Including Multi-feature Interactions and Redundancy for Feature Ranking in Mixed Datasets

Arvind Kumar Shekar[1](✉), Tom Bocklisch[2], Patricia Iglesias Sánchez[1], Christoph Nikolas Straehle[1], and Emmanuel Müller[2]

[1] Robert Bosch GmbH, Stuttgart, Germany
{arvindkumar.shekar,patricia.iglesiassanchez, Christoph-Nikolas.Straehle}@de.bosch.com
[2] Hasso Plattner Institute, Potsdam, Germany
{tom.bocklisch,emmanuel.mueller}@hpi.de

Abstract. Feature ranking is beneficial to gain knowledge and to identify the relevant features from a high-dimensional dataset. However, in several datasets, few features by itself might have small correlation with the target classes, but by combining these features with some other features, they can be strongly correlated with the target. This means that multiple features exhibit interactions among themselves. It is necessary to rank the features based on these interactions for better analysis and classifier performance. However, evaluating these interactions on large datasets is computationally challenging. Furthermore, datasets often have features with redundant information. Using such redundant features hinders both efficiency and generalization capability of the classifier. The major challenge is to efficiently rank the features based on relevance and redundance on mixed datasets. In this work, we propose a filter-based framework based on **R**elevance **a**nd **R**edundancy (RaR), RaR computes a single score that quantifies the feature relevance by considering interactions between features and redundancy. The top ranked features of RaR are characterized by maximum relevance and non-redundance. The evaluation on synthetic and real world datasets demonstrates that our approach outperforms several state-of-the-art feature selection techniques. Code and data related to this chapter are available at: https://doi.org/10.6084/m9.figshare.5418706.

1 Introduction

In high-dimensional feature spaces, feature ranking is an essential step for feature analysis and elimination of irrelevant features. Such irrelevant features affect the prediction and performance of classifiers [14]. 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,

M. Ceci et al. (Eds.): ECML PKDD 2017, Part I, LNAI 10534, pp. 239–255, 2017.
https://doi.org/10.1007/978-3-319-71249-9_15

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 [23]. All aforementioned problems are motivated with examples from our application, but they exist in several other domains such as Bio-informatics [7] and Media [4].

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. Nevertheless, existing filter-based feature selection methods [14,15,23,27] do not focus on considering all three challenges together: relevance based on multi-feature interactions, redundance and mixed data.

In this work, 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:

(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.

2 Related Work

Feature selection is an extensively researched topic and can be broadly classified into filter, wrapper, hybrid, embedded and unsupervised approaches [14,16,24,25]. We compare the related work based on the four properties summarized in Table 1.

Wrapper approaches with sequential forward selection (SFS) can handle redundancy, but it is not capable of evaluating feature interactions. Using recursive elimination addresses the problem of multi-feature interactions [28]. However, the major problem of this paradigm is efficiency, as the selection always depends on training the classifier several times.

To overcome this computational challenge, hybrid approaches were introduced. A well-known hybrid approach, Mixed Feature Selection (MFS) [27] is based on the decomposition of continuous feature space along the states of each categorical feature. A hybrid approach presented in [8], addresses the problem of inefficiency by building fewer classifier models. Hybrid paradigms are still inefficient on high-dimensional datasets, as it involves training of classifier multiple times. Hence, this work focuses on the filter-based paradigm which does not require training of a classifier multiple times.

Correlation-based Filter Selection (CFS) is an advanced version of Pearson's correlation, that is capable of handling redundance among features [15]. Similarly, the correlation measure mRmR [23] ranks the features based on relevance and redundancy. Tree-based embedded techniques are also well-known techniques for handling mixed data and redundancy [24]. However, CFS, mRmR and embedded techniques do not address interactions amidst features.

Unlike the aforementioned methods, unsupervised subspace search techniques [16,20] consider multi-feature interactions. However, these approaches focus on providing a score for the entire subspace. In contrast to this, 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. CMIM [10] and JMI [2] take relevance and redundancy for feature evaluation. However, CMIM is limited to boolean features and both have limitations for computing feature interactions between more than two features.

We propose a feature ranking framework RaR. RaR is an efficient filter-based feature ranking framework for evaluating relevance based on multi-feature interactions and redundancy on mixed datasets.

Table 1. Overview of the related work on feature selection

Paradigm	Approach	Mixed data	Redundancy	Feature interactions	Efficiency
Wrapper	SFS [28]	✓	✓	✗	✗
	Recursive elimination [28]	✓	✓	✓	✗
Hybrid	MFS [27]	✓	✗	✗	✗
	Doquire [8]	✓	✓	✗	✗
Subspace Ranking	HiCs [16]	✗	✗	✓	✓
Embeddedd	C4.5 [24]	✓	✓	✗	✓
Filter	mRmR [23]	✓	✓	✗	✓
	CFS [15]	✓	✓	✗	✓
	RaR	✓	✓	✓	✓

3 Problem Overview

In this section, we define the problem that we aim to solve. Let $\mathcal{F}$ be a d-dimensional mixed dataset $f_j \in \mathcal{F} \mid j = 1, \cdots, d$ with N instances. As a supervised learning process, the target $\boldsymbol{Y}$ is a collection of discrete classes. The mixed feature space $\mathcal{F}$ is defined by a set $\boldsymbol{X} \subseteq \mathcal{F}$ of continuous and set $\boldsymbol{Z} \subseteq \mathcal{F}$ of categorical features, i.e., $\mathcal{F} = \boldsymbol{X} \cup \boldsymbol{Z}$. In the following, we denote $error(\boldsymbol{S})$ as the error function of the classifier, trained using a subset of features $\boldsymbol{S} \subseteq \mathcal{F}$. For the given mixed dataset, we aim to (1) compute feature relevance by including their interactions with other features, as well as (2) evaluate the redundance score of each feature.

Evaluation of feature interactions requires a multivariate correlation measure, that quantifies the relevance of $\boldsymbol{S}$ to $\boldsymbol{Y}$. Given such a subspace relevance score $rel : \boldsymbol{S} \mapsto \mathbb{R}$, rel is a function of individual feature relevancies, i.e., $rel(\boldsymbol{S}) = \phi\big(\{r(f_j) \mid \forall f_j \in \boldsymbol{S}\}\big)$, where ϕ is an unknown function such that $\phi : \mathbb{R}^{|\boldsymbol{S}|} \mapsto \mathbb{R}$. 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., $red : (f_j, \boldsymbol{S}) \mapsto \mathbb{R}$. Given a feature $f_i \in \boldsymbol{S}$, non-redundant to $\boldsymbol{S}$ and $f_j \in \boldsymbol{S} \mid i \neq j$, with redundant information to $\boldsymbol{S}$, we intend to quantify a redundance score such that, $red(f_j, \boldsymbol{S}) > red(f_i, \boldsymbol{S})$. Addition of redundant feature information to a classifier does not contribute to the prediction quality, i.e., $error(\boldsymbol{S}) \approx error(\boldsymbol{S} \setminus f_j)$ [23]. 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 Relevance and Redundancy Ranking (RaR)

RaR consists of three major steps, computing the feature relevance by including feature interactions, redundancy and finally combining the two scores. To evaluate the feature relevance in a mixed dataset by including the feature interactions, we begin by computing the relevance (to the target class) scores of multiple subsets. We aim to infer the feature relevance based on their contribution to various subspace scores. Thus, the relevance of a feature to the target is decided based on its interaction with other features in multiple subspaces. This requires a multivariate correlation measure that can quantify the relevance of a subspace to the target. Hence, we begin with the introduction of a subspace correlation measure that we employ. This section is followed by the introduction of our heuristic to estimate the feature relevance based on multi-feature interactions. Finally, we elaborate our redundancy estimation and unification of the two scores.

4.1 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 [16, 21].

Definition 1. *Subspace Relevance.*
Given a subspace $\boldsymbol{S} \subseteq \mathcal{F}$ *and a divergence function* D*, the subspace relevance score* $rel(\boldsymbol{S})$ *to the target* $\boldsymbol{Y}$ *is defined as:*

$$rel(\boldsymbol{S}) = D\Big(p(\boldsymbol{Y} \mid \boldsymbol{S}) \mid\mid p(\boldsymbol{Y})\Big).$$

For a set of discrete target classes $\boldsymbol{Y}$, the marginal of the target is compared to its conditional distribution. This definition enables the measuring of multivariate and non-linear correlations [16] in mixed datasets. For $f_j \in \boldsymbol{X}$, the conditional is estimated based on a slice of continuous instances drawn from f_j. Similarly, for a $f_j \in \boldsymbol{Z}$, 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 [19]. However, the instantiation of divergence function D will be done in the Sect. 4.5.

4.2 Decomposition for Feature Relevance Estimation

A simple solution to estimate the relevance of f_j using Definition 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 $rel(\boldsymbol{S})$. 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 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 $(rel(\boldsymbol{S}_1) \mid \boldsymbol{S}_1 = \{f_1, f_2, f_3\})$ is greater than the relevance of a subset $(rel(\boldsymbol{S}_2) \mid \boldsymbol{S}_2 = \{f_1, f_2, f_4\})$ with an incomplete interactions.

A naïve decomposition is to decompose $rel(\boldsymbol{S})$ as the sum of individual feature relevancies. On applying naïve decomposition to our Example 1, we obtain $rel(\boldsymbol{S}_1) = r(f_1) + r(f_2) + r(f_3)$ and $rel(\boldsymbol{S}_2) = r(f_1) + r(f_2) + r(f_4)$. With an incomplete interaction structure, $rel(\boldsymbol{S}_2)$ 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 $\boldsymbol{Y}$. 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 2. *Feature Constraint.*
Let $r(f_j) \in \mathbb{R}$ be the relevance of individual features within the subspace $\boldsymbol{S} \subseteq \mathcal{F}$, we define the feature constraint as:

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

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., $\boldsymbol{S} \mid \boldsymbol{S} \in 2^{\mathcal{F}}$ and $f_j \in \boldsymbol{S}$. 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 2) for all possible feature combinations. We address this challenge by running M Monte Carlo iterations. For each iteration, we select a subspace $\boldsymbol{S}$ and define a constraint based on the subspace relevance $rel(\boldsymbol{S})$ score and the features belonging to $\boldsymbol{S}$. 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 2 shows an illustrative example of how our idea of generating constraints works for a dataset (in Example 1) with multi-feature interactions.Our approach draws several random subspaces as shown in Table 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 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.

Table 2. Illustrative example of feature constraints for 3 Monte Carlo iterations

i	$\boldsymbol{S}$	$rel(\boldsymbol{S})$	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	$r(f_1) + r(f_2) + r(f_4) \geq 0.15$

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,

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

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 Eq. 1 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 1) for each iteration requires the estimation of conditional probability distributions. However, evaluating the empirical conditional probabilities for large $|\boldsymbol{S}|$ is inaccurate. We demonstrate this by empirical evaluation in Sect. 5.3. Hence, it is necessary to restrict the size of the subspace to a maximum of k. That is, each randomly drawn $\boldsymbol{S}_i \mid \boldsymbol{S}_i \subseteq \mathcal{F}$ and $|\boldsymbol{S}_i| \leq k$. Algorithm 1 shows the pseudo-code for feature relevance estimation.

Algorithm 1. Estimation of Feature Relevance

```
Input: $\mathcal{F}$, $\boldsymbol{Y}, M, k$
$C = \emptyset$
for $i = 1 \rightarrow M$ do
    Sample $\{\boldsymbol{S}_i \mid \boldsymbol{S}_i \subseteq \mathcal{F} \wedge |S_i| \leq k\}$
    Compute $rel(\boldsymbol{S}_i)$ using Definition 1
    Construct constraint (cf. Definition 2)
    Add constraint to set $C$
end for
Optimize objective function Eq. 1 subject to $C$
return $r(f) \mid \forall f \in \mathcal{F}$
```

4.3 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 Sect. 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 1 to evaluate redundancy.

Definition 3. *Feature Redundancy.*
Given a set of features $\boldsymbol{R} \subseteq \mathcal{F}$*, a feature* $f_j \mid (f_j \in \mathcal{F}$ *and* $f_j \notin \boldsymbol{R})$ *is non-redundant w.r.t* $\boldsymbol{R}$ *iff:*

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

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 $\boldsymbol{R}_n$ 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.

$$red(f_j, \boldsymbol{R}) \equiv D\Big(p(f_j \mid \boldsymbol{R}) \mid\mid p(f_j)\Big) \tag{2}$$

If f_j is independent of $\boldsymbol{R}$, the 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 $f \in \boldsymbol{R}$, the deviation between the distributions in Eq. 2 will be 0. We illustrate the steps with an example.

Example 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 2, the features are sorted based on relevance scores following the step 1, i.e., $\boldsymbol{R}_n = \{f_5, f_3, f_1, f_2, f_4\} \mid n = |\mathcal{F}|$. 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 $\boldsymbol{R}_1 = \{f_5\}$ is estimated by applying Eq. 2. 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 2.

Algorithm 2. Estimation of Redundancy

```
Input: $\mathcal{F}$, $\boldsymbol{Y}$
$\boldsymbol{R}_n$=Sort $\forall f_j \in \mathcal{F}$ based on $r(f_j)$ from Algorithm 1
for $n = 2 \rightarrow |\mathcal{F}|$ do
    Compute $red(\boldsymbol{R}_n \triangle \boldsymbol{R}_{n-1}, \boldsymbol{R}_{n-1})$ c.f. Eq. 2  ▷ $\triangle$ dentotes symmetric difference
end for
return Calculate redundancy scores $red$ $\forall f_j \in \mathcal{F}$
```

For estimation of feature relevance, we restricted the subspace size to k (c.f. Sect. 4.2). This avoids inaccurate conditional probability estimates. Algorithm 2 also involves estimation of conditional probabilities. For a large $|\boldsymbol{R}_n|$, the conditional probability estimations using Eq. 2 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 $\forall s \subseteq \boldsymbol{R}_{n-1}$. From $\boldsymbol{R}_{n-1}$, 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.

4.4 RaR: Relevance and Redundancy Scoring

Having estimated the relevance and redundance of the features in Sects. 4.2 and 4.3, our final goal is to rank features based on a single score that combines both the properties.

Definition 4. *RaR score.*
Given the relevance $r(f_j)$ and redundancy score $red(f_j, \boldsymbol{R})$ of feature f_j, we define RaR(f_j) score as,

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

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

Example 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.

Time complexity analysis of RaR consists of three major phases: subspace sampling for constraint generation (Lines 2–7 of Algorithm 1), quadratic optimization (Line 8 of Algorithm 1) and redundancy estimation (Algorithm 2). 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 [16]. 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 1). 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 Eq. 1 subject to M constraints (Line 8).

A quadratic programming algorithm with M constraints and d-dimensional feature space has a time complexity $\mathcal{O}(M + \sqrt{d} \cdot ln\frac{1}{\epsilon})$ [12]. 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 Eq. 2. 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,

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

4.5 Instantiations for RaR

In Algorithm 1, a random subspace $\boldsymbol{S}$ is selected with maximum dimensionality k for each iteration. In order to estimate $rel(\boldsymbol{S})$, we compute the distribution of $\boldsymbol{Y}$ under some conditional slice of $\boldsymbol{S}$. That is, we aim to obtain a slice of $\boldsymbol{S}$ which satisfies a specific set of conditions, i.e., $D(p(\boldsymbol{Y} \mid \boldsymbol{S} \in [c_1, \cdots, c_{|\boldsymbol{S}|}]), p(\boldsymbol{Y}))$. Defining explicit conditions is a tedious task. Hence, we use adaptive subspace slicing, more details can be found in [16]. 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 Eq. 1 subject to these constraints.

RaR requires a divergence function to quantify the difference between distributions. We use KLD for our experiments. As KLD is formulated for both continuous and discrete probability distribution, it is directly applicable for redundancy estimation (c.f. Eq. 2) on mixed feature types. As a non-symmetric measure, we instantiate RaR with $KLD\big(p(\boldsymbol{Y} \mid \boldsymbol{S}) \mid\mid p(\boldsymbol{Y})\big)$ as it converges to mutual information and $KLD\big(p(\boldsymbol{Y}) \mid\mid p(\boldsymbol{Y} \mid \boldsymbol{S})\big)$ does not[1].

5 Experiments

5.1 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) [28] with K-Nearest Neighbors (KNN) [17], capable of handling redundant features. As hybrid technique, we consider the heuristic of Doquire [8]. The scheme requires a correlation measure and a classifier, hence we employ mRmR [23] and KNN with the heuristic of Doquire [8]. As filter approach, we test Maximal Information Coefficient (MIC) [18], mRmR [7,23], ReliefF [25] and

[1] https://hpi.de//mueller/rar.html.

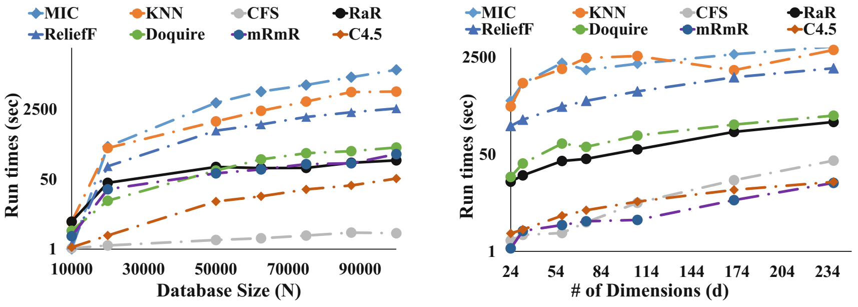
Correlation Filter Selection (CFS) [15]. Finally, we test the embedded scheme of decision trees (C4.5 [24]). We provide the implementation of RaR, competitor approaches, synthetic data generator[2] and the parameters[3] of our experiments. The results of our experiments on other classifiers are also made available. Additionally, we employ Gurobi [13] 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 [22] 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 [1].

For evaluation of our feature ranking framework, we also use 6 public datasets from the UCI repository with different dimensionalities and database sizes. The datasets contain both continuous and categorical features. The NIPS feature selection challenge [5] (2000 Instances/500 features), Ionosphere [26] (351 Instances/24 features), Musk2 [6] (6598 Instances/166 features), Isolet [9] (2000 Instances/500 features), Semeion [3] (1593 Instances/164 features) and Advertisement [11] (3279 Instances/1558 features) datasets. Experiments that had run times more that one day are denoted as ** in Tables 3 and 4.

5.2 Synthetic Data

We perform scalability analysis by evaluating the run times with increasing dimensionality and database size. Figure 1 shows the efficiency of RaR with



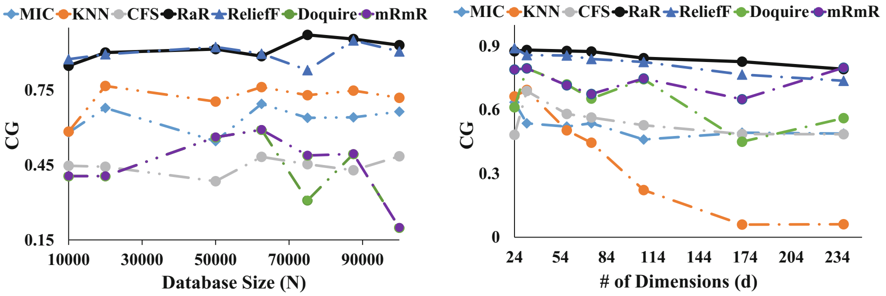
(a) Fixed dimensionality (50) and increasing database size

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

Fig. 1. Run time evaluation: run times of RaR vs. competitor approaches

[2] https://github.com/tmbo/rar-mfs.

[3] https://hpi.de//mueller/rar.html.



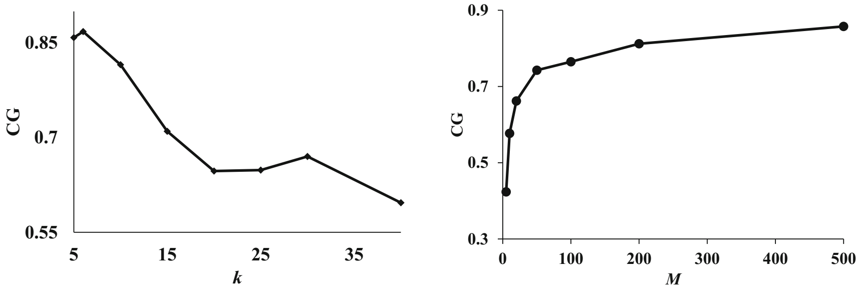
(a) Fixed dimensionality (50) and increasing database size (b) Fixed database size (20000) and increasing dimensionality

Fig. 2. Quality evaluation: CG of RaR vs. competitor techniques

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. Fig. 2). In comparison to ReliefF, which ranks features based on multi-feature interactions, RaR has lower run times and better feature ranking quality.

5.3 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. Fig. 3(a)) on synthetic data, we observe that the CG decreases with increasing k. The size of the conditional slices is determined by the α parameter [16]. 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 $\alpha^{\frac{1}{k}} \cdot N$ [16]. Hence, the conditional slice has



(a) Influence of k on feature ranking (M=100) (b) Influence of M on feature ranking (k=5)

Fig. 3. Parameter study, on synthetic dataset of 50 features and 20000 instances

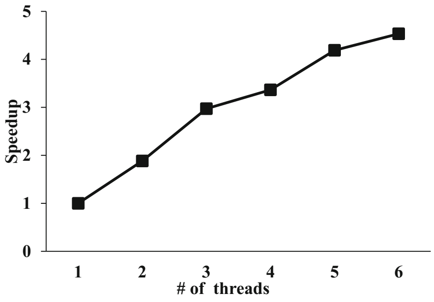


Fig. 4. Speedup of RaR

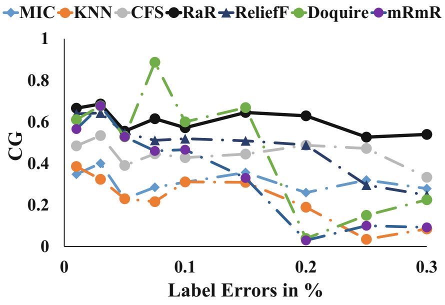


Fig. 5. Robustness of feature ranking

approximately 95% of all the instances. This leads to a very similar conditional and marginal distributions and distorted feature ranking. In Fig. 3(b), we vary 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 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.

5.4 Robustness w.r.t. Erroneous Labels

In several application scenarios, the target labels $\boldsymbol{Y}$ 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 [8] was able to perform well on a few cases (c.f. Fig. 5). 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.

5.5 Real World Datasets

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

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

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 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 3). Similar to our experiments on synthetic datasets (c.f. Figs. 1 and 2), 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.

5.6 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 [17]) in the order (best to worst) of their ranks. As shown in Fig. 6, 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,

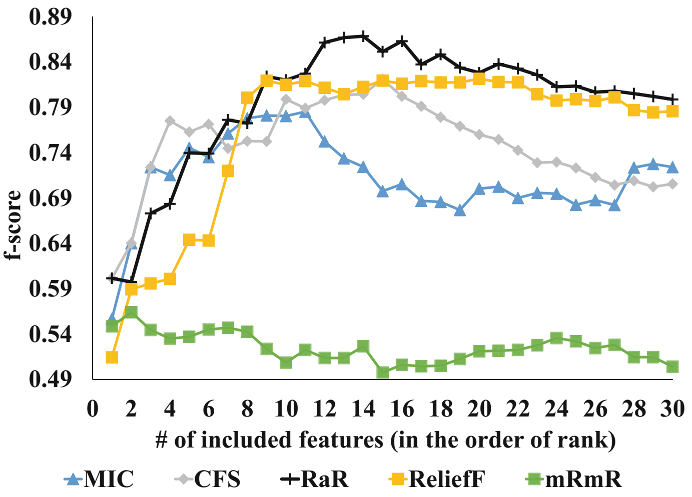


Fig. 6. f-scores of top 30 features on Isolet dataset

Table 5. Number of features required to obtain the quality in Table 3

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**

ReliefF has very similar prediction quality (c.f. Table 3) to RaR. By ranking the non-redundant features ahead, RaR achieves better f-score with fewer features (c.f. Fig. 6), 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 we show the number of features (c.f. Table 5) at which the maximum f-score (c.f. Table 3) was observed. Table 5 shows the number of top ranked features required to obtain the quality in Table 3, and RaR achieves the best f-score with fewer features.

6 Conclusions and Future Works

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.

As future works, we intend to address two directions to enhance our approach. In the event where two features are exactly identical to each other and are also maximally relevant, after penalization for redundancy, one of the feature can have a RaR score lower than the noisy features. This calls for a more sophistication in the combining of relevance and redundancy scores. RaR is based on the distribution of target class. Hence, RaR is currently limited to non-sparse datasets.

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval, vol. 463. ACM Press, New York (1999)
2. Brown, G., Pocock, A., Zhao, M.J., Luján, M.: Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. J. Mach. Learn. Res. **13**(Jan), 27–66 (2012)
3. Buscema, M.: Metanet*: the theory of independent judges. Subst. Use Misuse **33**(2), 439–461 (1998)
4. Chen, L.S., Liu, C.C.: Using feature selection approaches to identify crucial factors of mobile advertisements. In: Proceedings of the International MultiConference of Engineers and Computer Scientists, vol. 1 (2015)
5. Chen, Y.W., Lin, C.J.: Combining SVMs with various feature selection strategies. In: Guyon, I., Nikravesh, M., Gunn, S., Zadeh, L.A. (eds.) Feature Extraction: Foundations and Applications. Studies in Fuzziness and Soft Computing, vol. 207, pp. 315–324. Springer, Heidelberg (2006)
6. Dietterich, T.G., Jain, A.N., Lathrop, R.H., Lozano-Perez, T.: A comparison of dynamic reposing and tangent distance for drug activity prediction. In: Advances in Neural Information Processing Systems, p. 216 (1994)
7. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. J. Bioinform. Comput. Biol. **3**(02), 185–205 (2005)
8. Doquire, G., Verleysen, M.: An hybrid approach to feature selection for mixed categorical and continuous data. In: KDIR, pp. 394–401 (2011)
9. Fanty, M.A., Cole, R.A.: Spoken letter recognition. In: NIPS, pp. 220–226 (1990)
10. Fleuret, F.: Fast binary feature selection with conditional mutual information. J. Mach. Learn. Res. **5**(Nov), 1531–1555 (2004)
11. Fradkin, D., Madigan, D.: Experiments with random projections for machine learning. In: Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 517–522. ACM (2003)
12. Gondzio, J.: Interior point methods 25 years later. Eur. J. Oper. Res. **218**(3), 587–601 (2012)
13. Gurobi Optimization, Inc.: Gurobi optimizer reference manual (2015). http://www.gurobi.com
14. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**(Mar), 1157–1182 (2003)
15. Hall, M.A.: Correlation-based feature selection of discrete and numeric class machine learning (2000)

16. Keller, F., Müller, E., Bohm, K.: HiCS: high contrast subspaces for density-based outlier ranking. In: 2012 IEEE 28th International Conference on Data Engineering, pp. 1037–1048. IEEE (2012)
17. Keller, J.M., Gray, M.R., Givens, J.A.: A fuzzy K-nearest neighbor algorithm. IEEE Trans. Syst. Man Cybern. (4), 580–585 (1985)
18. Lin, C., Miller, T., Dligach, D., Plenge, R., Karlson, E., Savova, G.: Maximal information coefficient for feature selection for clinical document classification. In: ICML Workshop on Machine Learning for Clinical Data, Edingburgh, UK (2012)
19. Lin, J.: Divergence measures based on the Shannon entropy. IEEE Trans. Inf. Theory **37**(1), 145–151 (1991)
20. Nguyen, H.V., Müller, E., Vreeken, J., Keller, F., Böhm, K.: CMI: an information-theoretic contrast measure for enhancing subspace cluster and outlier detection. In: 13th SIAM International Conference on Data Mining (SDM), Austin, TX, pp. 198–206. SIAM (2013)
21. Nilsson, R., Peña, J.M., Björkegren, J., Tegnér, J.: Consistent feature selection for pattern recognition in polynomial time. J. Mach. Learn. Res. **8**(Mar), 589–612 (2007)
22. NIPS: Workshop on variable and feature selection (2001). http://www.clopinet.com/isabelle/Projects/NIPS2001/
23. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. Pattern Anal. Mach. Intell. **27**(8), 1226–1238 (2005)
24. Quinlan, J.R.: C4.5: Programs for Machine Learning. Elsevier, Amsterdam (2014)
25. Robnik-Šikonja, M., Kononenko, I.: Theoretical and empirical analysis of ReliefF and RReliefF. Mach. Learn. **53**(1–2), 23–69 (2003)
26. Sigillito, V.G., Wing, S.P., Hutton, L.V., Baker, K.B.: Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Tech. Dig. **10**(3), 262–266 (1989)
27. Tang, W., Mao, K.: Feature selection algorithm for mixed data with both nominal and continuous features. Pattern Recogn. Lett. **28**(5), 563–571 (2007)
28. Theodoridis, S., Pikrakis, A., Koutroumbas, K., Cavouras, D.: Introduction to Pattern Recognition: A Matlab Approach: A Matlab Approach. Academic Press, Cambridge (2010)