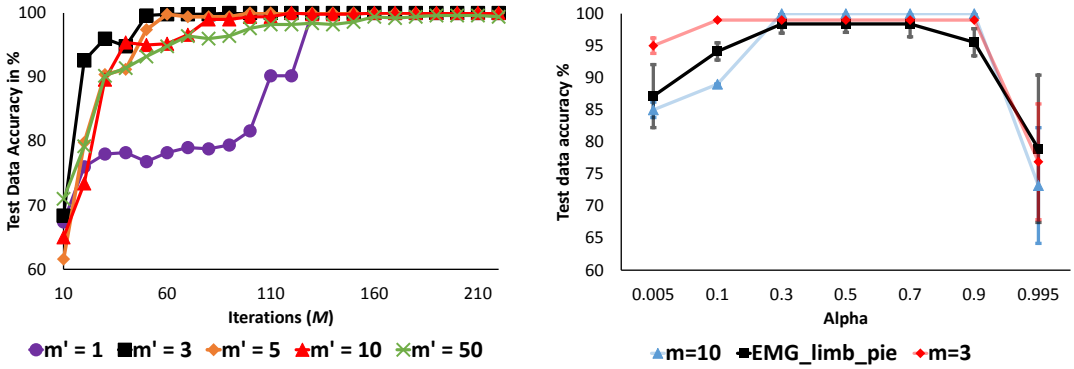


Figure 5.9: Robustness of *ordex* with varying number of irrelevant dimensions and fixed number (5) of relevant dimensions

In this section we analyze the robustness of our approach against increasing number of irrelevant dimensions. For synthetic datasets with different dimensionality ($m = 40, 70, 130, 160$), of which only five are relevant for classification, we aim to identify the influence of *ordex* on prediction accuracy.



(a) Effect of m' on prediction quality, where $\alpha = 0.3$

(b) Effect of α on prediction quality, where $m' = 3$

Figure 5.10: Analysis of the parameter using synthetic data, where $\mathscr{d} = 5$

For datasets with a large number of irrelevant features, Figure 5.9 shows that the random selection process has a higher probability of selecting irrelevant ordinal patterns in the early iterations of the selection phase. This demands several iterations (M) to reach the best accuracy. For example, a dataset with 130 dimensions required 60 iterations to reach the best accuracy and a dataset with 40 dimensions required only 20 iterations to reach the same accuracy.

Parameter Analysis

Ordex has two major parameters, m' and α. The parameter m' decides the maximum number of dimensions to include for the extraction of pattern set s. Large values of m' include several irrelevant dimensions and setting m' to very small values restrict the search space of pattern combinations to evaluate. Thus, both cases requires a higher number of iterations to identify the best combination. From experimental analysis (c.f. Figure 5.10a), we observe $1 < m' \leq 5$ to be a reasonable range to set for an optimal trade-off between quality and runtime. All real world experiments in the forthcoming section will use m' values within this range.

The α parameter decides the number of ordinalities to include from each dimension. Setting α to a large value, leads to the inclusion of several irrelevant ordinalities for classification. Hence, large α values lead to inconsistent results (higher standard deviation) and lower test data prediction quality. Setting α to lower values does not largely affect the average prediction quality. However, their

Dataset	*ordex*	Nanopoulos	DTW	SAX	Wang	FFT	HCTSA	LSTM
EMG limb sen	**93.33 ± 3.1**	33.3	83.3	33.3	92.3	66.7	50	50 ± 0
EMG limb pie	**85 ± 2**	16.7	33.3	50	66.6	33.3	50	66.6 ± 0
EMG limb mar	**95 ± 3.6**	83.3	66.7	95	66.6	66.7	92	63.5 ± 0
Character	75.37 ± 1.7	27.1	**88.3**	8.2	70	17.6	25.4	11.98 ± 5.1
Activity recognition	**100± 0**	44.5	**100**	2.8	91	**100**	17.4	**100 ± 0**
User Movement	**57.98 ± 1.9**	45.2	46.8	52.4	45.2	42.9	50.8	47.6 ± 0.8
Occupancy	**94.1 ± 1.9**	63.6	**94.1**	78.4	78.4	70.6	75.4	84.7 ± 8.13
Bosch	**97.08 ± 1.5**	37.7	**	60.2	95.3	59.2	**	56.6 ± 3.4

Table 5.3: Test data accuracy in % with *ordex* $\not{d} = 5$ and $m' = 3$. SAX word size and alphabet size is 3. LSTM of maximum epochs 100 and mini-batch size 10. Experiments that had run times more than one day are denoted as **

Dataset	*ordex*	Nanopoulos	DTW	SAX	Wang	FFT	HCTSA	LSTM
EMG limb sen	130.3	8.3	840.3	298.1	1512	8.6	9498	372
EMG limb pie	130	7.9	830.5	266.9	1087	7.9	4088	450.6
EMG limb mar	100.3	7	619.4	278.9	1232	7.1	11999	272
Character	105.3	23	852	458.3	5020	22.06	5511	263
Activity recognition	210.3	5.6	19.9	166.3	1235	5.2	797.2	1808
User Movement	155	2.1	46.8	111.61	428.4	2.8	180.15	174
Occupancy	126.7	1.2	15.6	113.4	49.38	1.1	399.4	125
Bosch	2775.7	344.3	**	4920.4	6876	265.2	**	7335

Table 5.4: Runtime in *sec*, experiments that had run times more than one day are denoted as **

standard deviation over five test runs was high. Our experiments on synthetic data in Figure 5.10b shows $0.3 \leq \alpha \leq 0.9$ range to be a reasonable α value for datasets with different dimensionality. In addition, using EMG Lower Limb Pie dataset, the Figure 5.10b shows that this range of alpha value is practically applicable for real world data.

5.8.2 Real world Data sets

Table 5.3 compares the prediction accuracy of various approaches against *ordex*. Overall, we observe that considering relevance and redundancy during feature extraction improves the prediction quality. In addition, by including the multivariate nature of ordinalities, *ordex* shows better prediction accuracy w.r.t. the competitor approaches on several datasets. In the character dataset, *ordex* was the second best amidst competitor approaches falling behind DTW. However, DTW approach [Kat16] has higher run times for dataset with large number of samples, e.g., the

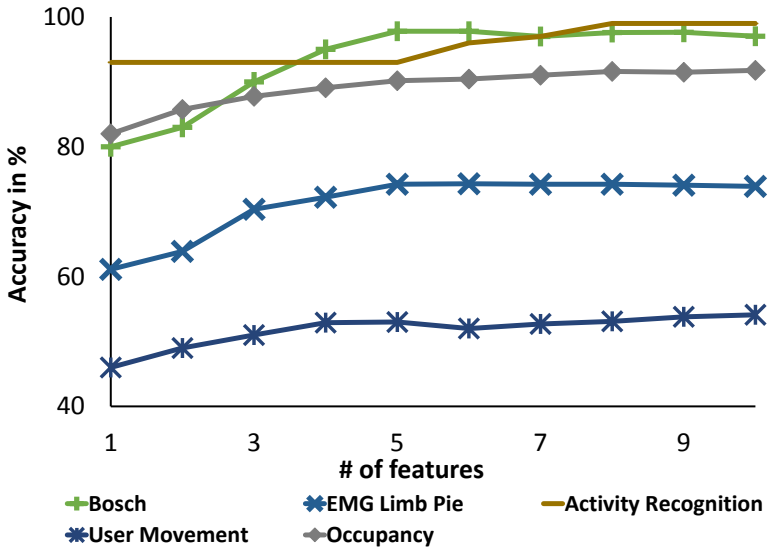


Figure 5.11: Accuracy of top 10 features of *ordex*

DTW approach took more than a day for computations on our Bosch dataset with 5722 time series samples.

Table 5.4 compares the run times (test and train) of the various approaches against *ordex*. As discussed in Section 5.3, *ordex* evaluates a combinatorial search space. Considering the complexity of the challenge, *ordex* performs reasonable w.r.t. run times in Table 5.4. By performing the feature extraction and evaluation simultaneously, *ordex* has lower run times in comparison to HCTSA that performs feature selection after extraction of a high-dimensional feature space from the time series. As shown in Section 5.8.1, the major execution time of *ordex* is dominated by the conversion and selection process. Considering the improvement in the prediction quality with negligible time for transforming the relevant and non-redundant ordinalities into numeric features (c.f. Figure 5.8a), *ordex* is a better choice than the competitor approaches.

Redundancy Evaluation

The ground truth of feature redundancy is unknown for real world datasets. Using redundant features does not provide novel information for classification, i.e., redundant features do not improve the classification accuracy. Thus, following the work of [SBS+17], we evaluate redundancy based on the classifier accuracy in Figure

5.11. For a set of o best features extracted using *ordex*, the top scored features of *ordex* are relevant and non-redundant. Hence, the initial features have increasing prediction quality in Figure 5.11. For example, EMG Limb Pie dataset requires 6 features, after which the features are relevant but have redundant information and the classifier accuracy does not improve.

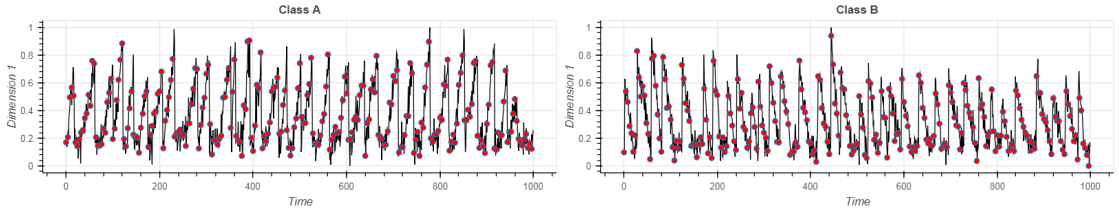
5.9 Illustration of ordinality

In this section we aim to visually illustrate the potential of ordinal patterns for multivariate time series classification on synthetic and real world datasets.

Synthetic dataset with 12 time series samples (n) of dimensionality (m) 4 and length (l) 1000 is generated for classification between class A and B. The synthetic dataset consists of two relevant (*Dimension* 1 and *Dimension* 2) and two irrelevant (*Dimension* 3 and *Dimension* 4) dimensions (c.f. Figure 5.12). For $\mathscr{d} = 3$, an increasing ordinality ($u = 012$) in *Dimension* 1 and 2 together is characteristic for class A (c.f. Figure 5.12a and 5.12c). On the other hand, a decreasing ordinality ($z = 210$) in the relevant dimensions are characteristic for class B (c.f. Figure 5.12b and 5.12d). Our transformation function (c.f. Definition 5.2) scores the discriminative power of ordinal patterns based on its frequency of occurrence in each class.

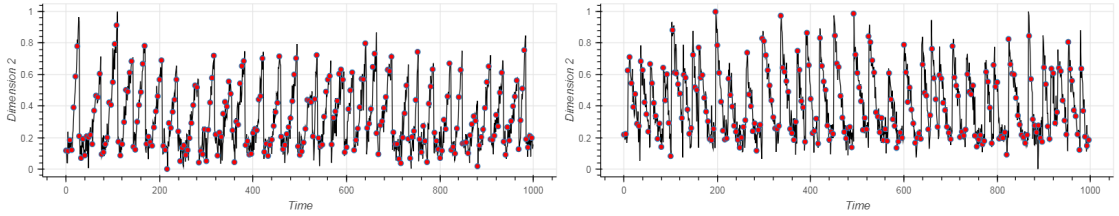
In Figure 5.13, we visualize the frequency of ordinalities in each dimension. The visualization was performed as follows:
(1) **Step 1**: Append all the time series samples that represent class A into a single long series.
(2) **Step 2**: To this long time series, we additionally append all time series samples that represent class B.
Hence, we have a long time series such that it has consecutive series representing class A and after a certain point (denoted as x in Figure 5.13) it consecutively represents class B. This long time series was transformed into its ordinal representation of degree ($\mathscr{d}$) 3[†] and the frequency of ordinality within a fixed duration (30

[†]The time series dataset is transformed also based on delay parameter τ, which was set to 3. Its role in ordinal transformation is discussed in Section 5.10)



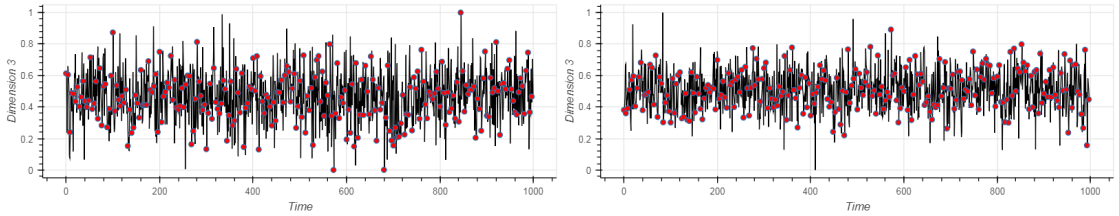
(a) Time series sample of Dimension 1 relevant for Class A

(b) Time series sample of Dimension 1 relevant for Class B



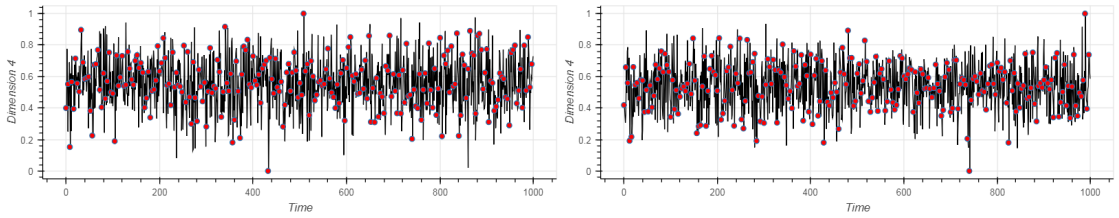
(c) Time series sample of Dimension 2 relevant for Class A

(d) Time series sample of Dimension 2 relevant for Class B



(e) Time series sample of Dimension 3 irrelevant for Class A

(f) Time series sample of Dimension 3 irrelevant for Class B

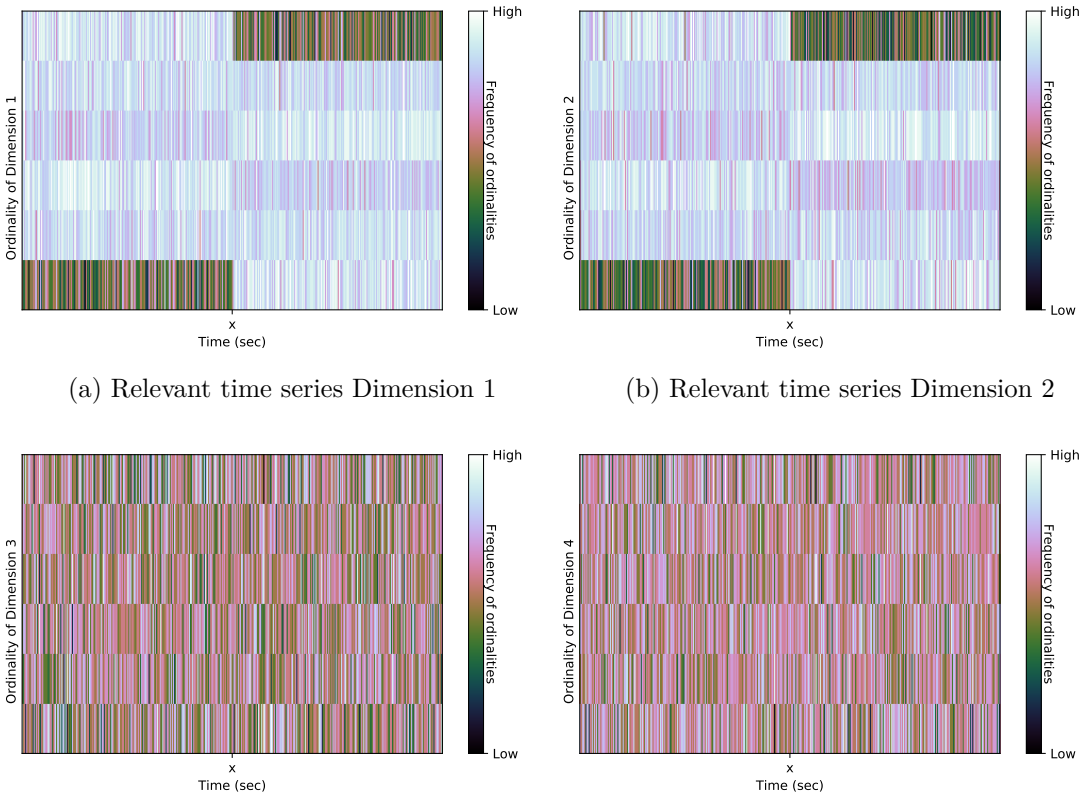


(g) Time series sample of Dimension 4 irrelevant for Class A

(h) Time series sample of Dimension 4 irrelevant for Class B

Figure 5.12: Synthetic dataset for illustration of ordinal patterns of $d = 3$

seconds) are plotted as a heat map in Figure 5.13. The heat map shows that until the point x, the ordinality u occurs with a high probability in *Dimension*1 and 2. Beyond this point, the probability of u is low but that of ordinality z is high. Ordex exploits this change in the frequency of time series ordinalities for feature extraction.



(a) Relevant time series Dimension 1

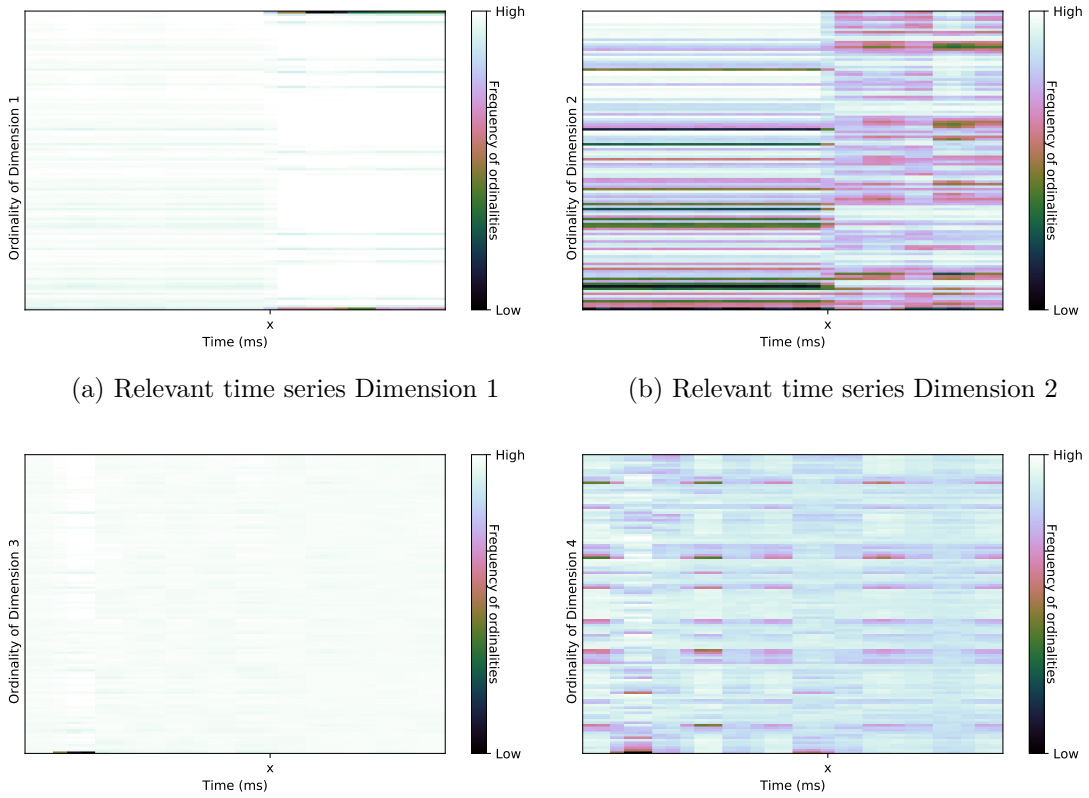
(b) Relevant time series Dimension 2

(c) Irelevant time series Dimension 3

(d) Irelevant time series Dimension 4

Figure 5.13: Frequency of ordinalities denoted using color bar for the synthetic dataset shown in Figure 5.12. Where, x-axis and y-axis denote the time and ordinalities. The x-mark on the time axis signifies the point where the class changes from A to B

Real world dataset from Bosch is subjected to the same steps explained in synthetic dataset. The lengthy series are transformed in to ordinal domain of degree ($\mathscr{d}$) 5. The ordinalities are visualized in Figure 5.13. We understand that *Dimension* 1 and 2 of the Bosch dataset consists of ordinalities that are discriminative for classification. On contrary, the *Dimension* 3 and 4 do not exhibit ordinalities that are highly characteristic for a particular class.



(a) Relevant time series Dimension 1

(b) Relevant time series Dimension 2

(c) Irrelevant time series Dimension 3

(d) Irrelevant time series Dimension 4

Figure 5.14: Frequency of ordinalities denoted using color bar for Bosch multivariate time series dataset. Where, x-axis and y-axis denote the time and ordinalities. The x-mark on the time axis signifies the point where the class changes

5.10 Parameters of Ordex

In this chapter (c.f Equation 5.1 in Section 5.3) we defined ordinality at time t in a time series X as,

$$\mathbb{O}_{\mathscr{d}}(X,t) = (rank(X[t]), rank(X[t-1]), ..., rank(X[t-(\mathscr{d}-1)])).$$

In the original work [BP02] that introduced the property of ordinality, the computation of an $\mathbb{O}$ at t involves an additional delay parameter $\tau \geq 1$. That is,

$$\mathbb{O}^{\tau}_{\mathscr{d}}(X,t) = (rank(X[t]), rank(X[t-\tau]), ..., rank(X[t-(\mathscr{d}-1)\tau])). \quad (5.7)$$

The τ parameter was also used in our work for all synthetic and real world experiments in Section 5.8 and we list the parameters values in Table 5.5.

Dataset	$\mathscr{d}$	M	m'	α	o τ
EMG limb sen	5	200	3	0.5	10 4
EMG limb pie	5	300	3	0.1	15 4
EMG limb mar	5	200	3	0.3	20 4
Character	5	300	3	0.1	5010
Activity recognition	5	100	2	0.3	20 3
User Movement	5	300	3	0.8	30 4
Occupancy	5	100	3	0.5	10 4
Bosch	5	200	3	0.1	10 4

Table 5.5: Real world data experiment parameter settings

5.11 Summary

In this chapter we proposed a feature-based time series classification approach called *ordex* that is purely based on the ordinality of the raw time series. *Ordex* extracts features based on co-occurrence of ordinalities from multiple dimensions. Hence, the interactions of multiple dimensions in the time series data are honored. The extracted features are evaluated for relevance based on our novel and efficient scoring methodology. In addition to the relevance, we evaluate the monotonicity of extracted features to estimate feature redundancy. Finally, the relevance and redundancy scores are combined to exemplify the importance of the feature to the classification task and the novelty with respect to other extracted features. By scoring relevance and non-redundancy, *ordex* achieves better prediction quality with fewer features.

The results of various state-of-the-art feature-based algorithms on the synthetic and real world datasets show that our method is suitable for multivariate time series datasets. For high-dimensional time series our approach efficiently converts the relevant ordinalities into features. Therefore, in real world applications, relevant

and novel ordinalities are encoded into static features. These features can be used for various data mining tasks and analysis.

Approaches presented in the previous chapters, i.e., RaR (c.f. Chapter 4) and *ordex*, involves high number of Monte-Carlo iterations. The large number of computations in addition to the dimensionality makes the task of understanding the feature selection algorithm difficult. In the forthcoming chapter we introduce a software framework that helps the user to understand multivariate correlations. The aim of such a software tool is to make multivariate correlation analysis transparent.

Chapter 6

Understanding Multivariate Correlations

6.1 Motivation

Feature selection aims to score the importance of a feature based on its correlation with the target. However, a causal relationship cannot be inferred from all correlations. For example, both wrinkles and cancer risk increase with age, but wrinkles does not cause cancer or vice-versa [LWIG06]. Here, the correlation between wrinkles and cancer risk is a mere statistical coincidence and not a causality. This makes it essential for the domain experts to understand the multivariate correlations in the datasets and decide if a correlation is causal or not. In this context, the number of feature combinations grows exponentially with the dimensionality of the feature space. This hinders the user's understanding of the feature-target relevance and feature-feature redundancy.

In order to provide a smaller yet predictive subset of features, a large variety of existing approaches [GE03, Qui14, RŠK03, KMB12, SBS+17] compute the relevance of each feature to the target class, as well as the redundancy between features. However, the user does not get an overview of all correlations in the dataset. Furthermore, the selection process is non-transparent because the reason for a feature's relevance or redundancy is not explained by these algorithms. Hence, the first challenge for explaining the feature selection process is to present relevance and redundancy jointly in an informative layout. The second challenge is to guide the user in understanding how features are correlated as opposed to merely returning a correlation score. We address these two challenges by contributing a **F**ramework for **Ex**ploring and **U**nderstanding **M**ultivariate Correlations (FEXUM)*, that provides:
(1) A visual embedding of feature correlations (relevances and redundancies).
(2) User-reviewable multivariate correlations.

This leads to a more comprehensible selection process in comparison to state-of-the-art tools tabulated in Table 6.1. While most tools focus on fully-automated statistical selection of features, with FEXUM we aim at explaining the feature se-

*Adapted by permission from Springer Nature: Framework for Exploring and Understanding Multivariate Correlations in the proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), 2017 [KRT+17]

Tools	Relevance	Redundancy	Correlation overview	Correlation explanation
KNIME	✓	✗	✗	✗
RapidMiner	✓	✓	✗	✗
Weka	✓	✓	✗	✗
FEXUM	✓	✓	✓	✓

Table 6.1: Comparison of feature selection tools

lection algorithm. KNIME is a renowned tool that offers filter-based feature selection using linear correlation and variance measures. However, without customized extensions, it does not address feature redundancy during selection. RapidMiner and Weka take redundancy into account, but do not provide an overview of all feature correlations. Additionally, they do not explain the reason for the relevance of a feature.

FEXUM is an application that allows instant access with a web browser. We achieve this by basing our infrastructure on AngularJS and the Django web framework. To ensure scalability for large datasets, we distribute computations to multiple machines with Celery.

6.2 Correlation Summary

FEXUM being a correlation analysis tool, it requires instantiation of the correlation function. In this work we instantiate the correlation function based on the principle of statistical dependence (see Chapter 4, Definition 4.1 and 4.3). However, the framework allows instantiation with other relevance scoring methods as well. In order to provide an overview of all correlations in a high-dimensional dataset, we embed the feature to target relevance and pairwise feature redundancy in a force-directed graph layout.

Our visualization provides a layout in which a smaller distance of a feature to the target denotes a higher relevance, while a smaller distance between two features denotes a higher redundancy. We interpret this as a graph in which nodes represent features and weighted edges represent distances. These distances do not obey the

triangle inequality and therefore cannot be mapped to metric space. Starting from a random placement of features in the layout, our algorithm applies forces proportional to the difference between their current distance and their correlation-defined distance. We run this simulation for a defined number of iterations and over each iteration the distances are updated in the graph layout, thereby minimizing the waiting time for the user.

The force-directed graph allows soft-clustering of features. That is, it does not perform binary cluster assignment. Instead it provides the degree to which a feature belongs to a bi-cluster. We exploit this property of the force-directed graph to visualize feature redundancy. This enables an expert to select a feature from each cluster in accordance with the domain knowledge.

Using the Wisconsin Breast Cancer (Diagnostic) dataset [Lic13] from the UCI repository, in Figure 6.1 we show the summary of correlations in a dataset. The dataset has 32 dimensions, 569 samples and two-classes (malignant or benign, denoted as 0 or 1). The target vector is denoted as *diagnosis*.

From Figure 6.1, we understand that features such as *perimeter_worst* and *area_worst* achieve comparable relevances and are redundant to each other. On contrary, features such as *texture_worst* and *concave_points_se* which are located farther off from the target are less relevant features. Hence, the visualization provides a consolidated summary of all feature correlations in a dataset.

6.3 Multivariate Correlations

In Section 6.2 we introduced the concept of embedding the feature correlations in a force-directed graph for understanding all correlations in the dataset. However, it does not provide an understanding of the multivariate correlations. In this section we elaborate the additional attributes of FEXUM that enhances the user's understanding of the multivariate correlations.

Having selected a feature set $S \subset \mathcal{F}$, the goal of our framework is to provide insight into its correlations with the target Y. FEXUM was instantiated with our relevance scoring methodology discussed in Chapter 4. Hence, we aim to explain the nature of correlation using the same. That is, the average divergence between the marginal probability of Y and the probability of Y conditioned on different

Figure 6.1: Features drawn using a force-directed graph (right), with the target highlighted in green. An analysis view of two features (left) for inspecting the correlations.

value ranges of S. For every feature $f \in S$, a value range of interest can be chosen. If a feature f correlates with the target Y, there exists a value range of f which changes the distribution of Y in contrast to Y's marginal distribution [KMB12]. We follow the same principle for pairwise redundancy estimation (c.f. Definition 4.3 in Chapter 4). However, evaluating the redundancy for every feature pair, i.e., $red(f_i, f_j) \mid f \in \mathcal{F}$ and $i \neq j$, is inefficient. Hence, in Section 6.4, we explain a simple heuristic for fast approximation of pairwise redundancy.

Using a set of selected features, $S = \{perimeter\_worst, concavity\_mean\}$, we demonstrate the effectiveness of FEXUM in understanding the multivariate correlation. The target's marginal probability distribution and the distribution conditioned on the selected value ranges are rendered in Figure 6.1:(1). The conditional slices that contribute for relevance score are highlighted on the feature's histogram in Figure 6.1:(2). By this way, we can easily find the influential value ranges. In addition, the probability distribution is updated in real time for the user defined selection of conditional slices. This information enables the user to understand which value ranges from the selected feature contributes for the class prediction. In the above example, $108.6 > perimeter\_worst < 171.1$ and

0.0. $>$ *concavity_mean* $<$ 0.04 (c.f. Figure 6.1:(3)) combined together is statistically dependent on target class 0 (c.f. scatter plot in Figure 6.1:(4)). Hence, we understand that the multivariate correlation between these two features at the observed range is highly correlated to the particular class. Similarly, the tool enables understanding of complex interactions between more than two features and multiple classes in a dataset.

6.4 Redundancy approximation for FEXUM

Our software Framework for Exploring and Understanding Multivariate Correlations (FEXUM) embeds the feature-target relevance and feature-feature redundancy on a force-directed graph layout. Computing all the pairwise feature redundancies can be highly time-consuming for large number of dimensions. For example, for a dataset with 1000 features, there are $\binom{1000}{2}$ different feature pairs to evaluate. As evaluating every pair is not efficient and leads to long waiting time for the users, it is preferable to have a faster and an approximate solution. Hence, we explain a simple heuristic for an efficient approximation using an example.

Algorithm 7 Pairwise redundancy estimation for FEXUM

Input: $\mathcal{F}, M, k$
1: $red\_collection = empty\ dictionary\ of\ size\ \binom{d}{2}$
2: **for** $i = 1 \rightarrow M$ **do**
3: Sample $S \mid S \subseteq \mathcal{F} \wedge |S| \leq k$
4: Sample a feature $f \in \mathcal{F} \mid f \notin S$
5: Compute redundancy $score = red(f, S)$ (c.f. Equation 4.7)
6: **for each** $j \in S$ **do**
7: $red\_collection(f, j) = min(score, red\_collection(f, j))$
8: **end for each**
9: **end for**
10: **return** *red_collection*

Example 6.1. Let us assume a dataset with four dimensions $\mathcal{F} = \{f_1, f_2, f_3, f_4\}$, such that f_1 is highly redundant to f_3.

Using the dataset from Example 6.1, a random subset $\{f_2, f_3\}$ and feature f_1 is drawn (c.f. Table 6.2) to yield a *score* of 0.95 in the first Monte-Carlo iteration. The redundancy *score* is high as we assumed f_1 to be redundant with f_3. However,

at this point we do not have any evidence to suggest if the redundancy score is high because of informational redundancy between (f_1, f_2) or (f_1, f_3). Hence, the dictionary of redundancies $red\_collection(f_1, f_2)$ and $red\_collection(f_1, f_3)$ are both updated with 0.95 based on the first iteration (c.f Table 6.3).

i	S	f	$score(S, f)$
1	$\{f_2, f_3\}$	f_1	0.95
2	$\{f_2, f_4\}$	f_1	0.6
3	$\{f_1, f_3\}$	f_2	0.35

Table 6.2: Illustrative example of Algorithm 7

$red\_collection(f, j)$	Approximated redundancy
$red\_collection(f_1, f_2)$	min (0.95,0.6,0.35) =0.35
$red\_collection(f_1, f_3)$	0.95
$red\_collection(f_1, f_4)$	0.6
$red\_collection(f_2, f_3)$	0.35
$red\_collection(f_2, f_4)$	*empty*
$red\_collection(f_3, f_4)$	*empty*

Table 6.3: *red_collection* dictionary based on our example in Table 6.2

The system of redundancy assignment has clearly led to overestimation of $red\_collection(f_1, f_2)$. That is, we have assigned a large redundancy score to (f_1, f_2) pair for the redundancy of information that (f_1, f_3) feature pair exhibits. This is because our dependency scoring methodology is monotonic in nature, i.e., $red(j \in S, f) \leq red(S, f)$ (c.f. Lemma 4.2). Eventually, in further iterations, we aim to enhance the estimate by sampling more subset combinations. For instance, we observe that $red\_collection(f_1, f_2)$ is assigned 0.6 (using $i = 2$) and 0.35 (using $i = 3$). The overestimation is now compensated by assigning the approximate redundancy between the (f_1, f_2) feature pair as the minimum of all values in that collection, which yields to 0.35. From this example, we intend to show that over multiple Monte-Carlo's the algorithm aims to balance the overestimation.

6.5 Summary

In this work we introduced the idea of embedding the correlations, i.e., relevance of the features to the target and the redundancy between features, on a force directed graph layout. By this way we present a consolidated summary of the correlations for the domain experts to validate. The major aim of the framework is to aid the users to efficiently explore and evaluate the multivariate interactions in the dataset. The framework allows the user to corroborate the feature relevance score by analyzing several individual value ranges, which can be chosen based on the framework's recommendations or expert knowledge. Since we support multivariate correlations, the current subset can be iteratively expanded in a similar fashion.

As demonstrated, the framework guides in exploration and review of correlations. This is of great importance especially for automotive applications where we predominantly face challenges with dependency oriented data. The domain experts spend a large effort to understand the complex multivariate dependencies between the driver behavior, sensor signals and external factors. The tool bridges the gap between the domain experts and data mining algorithms presented in the previous chapters. The software tool was used by various engineering departments from different application domains at Bosch. Overall, the common feedback was:

- FEXUM enhances the user's understanding of all correlations in a high-dimensional dataset.
- FEXUM aids the user to decide if a correlation is a mere statistical coincidence or causality.

Chapter 7

Summary and Future Research

7.1 Summary

In the following chapter, we summarize the major research results and describe an outlook of possible future research directions in the topic of multivariate correlation analysis. In Chapter 1, we have elaborated the role of feature selection and extraction steps in the KDD process. From the aspect of training a supervised prediction model, both steps are strongly founded by the principles of correlation analysis between the high-dimensional feature space and the target. In addition, we have given a brief overview of the goals, challenges and contributions of this dissertation such as:
(1) Algorithmic framework for selection of multiple relevant and diverse views of the feature space in Chapter 3.
(2) Algorithmic framework for including higher-order interactions for relevance and redundancy estimation in mixed dataset in Chapter 4.
(3) Algorithmic framework for multivariate feature extraction in time series applications in Chapter 5.
(4) Software framework for understanding and evaluating multivariate correlations in Chapter 6.
The importance of our contributions were substantiated and motivated using examples from a variety of application domains.

In Chapter 2, we have elaborated how feature selection encounters the curse-of-dimensionality. With an illustrative example, we explained that a large number of features may lead to poor prediction quality of a classifier. Hence, feature selection is an essential step to improve the prediction quality. On the other hand, we also discussed the importance of feature extraction in various application domains. For the dissertation to be self-contained, we have briefly summarized the strength and weaknesses of the traditional methods with an introduction to the commonly used notations in the literature. In addition, we provided a discussion of the research gaps that this dissertation aims to bridge w.r.t. the existing selection and extraction methodologies.

Following the hybrid paradigm, in Chapter 3 we presented DS3, a feature selection method for selection of multiple relevant subsets by capturing the different intrinsic properties in the dataset and also enhancing the diversity between the fea-

ture subsets. Technically, we exploit the ability of different correlation measures to evaluate different intrinsic properties between the features and the target to generate initial candidates. These initial candidates are systematically combined such that the diversity between the subsets and the prediction quality are enhanced. From experimental analysis on synthetic and real world data, we demonstrated that such an approach has many advantages. In comparison to the traditional feature selection methods, we aim to capture the local interactions in the dataset with multiple correlation measures and enhance diversity. However, traditional methods predominantly provide a single relevant projection of the dataset or they do not address both criterion, i.e., multiple correlation measure and diversity enhancement.

In Chapter 4 we extended the idea of multiple views for datasets with mixed data types, i.e., continuous and categorical. In comparison to our previous contribution, we enhanced the algorithmic efficiency by following the filter paradigm. Our algorithm RaR aims to evaluate a feature's relevance by including its higher-order interactions and the magnitude of redundancy. Principally, the presented approach evaluates the relevance of a feature based on its interactions with features in multiple subset combinations. Hence, we deduce the feature relevance by combining the knowledge from multiple views of the dataset. As a measure of statistical dependency between a feature subset and the target, we use the divergence between the conditional and marginal distributions. This implicitly evaluates the mutual information between the subset and the target. As the probability estimations are not affected by the data type, we show that the method is applicable for both continuous and categorical data types. The estimated feature relevances are penalized based on its magnitude of redundancy. Finally, the features are ranked based on a combined relevance and redundancy score. With rigorous experiments on synthetic and real world data, we test the scalability, robustness, ranking quality and the redundancy of RaR in comparison to the traditional approaches. By estimating the higher-order interactions between more than two features and evaluating redundancy of features in mixed datasets, our work bridges an important gap in the research community.

In Chapter 5 we introduced *ordex* to address the multivariate correlations in time series data type. In comparison to both the aforementioned contributions which are limited to steady-state data, we enhance the multiple views idea to time

series applications. Unlike the steady state data, time series needs extraction of specific discriminative subsequences. However, the fundamental task of correlation analysis remains the same. In this work, we extract the multivariate subsequences based on the property of ordinality in the time series. We introduced the concept of multivariate ordinal patterns in time series to transform the subsequences in to features. We efficiently evaluate the relevance of the extracted ordinalities for the target prediction based on our scoring methodology inspired from the principles of Chebychev's inequality. From our experiments on synthetic datasets, we show that our approach efficiently transforms the relevant ordinalities into features that can be used for different prediction tasks. In addition, we experimentally evaluate the robustness of our approach with a large number of noisy time series. Our contributions in Chapter 5 addresses various research gaps such as, (1) Introduction of the concept of multivariate ordinal patterns and (2) Simultaneous extraction and evaluation of relevance and redundancy for feature extraction in multivariate time series.

In Chapter 6 we introduced a software tool that is equipped with different exploratory analysis methods to help users in understanding and visualizing the correlations in the data. All traditional feature selection methods provide a set of relevant and non-redundant feature subsets or feature ranks or weights. However, in this dissertation our software tool FEXUM embeds the feature-target relevance and feature-feature redundancy on a force-directed graph layout. This provides an user with a consolidated summary of all correlations in the dataset. With such a summary, the domain experts learn how different features interact with the target and amongst themselves. As discussed in Chapter 1, the task of multivariate correlation analysis is often perceived as a black-box methodology by the domain experts. However, FEXUM provides interactive visualization options that help a domain expert to comprehend higher-order interactions in the dataset. Therefore, this dissertation acts as a bridge between the research and the application worlds. The software tool was well-received by a variety of engineers from different application and research domains at Bosch.

Throughout the dissertation, we demonstrated the quality and efficiency of our algorithmic frameworks introduced in Chapter 3, 4 and 5 with theoretical assessment and experimental evaluations. The theoretical assessment involves estimation of the algorithmic time complexity. The experimental evaluation involves

experiments on synthetic and real world data. We compared the quality of our contributions with several state-of-the-art algorithms for feature selection and extraction algorithms. Similarly, we demonstrate the effectiveness of our software framework in Chapter 6 using a real world dataset.

7.2 Future Research Directions

In this section, we aim to propose possible future research directions for data mining and high performance computing research groups.

Enhancement of the transformation phase to inherently handle datasets with (1) Missing values, (2) Sparsity and (3) Class imbalance are few plausible and immediate extensions of RaR that we propose. Handling datasets with any of the aforementioned properties requires application of preprocessing techniques prior to feature selection. For example, interpolation of missing values, under-sampling or oversampling of data to handle class imbalance are common preprocessing techniques. Disadvantages of such methods include loss or distortion of information due to its artificial nature [WMZ07]. Enriching the transformation phase of the KDD process to address these challenges will largely reduce the effort of industries that spend millions of dollars on data preprocessing techniques [Red96]. Overall, the proposed research direction will enhance the transformation phase by reducing the time and effort spent on the preprocessing phase of the KDD process (c.f. Chapter 1). As an extension of RaR, we currently pursue on the idea of assigning weights to the correlation function based on the class distribution to encounter class imbalance.

Active learning of search space: In Chapter 4 we introduced the Relevance and Redundancy ranking (RaR) where random feature subsets are drawn for each Monte-Carlo iteration. In such random sampling techniques, each iteration is independent of the other. That is, the second iteration does not draw a subset based on the knowledge we gained from the previous iteration. However, incorporating the relevance information of subsets sampled in the previous iterations will lead to improvised exploration of the exponentially growing search space. The literature

of active sampling and Monte Carlo Tree Search (MCTS) aims to address this challenge by acquiring the most informative regions in the search space [Agg15, CL18]. The proposed extension is also applicable for the feature extraction methodology *ordex*, which we introduced in Chapter 5. The usefulness of such exploration methods for RaR and *ordex* is yet an open research question. Hence, application and evaluation of active sampling and MCTS techniques for RaR can be an immediate extension that has a positive impact on algorithmic efficiency. The major challenge is to design an ideal criterion function that can steer the search space exploration.

Inclusion of amplitude for ordinal analysis: In Chapter 5, we introduced the concept of using ordinality as a property for feature extraction from multivariate time series. Ordinality of a series is assigned based on the ordinal relation between the values. However, in automotive applications, we observed that two series can exhibit same ordinal-relationship with large differences in their amplitudes. This can contribute to higher classification error. Hence, we propose the extension of ordinal analysis to include the effect of amplitude as a possible future research. Such an extension will further enrich the feature extraction process. As discussed in Chapter 5, there exists an exponential number of multivariate ordinal pattern combinations. And including the effect of signal amplitude will further blow up the search space and poses a greater challenge.

High performance computing: All techniques presented in the work include Monte Carlo iterations. As a randomization component is involved in it, a larger number of iterations always enhance the search space exploration and positively influences the quality of selection. To speed up the process, we introduced the idea of parallelizing the Monte Carlo's on multiple processor threads in Chapter 4. With the mounting interest of the Graphical Processing Unit (GPU) hardwares, a study on the acceleration potential of the selection process using such hardware will benefit different application domains. The major challenge is to perform the computations with minimal latency between host (CPU) and device (GPU). Secondly, with limited global and shared memory of the GPUs, the challenge is to design the parallelization by ensuring its optimal usage.

Niche research: In automotive applications, time series arise not only from different sources, but also with different data types. For example, the engine state is a categorical time series which influences several components in the automotive engine. In this work we focus on multivariate continuous time series values. However, a recommended research direction is to perform feature extraction in time series dataset with continuous and categorical data types. We consider this as a niche research area because current time series datasets are predominantly continuous (e.g., 75 out of 81 time series datasets in the UCI repository are continuous in nature). However, in the past few years, time series with mixed data types are getting common in our automotive application domains. Hence, addressing this problem will further enhance the future research frontiers.

Bibliography

[AEG14] Sercan Arik, Sukru Burc Eryilmaz, and Adam Goldberg. Supervised classification-based stock prediction and portfolio optimization. *arXiv preprint arXiv:1406.0824*, 2014.

[Agg15] Charu C Aggarwal. *Data mining: the textbook*. Springer, 2015.

[BALD14] O Babatunde, Leisa Armstrong, Jinsong Leng, and Dean Diepeveen. A genetic algorithm-based feature selection. *International Journal of Electronics Communication and Computer Engineering*, 4:889–905, 2014.

[BP02] Christoph Bandt and Bernd Pompe. Permutation entropy: a natural complexity measure for time series. *Physical review letters*, 88(17):174102, 2002.

[BPZL12] Gavin Brown, Adam Craig Pocock, Ming-Jie Zhao, and Mikel Luján. Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *Journal of Machine Learning Research*, 13:27–66, 2012.

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[Bus98] Massimo Buscema. Metanet*: The theory of independent judges. *Substance use & misuse*, 33(2):439–461, 1998.

[BYRN99] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern information retrieval*, volume 463. ACM press New York, 1999.

[Cam97] Rui Camacho. Delta ailerons dataset of a f16 aircraft. `http://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html`, 1997. Accessed: 2015-12-25.

[CKH+15] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. The ucr time series classification archive. `www.cs.ucr.edu/~eamonn/time_series_data/`, 2015. Accessed on: 2017-08-15.

[CL06] Yi-Wei Chen and Chih-Jen Lin. Combining svms with various feature selection strategies. In *Feature extraction*, pages 315–324. Springer, 2006.

[CL15] Long-Sheng Chen and Chun-Cheng Liu. Using feature selection approaches to identify crucial factors of mobile advertisements. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1, 2015.

[CL18] Muhammad Chaudhry and Jee-Hyong Lee. Motifs: Monte carlo tree search based feature selection. *Entropy*, 20(5):385, 2018.

[CM14] Verónica Bolón Canedo and Noelia Sánchez Marono. *Novel feature selection methods for high dimensional data*. PhD thesis, Universidade da Coruña, 2014.

[CS14] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[CT12] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[CTG+04] Yinhe Cao, Wen-wen Tung, JB Gao, Vladimir A Protopopescu, and Lee M Hively. Detecting dynamical changes in time series using the permutation entropy. *Physical review E*, 70(4):046217, 2004.

[DJLLP94] Thomas G Dietterich, Ajay N Jain, Richard H Lathrop, and Tomas Lozano-Perez. A comparison of dynamic reposing and tangent distance for drug activity prediction. In *Advances in Neural Information Processing Systems*, pages 216–223, 1994.

[DP05] Chris Ding and Hanchuan Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02):185–205, 2005.

[DSL15] Anthony Mihirana De Silva and Philip HW Leong. *Grammar-based feature generation for time-series prediction*. Springer, 2015.

[DV11] Gauthier Doquire and Michel Verleysen. An hybrid approach to feature selection for mixed categorical and continuous data. In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, pages 394–401, 2011.

[FC90] Mark A Fanty and Ronald A Cole. Spoken letter recognition. In *Advances in Neural Information Processing Systems*, pages 220–226, 1990.

[FH19] Mohsen Ahmadi Fahandar and Eyke Hüllermeier. Feature selection for analogy-based learning to rank. In *International Conference on Discovery Science*, pages 279–289. Springer, 2019.

[FJ14] Ben D Fulcher and Nick S Jones. Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):3026–3037, 2014.

[FKZ15] Tai Fei, David Kraus, and Abdelhak M Zoubir. Contributions to automatic target recognition systems for underwater mine classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 53(1):505–518, 2015.

[Fle04] François Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5:1531–1555, 2004.

[FM03] Dmitriy Fradkin and David Madigan. Experiments with random projections for machine learning. In *Proceedings of the 9th SIGKDD international conference on Knowledge discovery and data mining*, pages 517–522. ACM, 2003.

[FPSS96] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37, 1996.

[FSA99] Yoav Freund, Robert Schapire, and N Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[GE03] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[GE06] Isabelle Guyon and André Elisseeff. An introduction to feature extraction. In *Feature extraction*, pages 1–25. Springer, 2006.

[GGK+13] G Graff, B Graff, A Kaczkowska, D Makowiec, JM Amigó, J Piskorski, K Narkiewicz, and P Guzik. Ordinal pattern statistics for the assessment of heart rate variability. *The European Physical Journal Special Topics*, 222(2):525–534, 2013.

[GLF89] John H Gennari, Pat Langley, and Doug Fisher. Models of incremental concept formation. *Artificial intelligence*, 40(1-3):11–61, 1989.

[Gon12] Jacek Gondzio. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587–601, 2012.

[Hal99] Mark A Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[Hal00] Mark A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 359–366. Morgan Kaufmann Publishers Inc., 2000.

[HHL11] Hui-Huang Hsu, Cheng-Wei Hsieh, and Ming-Da Lu. Hybrid feature selection by combining filters and wrappers. *Expert Systems with Applications*, 38(7):8144–8150, 2011.

[HK11] Jan Hauke and Tomasz Kossowski. Comparison of values of pearson's and spearman's correlation coefficients on the same sets of data. *Quaestiones geographicae*, 30(2):87–93, 2011.

[HLY08] Qinghua Hu, Jinfu Liu, and Daren Yu. Mixed feature selection based on granulation and approximation. *Knowledge-Based Systems*, 21(4):294–304, 2008.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HS98] Mark A Hall and Lloyd A Smith. Practical feature subset selection for machine learning. In *Computer science proceedings of the 21st Australasian computer science conference ACSC*, volume 98, pages 181–191, 1998.

[HS16] Xiaoming Huo and Gábor J Székely. Fast computing for distance covariance. *Technometrics*, 58(4):435–447, 2016.

[HWC13] Myles Hollander, Douglas A Wolfe, and Eric Chicken. *Nonparametric statistical methods*, volume 751. John Wiley & Sons, 2013.

[HWZ+18] Kai Han, Yunhe Wang, Chao Zhang, Chao Li, and Chao Xu. Autoencoder inspired unsupervised feature selection. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2941–2945. IEEE, 2018.

[JBB15] Alan Jović, Karla Brkić, and Nikola Bogunović. A review of feature selection methods with applications. In *38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1200–1205. IEEE, 2015.

[JGDE08] Andreas Janecek, Wilfried Gansterer, Michael Demel, and Gerhard Ecker. On the relationship between feature selection and classification accuracy. In *New Challenges for Feature Selection in Data Mining and Knowledge Discovery*, pages 90–105, 2008.

[Jol02] I.T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2002.

[Jos19] Andreas Joseph. Shapley regressions: A framework for statistical inference on machine learning models. *arXiv preprint arXiv:1903.04209*, 2019.

[JS02] Richard Jensen and Qiang Shen. Fuzzy-rough sets for descriptive dimensionality reduction. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, volume 1, pages 29–34. IEEE, 2002.

[Kat16] Rohit J Kate. Using dynamic time warping distances as features for improved time series classification. *Data Mining and Knowledge Discovery*, 30(2):283–312, 2016.

[KC02] Nojun Kwak and Chong-Ho Choi. Input feature selection by mutual information based on parzen window. *IEEE transactions on pattern analysis and machine intelligence*, 24(12):1667–1671, 2002.

[KDCGD13] François Kawala, Ahlame Douzal-Chouakria, Eric Gaussier, and Eustache Dimert. Prédictions d'activité dans les réseaux sociaux en ligne. In *4ième conférence sur les modèles et l'analyse des réseaux: Approches mathématiques et informatiques*, page 16, 2013.

[Kel15] Fabian Keller. *Attribute Relationship Analysis in Outlier Mining and Stream Processing*. PhD thesis, Karlsruhe Institute of Technology, 2015.

[KGG85] James M Keller, Michael R Gray, and James A Givens. A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, science, and cybernetics*, SMC-15(4):580–585, 1985.

[KJ97] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[KMB12] Fabian Keller, Emmanuel Müller, and Klemens Bohm. HiCS: high contrast subspaces for density-based outlier ranking. In *28th International Conference on Data Engineering*, pages 1037–1048. IEEE, 2012.

[Kon94] Igor Kononenko. Estimating attributes: analysis and extensions of relief. In *European conference on machine learning*, pages 171–182. Springer, 1994.

[KRT+17] Louis Kirsch, Niklas Riekenbrauck, Daniel Thevessen, Marcus Pappik, Axel Stebner, Julius Kunze, Alexander Meissner, Arvind Kumar Shekar, and Emmanuel Müller. Framework for exploring and understanding multivariate correlations. In *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 404–408. Springer, 2017.

[KS66] S. Karlin and W.J. Studden. *Tchebycheff systems: with applications in analysis and statistics*. Pure and applied mathematics. Interscience Publishers, 1966.

[KvD10] Erik Kole and Dick JC van Dijk. How to identify and predict bull and bear markets? *Paris December*, 2010.

[LC01] Stan Lipovetsky and Michael Conklin. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17:319 – 330, 10 2001.

[Lee09] Ming-Chi Lee. Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Systems with Applications*, 36(8):10896–10904, 2009.

[LG10] Miguel Lázaro Gredilla. *Sparse Gaussian processes for large-scale machine learning*. PhD thesis, Universidad Carlos III de Madrid, 2010.

[Lic13] M. Lichman. UCI machine learning repository. `http://archive.ics.uci.edu/ml`, 2013. Accessed: 2017-07-01.

[Lin91] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

[LKL12] Jessica Lin, Rohan Khade, and Yuan Li. Rotation-invariant similarity in time series using bag-of-patterns representation. *Journal of Intelligent Information Systems*, 39(2):287–315, 2012.

[LL17] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

[LLC15] Gurobi Optimization LLC. Gurobi optimizer reference manual. `http://www.gurobi.com`, 2015.

[LMD+12] Chen Lin, T Miller, D Dligach, RM Plenge, EW Karlson, and G Savova. Maximal information coefficient for feature selection for clinical document classification. In *ICML Workshop on Machine Learning for Clinical Data. Edingburgh, UK*, 2012.

[LOR07] Xiaoli Li, Gaoxian Ouyang, and Douglas A Richards. Predictability analysis of absence seizures with permutation entropy. *Epilepsy research*, 77(1):70–74, 2007.

[LP03] Michael Lewitt and Robi Polikar. An ensemble approach for data fusion with learn++. In *International Workshop on Multiple Classifier Systems*, pages 176–185. Springer, 2003.

[LSLZ09] Huawen Liu, Jigui Sun, Lei Liu, and Huijie Zhang. Feature selection with dynamic mutual information. *Pattern Recognition*, 42(7):1330–1339, 2009.

[LWIG06] Wentian Li, Mingyi Wang, Patricia Irigoyen, and Peter K Gregersen. Inferring causal relationships among intermediate phenotypes and biomarkers: a case study of rheumatoid arthritis. *Bioinformatics*, 22(12):1503–1507, 2006.

[MBN02] Luis Carlos Molina, Lluís Belanche, and Àngela Nebot. Feature selection algorithms: a survey and experimental evaluation. In *Proceedings of International Conference on Data Mining*, pages 306–313. IEEE, 2002.

[Mör03] Fabian Mörchen. Time series feature extraction for data mining using dwt and dft, 2003.

[NAM01] Alex Nanopoulos, Rob Alcock, and Yannis Manolopoulos. Feature-based classification of time-series data. *International Journal of Computer Research*, 10(3):49–61, 2001.

[Net11] Pascal Network. Stock closing prices for 156 companies and 3 indexes from 2000 to 2007. `http://mldata.org/repository/data/viewslug/stockvalues/`, 2011. Accessed: 2016-02-09.

[NIP01] NIPS. Workshop on variable and feature selection. `http://www.clopinet.com/isabelle/Projects/NIPS2001/`, 2001. Accessed: 2016-07-19.

[NMV+13] Hoang Vu Nguyen, Emmanuel Müller, Jilles Vreeken, Fabian Keller, and Klemens Böhm. Cmi: An information-theoretic contrast measure for enhancing subspace cluster and outlier detection. In *13th SIAM International Conference on Data Mining (SDM), Austin, TX*, pages 198–206. SIAM, 2013.

[NMV+14] Hoang Vu Nguyen, Emmanuel Müller, Jilles Vreeken, Pavel Efros, and Klemens Böhm. Multivariate maximal correlation analysis. In *International Conference on Machine Learning*, volume 32, pages 775–783. JMLR.org, 2014.

[NPBT07] Roland Nilsson, José M Peña, Johan Björkegren, and Jesper Tegnér. Consistent feature selection for pattern recognition in polynomial time. *Journal of Machine Learning Research*, 8:589–612, 2007.

[Ols86] John E Olson. On the symmetric difference of two sets in a group. *European Journal of Combinatorics*, 7(1):43–54, 1986.

[ON14] Dijana Oreski and Tomislav Novosel. Comparison of feature selection techniques in knowledge discovery process. *Technology, Education, Management, Informatics Journal*, 3(4):285–290, 2014.

[OTN99] Nikunj C Oza, Kagan Tumer, and Peter Norwig. Dimensionality reduction through classifier ensembles. *Technical Report NASA-ARCIC-1999-124, Computational Sciences Division*, 1999.

[PHHN16] Karlson Pfannschmidt, Eyke Hüllermeier, Susanne Held, and Reto Neiger. Evaluating tests in medical diagnosis: combining machine learning with game-theoretical concepts. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 450–461. Springer, 2016.

[PLD05] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

[Pow07] Warren B Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*, volume 703. John Wiley & Sons, 2007.

[Qui14] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[Red96] T.C. Redman. *Data Quality for the Information Age*. Artech House Telecommunications Library. Artech House, 1996.

[RRF+11] David N Reshef, Yakir A Reshef, Hilary K Finucane, Sharon R Grossman, Gilean McVean, Peter J Turnbaugh, Eric S Lander, Michael Mitzenmacher, and Pardis C Sabeti. Detecting novel associations in large data sets. *science*, 334(6062):1518–1524, 2011.

[RSA+18] Pascal Reuss, Rotem Stram, Klaus-Dieter Althoff, Wolfram Henkel, and Frieder Henning. Knowledge engineering for decision support on diagnosis and maintenance in the aircraft domain. In *Synergies Between Knowledge Engineering and Software Engineering*, pages 173–196. Springer, 2018.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

[RŠK03] Marko Robnik-Šikonja and Igor Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine learning*, 53(1-2):23–69, 2003.

[Ryu93] Keunkwan Ryu. Monotonicity of the fisher information and the kullback-leibler divergence measure. *Economics Letters*, 42(2-3):121–128, 1993.

[Sai00] Naoki Saito. Local feature extraction and its applications using a library of bases. In *Topics in Analysis and Its Applications: Selected Theses*, pages 269–451. World Scientific, 2000.

[SAVdP08] Yvan Saeys, Thomas Abeel, and Yves Van de Peer. Robust feature selection using ensemble feature selection techniques. In *Machine learning and knowledge discovery in databases*, pages 313–325. Springer, 2008.

[SBS+17] Arvind Kumar Shekar, Tom Bocklisch, Patricia Iglesias Sánchez, Christoph Nikolas Straehle, and Emmanuel Müller. Including multi-feature interactions and redundancy for feature ranking in mixed datasets. In *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 239–255, 2017.

[SdSFS18] Arvind Kumar Shekar, Cláudio Rebelo de Sá, Hugo Ferreira, and Carlos Soares. Building robust prediction models for defective sensor data using artificial neural networks. *CoRR*, abs/1804.05544, 2018.

[SGK12] Mathieu Sinn, Ali Ghodsi, and Karsten Keller. Detecting change-points in time series by maximum mean discrepancy of ordinal pattern distributions. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 786–794, Arlington, Virginia, United States, 2012. AUAI Press.

[Shl14] Jonathon Shlens. A tutorial on principal component analysis. *CoRR*, abs/1404.1100, 2014.

[SISB11] RM Sharkawy, K Ibrahim, MMA Salama, and R Bartnikas. Particle swarm optimization feature selection for the classification of conducting particles in transformer oil. *IEEE Transactions on Dielectrics and Electrical Insulation*, 18(6):1897–1907, 2011.

[ŠK14] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014.

[ŠKŠ09] Erik Štrumbelj, Igor Kononenko, and M Robnik Šikonja. Explaining instance classifications with interactions of subsets of feature values. *Data & Knowledge Engineering*, 68(10):886–904, 2009.

[SM11] Benjamin Schowe and Katharina Morik. Fast-ensembles of minimum redundancy feature selection. In *Ensembles in Machine Learning Applications*, pages 75–95. Springer, 2011.

[SN19] Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. *arXiv preprint arXiv:1908.08474*, 2019.

[SP10] Sohan Seth and Jose C Principe. Variable selection: A statistical dependence perspective. In *Ninth international conference on Machine learning and applications*, pages 931–936. IEEE, 2010.

[SPISM18] Arvind Kumar Shekar, Marcus Pappik, Patricia Iglesias Sánchez, and Emmanuel Müller. Selection of relevant and non-redundant multivariate ordinal patterns for time series classification. In *Discovery Science*, pages 224–240. Springer International Publishing, 2018.

[Spr14] Vincent Spruyt. The curse of dimensionality in classification. `http://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/`, 2014. Accessed: 2018-03-27.

[SRB07] Gábor J Székely, Maria L Rizzo, and Nail K Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, 2007.

[SSM17] Arvind Kumar Shekar, Patricia Iglesias Sánchez, and Emmanuel Müller. Diverse selection of feature subsets for ensemble regression. In *Proceedings of 19th International Conference on Big Data Analytics and Knowledge Discovery*, pages 259–273, 2017.

[SV97] Alex Smola and Vladimir Vapnik. Support vector regression machines. *Advances in neural information processing systems*, 9:155–161, 1997.

[SWHB89] Vincent G Sigillito, Simon P Wing, Larrie V Hutton, and Kile B Baker. Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, 10(3):262–266, 1989.

[SWM17] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017.

[TM07] Wenyin Tang and KZ Mao. Feature selection algorithm for mixed data with both nominal and continuous features. *Pattern Recognition Letters*, 28(5):563–571, 2007.

[TPKC10] Sergios Theodoridis, Aggelos Pikrakis, Konstantinos Koutroumbas, and Dionisis Cavouras. *Introduction to Pattern Recognition: A Matlab Approach*. Academic Press, 2010.

[TSKK13] P.N. Tan, M. Steinbach, A. Karpatne, and V. Kumar. *Introduction to Data Mining*. Pearson, 2013.

[UMC+18] Ryan J. Urbanowicz, Melissa Meeker, William La Cava, Randal S. Olson, and Jason H. Moore. Relief-based feature selection: Introduction and review. *Journal of Biomedical Informatics*, 85:189–203, 2018.

[WJW+15] Yu Wei, Libin Jiao, Shenling Wang, Yinfeng Chen, and Dalian Liu. Time series classification with max-correlation and min-redundancy shapelets transformation. In *International Conference on Identification, Information, and Knowledge in the Internet of Things*, pages 7–12. IEEE, 2015.

[WMZ07] Gary M Weiss, Kate McCarthy, and Bibi Zabar. Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? In *Proceedings of International Conference on Data Mining*, pages 35–41. CSREA Press, 2007.

[WSH06] Xiaozhe Wang, Kate Smith, and Rob Hyndman. Characteristic-based clustering for time series data. *Data mining and knowledge Discovery*, 13(3):335–364, 2006.

[WWW07] Xiaozhe Wang, Anthony Wirth, and Liang Wang. Structure-based statistical features and multivariate time series clustering. In *Proceedings of the Seventh International Conference on Data Mining*, pages 351–360. IEEE, 2007.

[XHHZ10] Weichao Xu, Yunhe Hou, Y. S. Hung, and Yuexian Zou. Comparison of spearman's rho and kendall's tau in normal and contaminated normal models. *CoRR*, abs/1011.2009, 2010.

[XKWMN07] Xiaopeng Xi, Eamonn Keogh, Li Wei, and Agenor Mafra-Neto. Finding motifs in a database of shapes. In *Proceedings of the SIAM International conference on data mining*, pages 249–260. SIAM, 2007.

[YK09] Lexiang Ye and Eamonn Keogh. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th SIGKDD international conference on Knowledge discovery and data mining*, pages 947–956. ACM, 2009.

[YL03] Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of International Conference of Machine Learning*, pages 856–863. AAAI Press, 2003.

[YM00] Howard Hua Yang and John Moody. Data visualization and feature selection: New algorithms for nongaussian data. In *Advances in Neural Information Processing Systems*, pages 687–693. The MIT Press, 2000.

[YYS05] Hyunjin Yoon, Kiyoung Yang, and Cyrus Shahabi. Feature subset selection and feature ranking for multivariate time series. *IEEE Transactions on Knowledge and Data Engineering*, 17(9):1186–1198, 2005.

[ZMRBRP14] Fransisco Zamora-Martínez, Pablo Romeu, Pablo Botella-Rocamora, and Juan Pardo. On-line learning of indoor temperature

forecasting models towards energy efficiency. *Energy and Buildings*, 83:162–172, 2014.

[ZZ18] Quan-shi Zhang and Song-Chun Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.

[ZZX13] Yinghua Zhang, Wensheng Zhang, and Yuan Xie. Improved heuristic equivalent search algorithm based on maximal information coefficient for bayesian network structure learning. *Neurocomputing*, 117:186–195, 2013.

Declaration

I hereby declare, that this Doctoral Thesis, entitled "Multivariate Correlation Analysis for Supervised Feture Selection in High-Dimensional Data", is original research work and was written independently using no other sources and aids than cited. I furthermore declare that whatever contributions of others are involved, this contributions are indicated and clearly acknowledged.

Bonn, September 2019

Arvind Kumar Shekar

Curriculum Vitae

Education

06.2007 - 05.2011	**Bachelors in Mechatronics Engineering,** Kumaraguru College of Technology, Coimbatore
10.2013 - 08.2015	**Master of Science in Electronics Engineering**, Hochschule Bremen
06.2015 - 05.2018	**Doctor of Philosophy** candidate at University of Bonn in cooperation with Bosch GmbH, Stuttgart

Work Experience

07.2011 - 08.2013	**Software Engineer** Robert Bosch Engineering and Business Solutions India Limited
08.2014 - 01.2015	**Software Intern** Etas GmbH
02.2015 - 07.2015	**Master Thesis**: Data Modelling for Exhaust Gas Prediction Robert Bosch GmbH
10.2015 - 10.2018	**Doctor of Philosophy Student** Robert Bosch GmbH
11.2018 - 07.2019	**Data Scientist for Air-mass sensors** Robert Bosch GmbH- Powertrain Solutions
08.2019 - Present	**Data Scientist for Urban automated driving** Robert Bosch GmbH- Chassis Systems Control

Publications

1. Arvind Kumar Shekar, Marcus pappik, Patricia Iglesias Sanchez, Emmanuel Mueller: Selection of relevant and non-redundant multivariate ordinal patterns for time series classification. DS 2018: 224-240

2. Arvind Kumar Shekar, Cláudio Rebelo de Sá, Hugo Ferreira, Carlos Soares: Building robust prediction models for defective sensor data using Artificial Neural Networks. CoRR abs/1804.05544 (2018)

3. Arvind Kumar Shekar, Patricia Iglesias Sánchez, Emmanuel Mueller: Diverse Selection of Feature Subsets for Ensemble Regression. DaWaK 2017: 259-273

4. Arvind Kumar Shekar, Tom Bocklisch, Patricia Iglesias Sánchez, Christoph Nikolas Straehle, Emmanuel Mueller: Including Multi-feature Interactions and Redundancy for Feature Ranking in Mixed Datasets. ECML/PKDD (1) 2017: 239-255

5. Louis Kirsch, Niklas Riekenbrauck, Daniel Thevessen, Marcus Pappik, Axel Stebner, Julius Kunze, Alexander Meissner, Arvind Kumar Shekar, Emmanuel Mueller: Framework for Exploring and Understanding Multivariate Correlations. ECML/PKDD (3) 2017: 404-408

Books

Contributed for *Chapter 5*: *Providing Proactiveness*: *Data Analysis Techniques Portfolios*, in The MANTIS Book-Cyber Physical System Based Proactive Collaborative Maintenance 2019. Published by the River Publisher Series in Automation, Control and Robotics