Filter-Based Information-Theoretic Feature Selection

Farinaz Pisheh
University of Houston
4800 Calhoun Rd.
Houston, TX, USA
zpisheh@uh.edu

Ricardo Vilalta
University of Houston
4800 Calhoun Rd.
Houston, TX, USA
rvilalta@uh.edu

ABSTRACT

Feature subset selection methods aim at identifying the smallest subset of features that maximize generalization performance, while preserving the true nature of the joint data distribution. In classification tasks, this is tantamount to finding an optimal subset of features relevant to the target class. A distinctive family of feature selection methods use a distance metric to identify relevant features, even under high feature interaction, by looking at the local class distribution. In this study we present EBFS: a new algorithm that is inspired by Relieff and uses an entropy-based metric to discover relevant features. Results on UCI data-sets show the effectiveness of our approach when compared to other filter-based feature selection methods.

CCS Concepts

•**Computing methodologies** → **Feature selection;**

Keywords

Feature selection; feature subset selection; supervised learning.

1. INTRODUCTION

Feature selection focuses on selecting only those features that will help to predict the target label with high generalization performance, e.g., with high accuracy. This task is essential in data-sets with a large number of features where often only a small subset of them is relevant. There are several approaches to select an optimal subset that does not include redundant and irrelevant features; in general, such approaches can be roughly divided into two categories: wrapper-based and filter-based [6]. The main difference between the two is that wrapper-based methods find an optimal set based on a specific classifier performance; as such, there is a strong dependency on the embedded classifier. Filter-based methods, on the other hand, evaluate features independently of any classifier; they rely on a metric of relevance that usually falls into one of several types, such as: information-theoretic measures, distance-based measures, or consistency measures [20].

In this work, we introduce a new filter-based feature selection algorithm named EBFS, that resembles Relieff [10], a well-known distance-based method. Like Relieff, our method uses the nearest neighbors of each instance to find the local weight of each feature. The sum of the weights of a feature in all the selected neighborhoods gives the final weight of that feature. Unlike Relieff, our method uses an information-theoretical measurement to select features. While many filter-based methods focus on the relation between the target class and one or more features (mutual information, conditional mutual information, or interaction information), our approach looks at the entropy of feature values in local neighborhoods, effectively capturing information not used in previous distance-based methods. Experimental results on real data-sets show the effectiveness of our approach to detect and identify relevant features.

The rest of this paper is organized as follows. In Section 2, we provide background information on information-theoretic measures commonly used in feature selection algorithms. In Section 3, we review related work on filter-based methods. In Section 4, we describe our methodology. In Section 5, we show experimental results on several synthetic and real-life data-sets, comparing the performance of EBFS with other methods. To finalize, Section 6 provides conclusions and future work.

2. BACKGROUND INFORMATION

Many feature selection algorithms integrate information-theory (IT) in their evaluation method. The fundamental measure in IT is the entropy [12] of a discrete variable X:

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i) \tag{1}$$

where X is a random variable, n is the number of unique values for X, and $P(x_i)$ is the probability of each possible value x_i. Entropy quantifies the amount of uncertainty in a random variable X (in bits). $H(X)$ is maximum when all $P(x_i)$ have the same value; in that case $H(X) = \log_2 n$. The term $\frac{H(X)}{\log_2 n} \in [0, 1]$ is referred to as *normalized entropy.*

Additional measures are defined next. Let Y be a random variable with m unique discrete values. The joint entropy [12, 2] of X and Y is defined as:

$$H(X,Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i, y_j) \log_2 P(x_i, y_j) \tag{2}$$

DOI:

where joint entropy is less than or equal to the sum of the individual entropy values of X and Y.

Conditional entropy[2] is defined using the definition of joint entropy:

$$H(Y|X) = H(X,Y) - H(X) \tag{3}$$

where the entropy of Y given X shows the decrease in the entropy of Y when knowing the entropy of X.

Mutual information[2] quantifies the amount of information shared by the two variables in terms of entropy and joint entropy:

$$\begin{aligned} MI(X,Y) &\equiv H(X) + H(Y) - H(X,Y) \\ &\equiv H(X) - H(X|Y) \equiv H(Y) - H(Y|X) \end{aligned} \tag{4}$$

Several forms of normalized MI have been proposed. One popular form in feature selection algorithm is as follows [2]:

$$U(X,Y) = 2\frac{MI(X,Y)}{H(X) + H(Y)} \tag{5}$$

Conditional mutual information [2] can be defined for three discrete random variables. Having three variables, X, Y and Z , where Z has q unique discrete values, the following shows the conditional mutual information of X and Y given Z:

$$\begin{gathered} MI(X;Y|Z) = \\ \sum_{k=1}^{q} P(z_k) \sum_{i=1}^{n} \sum_{j=1}^{m} P(x_i, y_j|z_k) \log_2 \frac{P(x_i,y_j|z_k)}{P(x_i|z_k)P(y_j|z_k)} \end{gathered} \tag{6}$$

Conditional MI can be obtained using entropy, joint entropy, and conditional entropy:

$$\begin{aligned} MI(X;Y|Z) &= H(X|Z) + H(Y,Z) - H(X|Y,Z) \\ &\equiv H(X,Z) + H(Y,Z) - H(X,Y,Z) - H(Z) \end{aligned} \tag{7}$$

Another important concept is that of interaction information (II) [8]; it shows the amount of information that is common in more than two random variables, but cannot be found in any subset of those variables:

$$\begin{aligned} II(X,Y,Z) &= MI(X,Y) - MI(X,Y|Z) \\ &= MI(X,Y|Z) - MI(X,Z) - MI(Y,Z) \end{aligned} \tag{8}$$

Unlike other metrics, interaction information can have negative values, indicating overlapping information between variables, a sign of feature redundancy. On the other hand, positive values can be interpreted as feature synergy. This is important because under feature interaction, the individual information of X or Y is not enough to know the value of Z. However, the synergetic interaction between them can reduce the uncertainty of Z to zero [8]. In this paper, we only consider positive interaction.

3. RELATED WORK

Mutual information is commonly used to measure the average dependency between a feature and the target class; features with high MI have a greater chance to be included in the final subset of features. A popular forward feature selection technique using MI is called MRMR [11]; to avoid redundancy, the algorithm penalizes high values of MI between two features. Another algorithm is called CMIM [4]; it looks for features corresponding to high $MI(X_i, Y|X_j)$ where X_j is a previously selected feature, Y is the target class, and X_i is a candidate feature; in this way, the algorithm discards features bearing high information about the target class, but that have similar information as previously selected features. Feature Selection Based on Joint Mutual Information (JMI) [17], evaluates the quality of a candidate feature X_i, by joining it to each of the previously selected features $\{X_j\}$ and measuring $MI(X_i, X_j|Y)$.

Previous work has emphasized the importance of maintaining a balance between feature redundancy and relevancy to achieve competitive results [1]; as an example, the algorithm JMI provides a trade-off between accuracy and stability for small data-sets. CFS [7] uses Pearson's correlation coefficient to find features bearing high correlation with the target class; to detect redundant features, the algorithm excludes features highly correlated to other features. FCBF [19] follows a similar approach by gathering a set of features exhibiting high symmetrical uncertainty with the class, removing redundant features within the selected features set.

None of the work above considers the feature interaction problem; each feature is simply assessed individually against the target class, while redundant features are discarded based on their correlation with other features. Unfortunately, interactive features might be removed in the process of discarding redundant ones. Employing the formula in equation 8 requires an exhaustive and computationally costly search for high-order interactive features.

3.1 Feature Interaction

Recent work advocates the search for interacting features jointly correlated with the target class. In [9], an interactive feature is called *weakly relevant* if is not highly correlated with the target class, but belongs to a subset of features that together are strongly correlated with the target class. In INTERACT [22], this definition is used to propose a criterion to remove features following a backward search technique. In IWFS [21], a metric called interaction weight factor based on symmetrical uncertainty and conditional mutual information is proposed; the algorithm captures the interaction of up to three features and the target class. CMICOT [13] detects the interaction between multiple features under a greedy search and a sequential forward selection approach; the algorithm uses a binary feature representation to estimate the CMIM measure in high-dimensional problems, and can identify interactions of up to t features and the target class (t is an input parameter). In [16], a four-dimensional joint mutual information measure is used to detect high order interactions. In general, identifying high-order interactive features by using an efficient algorithm remains an open problem.

3.2 Relieff

Relieff is a distance-based feature selection algorithm [10] that works as follows. It first selects m instances and identifies the neighborhood around each instance. Using a distance-based measure (e.g., Euclidean distance), Relieff finds k nearest instances that are from the same class of the selected instance (nearest hits), and the k nearest instances from each of the different classes (nearest misses). The weight of each feature f_j, $W[f_j]$, is then updated according to the following formula:

$$W[f_j] = W[f_j] - \sum_{l=1}^{k} \text{diff}(Z_i[f_j], \text{Hit}_l[f_j])/(m.k) + \sum_{C \neq \text{class}(Z_i)} [\tfrac{P(C)}{1-P(\text{class}(Z_i))} \sum_{l=1}^{k} \text{diff}(Z_i[f_j], \text{Miss}_l(C)[f_j])]/(m.k) \tag{9}$$

where $Z_i[f]$ is the value of feature f for the ith instance in the dataset, Miss_j and Hit_j are the jth nearest miss and the jth nearest hit respectively. $\text{diff}(A[f_j], B[f_j])$ is the distance between the values of f_j in A and B; for nominal features it is zero when the values are equal and one otherwise. For numerical attributes it is equal to $(A[f_j] - B[f_j])/\Gamma$ where Γ is the range of possible values $(\max - \min)$. Algorithm 1 shows pseudo code for the Relieff algorithm.

Algorithm 1: Relieff

Input : Data-set $D : \{Z_i | 0 \leq i \leq M\}$
with features: $F : \{f_j | 0 \leq j \leq n\}$ and
C: Classes.
k : Number of neighbors

Output: $W[f_j]$: Weight of each feature in the data-set

```
Set all W[f_j] = 0
for i=1 to m do
    Pick a random instance Z_i from D
    Find k nearest Hits: Hit_l
    foreach C ≠ class(Z_i) do
        Find k nearest Misses: Miss_l(C)
    end
    for j=1 to n do
        Update W[f_j] according to equation 9
    end
end
```

By looking at the local neighborhood of each instance, Relieff is able to update the weight of each feature. The update compares the value of a feature in the current instance with the value of the feature in each of the nearest miss and nearest hit neighbors. Features with values similar to to nearest hits, and farther away from the nearest misses will score higher. Jakulin [8] states that the common weaknesses of algorithms that use information theory to detect feature interaction does not exist in Relieff. By looking at the behavior of a feature in each neighborhood, Relieff is capable of identifying interactive features. One limitation of Relieff is its inability to remove feature duplicates. Several attempts have used the basic idea of Relieff to come up with more effective methods [5, 15, 14].

Our contribution is to introduce an algorithm inspired by Relieff that employs entropy to find relevant features. Experimental results on several data-sets of different sizes show that our algorithm is capable of identifying relevant features, including interactive features.

4. METHODOLOGY

While our method is inspired by the original Relieff, we have used entropy to create a new metric called *e_measure*. In this section we explain our new metric and our feature selection algorithm.

Relieff is an extension to the original Relief algorithm. One of the critical properties of the Relief family is evaluating the relevancy of a feature based on the different values it takes on different classes. At its core, Relief estimates the following to obtain the final weights:

$$\begin{aligned} W[f] \simeq\ & P(\text{diff. values of } f|\text{nearest misses}) \\ & -P(\text{diff. values of } f|\text{nearest hits}) \end{aligned} \tag{10}$$

With this interpretation of Relief, a feature is expected to attain a high weight if it leads to a set of different feature values among instances of different classes, while it leads to a set of similar feature values among instances of the same class. In other words. the formula penalizes features that separate instances of the same class. Now let Hits_i and Misses_i represent two sets that contain the nearest hits and nearest misses of Z_i respectively. Let $\text{val}(F_j, \text{Hits}_i)$ and $\text{val}(F_j, \text{Misses}_i)$ be the set of values for feature F_j in the jth feature of all nearest hits and nearest misses of instance Z_i respectively. We can re-formulate the weight update formula for a feature f_j with respect to instance Z_i as follows:

$$W[f_j, Z_i] \simeq H\left(\text{val}(f_j, \text{Misses}_i)\right) - H(\text{val}(f_j, \text{Hits}_i)) \tag{11}$$

This formula shows the difference in uncertainty of the values of f_j in nearest misses and nearest hits. The formula shows two entropy terms: the first one corresponds to the amount of "choices" we have for the value of f_j in the set of nearest misses. In the case of non-binary classification problems, at least two classes are considered different from the class of instance Z_i. In relevant features, we expect to have different values for each of the classes in the set of nearest hits; as a result we look for high entropy values where more choices of feature values lead to higher weight updates.

In the same way, the second term in equation 11 shows the amount of "choices" for f_j in the set of nearest hits. Here we look for low entropy values, meaning we expect most feature values to be almost the same. A value of zero means all nearest hits have the exact same value for f_j.

While the formula in equation 11 can quantify the uncertainty of feature values in the neighborhood of an instance Z_i, it fails to capture relevant information. Consider the following example. Assume a binary classification and neighborhoods of size $k = 4$. Assume the feature values for nearest misses is $\{1, 1, 1, 1\}$ and values for nearest hits is $\{0, 0, 0, 0\}$. Although the feature separates nearest hits from nearest misses perfectly, both terms in equation 11 yield the same entropy value. To overcome this, we add the value of f_j of the current instance Z_i, $Z_i[f_j]$, to both terms of equation 11. We call this measurement *e_measure*:

$$\begin{aligned} E[f_j, Z_i] = & H\big(\text{val}(f_j, \text{Misses}_i),\ Z_i[f_j]\big) \\ & -H\big(\text{val}(f_j, \text{Hits}_i),\ Z_i[f_j]\big) \end{aligned} \tag{12}$$

By adding this additional term, we expect to reduce the value of $\text{val}(f_j, \text{Hits}_i)$ in relevant features as we are adding a value that is anticipated to have higher probability of occurrence. On the other hand, adding it to $\text{val}(f_j, \text{Misses}_i)$ leads to an increase of uncertainty as it is anticipated that this value will have lower chance of occurrence. We used normalized entropy in the formula to keep the range of entropy values between $[0, 1]$. Algorithm 2 shows pseudo code for EBFS.

Algorithm 2: EBFS

Input : Data-set $D : \{Z_i | 0 \leq i \leq M\}$
with features: $F : \{f_j | 0 \leq j \leq n\}$ and
C : Classes.
k : Number of neighbors

Output: $W[f_j]$: Weight of each feature in D

Set all $W[f_j] = 0$
foreach Z_i *in* D **do**
    Hits = {}, Misses = {}
    Find k nearest Hits:
    Hits$_i$ ={Hit$_l$ | $1 \leq l \leq k$}
    foreach $C \neq class(Z_i)$ **do**
        Find k nearest Misses: Miss$_l(C)$
        Misses$_i$ =Misses$_i$ $\cup \{Miss_l(C) | 1 \leq l \leq k\}$
    end
    foreach f_j *in* F **do**
        Find values of f_j in Misses$_i$: val(f$_j$, Misses$_i$)
        Find values of f_j in Hits$_i$: val(f$_j$,Hits$_i$)
        $W[f_j] += e\_measure(Z_i, F_j)$ according to equation 12
    end
end

5. EXPERIMENTS AND RESULTS

To assess the performance of our method, we compare it with five state-of-the-the art feature selection algorithms: Relieff [10], MRMR [11], CMIM [4], IWFS [21], and JMI[18]. We implemented EBFS in Python 3.0. For Relieff, we used $k = 3$ and the built-in function from Matlab. IWFS was implemented in Matlab. The rest of the algorithms were provided by FEAST[1].

The data-sets employed in the experiments can be found at the UCI repository[2], except the Colon data-set[3]. We report on datasets having a variety of sample size, no. of features, and no. of classes. All the datasets were preprocessed to remove index values. On all the algorithms that use information theoretic measures, we discretized features with continuous features using the MDL method [3]. We used three classifiers in Scikit-learn[4]: Linear SVM, KNN with three neighbors, and Decision tree.

Similar to settings reported on recent publications [21, 13, 16], we planned our experiment as follows. We applied the feature selection algorithm on each of the datasets. Each algorithm was run n times on each dataset where $1 \leq n \leq$ min(50, #features in dataset). After the ith run, the top i feature(s) were selected and the rest were discarded. Then the dataset was fed to the three classifiers and run using ten-fold cross-validation. We report on average accuracy and standard deviation for the n runs. Results are shown in Table 2.

Each result was compared against EBFS for statistical significance using a two-tailed t-student test with $p = 0.05$. An (+) shows that EBFS is significantly better than the competitor, and an (-) shows the competitor is significantly better. For all the results with no signs, no significant difference were found. EBFS obtains the best results with decision tree classifier; no loss is seen against other feature selection algorithms in this classifier, and it achieved significantly better in fifteen cases.

[1]http://www.cs.man.ac.uk/ pococka4/FEAST/
[2]https://archive.ics.uci.edu/ml/datasets/
[3]http://genomics-pubs.princeton.edu/oncology/
[4]https://scikit-learn.org/

Table 1: Summary of the data-sets in our experiments

Data-set	#Features	#Samples	#Classes
Breast	9	699	2
CMC	9	1473	3
Wine	13	178	3
Zoo	16	101	7
Vehicle	18	846	4
Mushroom	22	8124	2
KR	36	3196	2
Lung	56	32	3
audio	69	200	24
Musk	166	476	2
SCADI	205	70	7
Colon	2000	62	2

In general, our method performed significantly better or equally well than other algorithms. In comparison to Relieff, EBFS lost in just one dataset with the KNN classifier. Looking at the three classifiers results, EBFS achieved the most wins against IWFS; it won six, seven and eight times with Decision trees, KNN and SVM respectively. In comparison to JMI and CMIM, EBFS performed mostly equal or better than these algorithms in all the classifiers. EBFS obtained similar results in comparison to MRMR when using decision trees and KNN. In SVM, however, EBFS lost three times and won twice; making MRMR our strongest competitor.

6. CONCLUSIONS AND FUTURE WORK

Feature subset selection is a crucial task during the preprocessing phase of almost any classification task. Finding (high-order) interactive features is an important challenge that needs further attention. In this paper, we introduce EBFS, an entropy-based feature selection algorithm that analyzes the neighborhood of a random instance (as originally proposed in the well-known Relieff algorithm) to estimate feature quality. Different from previous work, EBFS focuses on the entropy of feature values for nearest hits and misses. In comparison with five popular feature selection algorithms, our method shows competitive results, and is capable of finding relevant features, including interactive features. In the future, we plan to improve our proposed e_measure to relate more directly to generalization performance. In addition, we plan to investigate how to find neighborhoods in a selective manner, to increase the ability to identify interacting features.

7. REFERENCES

[1] G. Brown, A. Pocock, M.-J. Zhao, and M. Luján. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *Journal of machine learning research*, 13(Jan):27–66, 2012.

[2] T. Cover and J. Thomas. *Elements of information theory*. 2012.

[3] U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. 1993.

Table 2: Average accuracy with standard deviations.

Data-set	EBFS	Relieff	MRMR	CMIM	IWFS	JMI
C45						
Breast	92.22±0.03	92.03±0.03	92.44±0.01	92.39±0.01	92.58±0.01	92.61±0.01
CMC	49.38±0.04	46.79±0.02	47.65±0.01	46.94±0.02	48.44±0.01	48.12±0.01
Wine	93.45±0.04	88.48±0.09	94.21±0.04	94.58±0.04	90.26±0.05	94.42±0.04
Zoo	91.78±0.06	90.74±0.12	93.98±0.06	93.48±0.06	80.52±0.13(+)	93.92±0.06
Vehicle	66.35±0.07	65.55±0.09	66.22±0.05	66.6±0.05	63.4±0.07	64.34±0.06
Mushroom	96.4±0.01	91.69±0.11	96.5±0.01	96.6±0.0	96.34±0.01	96.56±0.0
Kr	92.35±0.05	93.68±0.06	93.49±0.06	93.29±0.06	90.53±0.06	93.87±0.06
Lung	54.85±0.1	52.03±0.09	53.32±0.09	50.72±0.09(+)	40.47±0.08(+)	54.23±0.11
Audio	72.11±0.11	70.35±0.13	73.93±0.06	74.12±0.06	41.13±0.05(+)	73.08±0.07
Musk	76.91±0.04	66.99±0.04(+)	64.61±0.03(+)	71.63±0.02(+)	66.25±0.04(+)	65.92±0.03(+)
SCADI	80.39±0.04	80.32±0.05	81.42±0.04	81.65±0.04	65.96±04(+)	81.61±0.04
Colon	88.7±0.03	71.68±0.04(+)	86.08±0.04(+)	85.3±0.04(+)	82.0±0.02(+)	86.19±0.04(+)
Win/Loss		2/1	2/0	3/0	6/0	2/0
KNN						
Breast	92.94±0.07	92.83±0.07	94.05±0.02	94.31±0.01	94.22±0.01	94.64±0.01
CMC	46.2±0.03	47.39±0.03	45.94±0.02	47.83±0.02	49.16±0.03(-)	49.11±0.03(-)
Wine	94.62±0.07	76.08±0.09(+)	96.26±0.06	96.74±0.06	90.15±0.08	96.3±0.06
Zoo	90.06±0.06	86.28±0.17	91.53±0.06	91.77±0.06	80.73±0.11(+)	91.42±0.06
Vehicle	64.32±0.09	64.42±0.09	60.55±0.04	63.41±0.04	60.28±0.06	61.49±0.04
Mushroom	94.77±0.01	88.93±0.12(+)	95.3±0.01(-)	94.59±0.01	94.52±0.01	94.9±0.01
Kr	82.33±0.09	84.93±0.07	83.46±0.08	83.24±0.07	74.17±0.09(+)	81.98±0.09
Lung	53.25±0.07	57.32±0.07(-)	55.72±0.11	50.2±0.07(+)	33.93±0.06(+)	55.33±0.09
Audio	57.02±0.08	58.37±0.11	53.76±0.04(+)	53.13±0.04(+)	40.71±0.06(+)	52.27±0.02(+)
Musk	75.26±0.05	76.39±0.06	66.3±0.04(+)	73.06±0.03(+)	68.11±0.03(+)	72.51±0.05(+)
SCADI	84.6±0.09	80.21±0.07(+)	84.85±0.08	83.38±0.07	35.02±0.05(+)	84.85±0.08
Colon	92.22±0.02	81.0±0.03(+)	94.9±0.02(-)	94.0±0.02(-)	84.57±0.06(+)	93.03±0.02
Win/Loss		4/1	2/2	3/1	7/1	2/1
SVM						
Breast	94.54±0.03	94.54±0.03	94.82±0.02	95.02±0.01	95.02±0.01	95.17±0.01
CMC	51.17±0.04	50.07±0.04	45.98±0.04(+)	50.29±0.03	50.29±0.03	50.3±0.03
Wine	95.66±0.06	91.93±0.07	96.59±0.05	96.49±0.05	90.08±0.07(+)	96.55±0.05
Zoo	90.2±0.07	90.03±0.12	90.92±0.08	92.74±0.06	81.17±0.12(+)	92.48±0.06
Vehicle	62.35±0.1	68.57±0.15	61.64±0.11	67.74±0.13	67.42±0.12	67.5±0.12
Mushroom	91.95±0.01	78.84±0.1(+)	85.9±0.09(+)	90.34±0.03(+)	88.44±0.02(+)	89.11±0.04(+)
Kr	91.75±0.05	91.16±0.05	90.28±0.05	91.96±0.05	89.84±0.05	92.06±0.05
Lung	50.9±0.07	53.4±0.08	54.35±0.08(-)	53.07±0.09	36.25±0.07(+)	53.2±0.1
Audio	73.31±0.12	68.63±0.13	72.69±0.05	72.75±0.06	41.7±0.07(+)	71.49±0.07
Musk	74.06±0.06	67.28±0.06(+)	73.35±0.03	64.81±0.04(+)	55.26±0.04(+)	65.04±0.05(+)
SCADI	81.33±0.05	79.21±0.04(+)	83.81±0.03(-)	82.89±0.03(-)	65.96±0.3(+)	83.41±0.03(-)
Colon	88.9±0.03	72.61±0.05(+)	94.52±0.02(-)	95.58±0.02(-)	84.57±0.1(+)	89.69±0.02
Win/Loss		4/0	2/3	2/2	8/0	2/1

[4] F. Fleuret. Fast Binary Feature Selection with Conditional Mutual Information. *Journal of Machine Learning Research*, 5:1531–1555, 2004.

[5] R. Gilad-Bachrach, A. Navot, and N. Tishby. Margin based feature selection - theory and algorithms. In *Proceedings of the 21st International Confer- ence on Machine Learning*, page 43, 2004.

[6] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[7] M. Hall. Correlation-based Feature Selection for Machine Learning. *Methodology*, 21i195-i20(April):1–5, 1999.

[8] A. Jakulin. Attribute interactions in machine learning. *Master's thesis University of Ljubljana, Slovenija*, (February), 2003.

[9] G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Machine Learning Proceedings 1994*, pages 121–129. Elsevier, 1994.

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

[11] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8):1226–1238, 2005.

[12] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(July 1928):379–423, 1948.

[13] A. Shishkin, A. Bezzubtseva, and A. Drutsa. Efficient High-Order Interaction-Aware Feature Selection Based on Conditional Mutual Information. *Nips*, (Nips):1–9, 2016.

[14] Y. Sun and J. Li. Iterative RELIEF for feature weighting. In *Proceedings of the 23rd international conference on Machine learning*, pages 913–920. ACM, 2006.

[15] Y. Sun, S. Todorovic, and S. Goodison. Local-learning-based feature selection for high-dimensional data analysis. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1610–1626, 2010.

[16] X. Tang, Y. Dai, Y. Xiang, and L. Luo. An Interaction-Enhanced Feature Selection Algorithm. pages 115–125. 2018.

[17] H. Yang and J. Moody. Feature selection based on joint mutual information. In *Proceedings of international ICSC symposium on advances in intelligent data analysis*, pages 22–25. Citeseer, 1999.

[18] H. Yang, J. M. P. o. i. I. s. on, and u. 1999. Feature selection based on joint mutual information. *Citeseer*.

[19] L. Yu and H. Liu. Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. *International Conference on Machine Learning (ICML)*, pages 1–8, 2003.

[20] L. Yu and H. Liu. Efficient Feature Selection via Analysis of Relevance and Redundancy. Technical report, 2004.

[21] Z. Zeng, H. Zhang, R. Zhang, and C. Yin. A novel feature selection method considering feature interaction. *Pattern Recognition*, 48(8):2656–2666, 2015.

[22] Z. Zhao and H. Liu. Searching for interacting features. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 1156–1161, 2007.