entropy

MDPI

Article

Quadratic Mutual Information Feature Selection

Davor Sluga * and Uroš Lotrič

University of Ljubljana, Faculty of Computer and Information Science, Ljubljana 1000, Slovenia; uros.lotric@fri.uni-lj.si

* Correspondence: davor.sluga@fri.uni-lj.si; Tel.: +386-1-479-8231

Academic Editor: Antonio M. Scarfone

Received: 13 December 2016; Accepted: 30 March 2017; Published: 1 April 2017

Abstract: We propose a novel feature selection method based on quadratic mutual information which has its roots in Cauchy–Schwarz divergence and Renyi entropy. The method uses the direct estimation of quadratic mutual information from data samples using Gaussian kernel functions, and can detect second order non-linear relations. Its main advantages are: (i) unified analysis of discrete and continuous data, excluding any discretization; and (ii) its parameter-free design. The effectiveness of the proposed method is demonstrated through an extensive comparison with mutual information feature selection (MIFS), minimum redundancy maximum relevance (MRMR), and joint mutual information (JMI) on classification and regression problem domains. The experiments show that proposed method performs comparably to the other methods when applied to classification problems, except it is considerably faster. In the case of regression, it compares favourably to the others, but is slower.

Keywords: feature selection; information-theoretic measures; quadratic mutual information; Cauchy–Schwarz divergence

1. Introduction

Modelling data using machine learning approaches usually involves taking some kind of learning machine (e.g., decision tree, neural network, support vector machine) to train a model using already known input and output data. For example, based on features collected about patients (gender, blood pressure, presence or absence of certain symptoms, etc.) and given the patients' diagnoses (the outputs), we can build a model and use it afterwards as a diagnosis tool for new patients. The input features and the output can be discrete (e.g., gender) or continuous (e.g., body temperature). In the first case we are dealing with a classification problem, and in the last with a regression problem.

Many classification or regression problems involve high-dimensional input data. For example, gene expression data can easily reach into tens of thousands of features [1]. The majority of these features are either irrelevant or redundant for the given classification or regression task. A large number of features can lead to poor inference performance, possible over-fitting of the model, and increased training time [2].

To tackle these problems, feature selection algorithms try to select a smaller feature subset which is highly relevant to the output. There exists a great number of approaches to feature selection. We can divide them into three main groups; specifically, wrapper, embedded, and filter. The wrapper approach [3] uses the performance of a learning machine to evaluate the relevance of feature subsets. They usually achieve good performance, but can be computationally costly and infeasible for use on large data sets. Moreover, their performance depends on the learning machine being used in the evaluation. The embedded approach integrates feature selection into the learning machine itself and performs selection implicitly during the training phase. This method is faster [1], but still dependent on the learning machine. Filters are faster than both of the previous approaches and use a simple

relevance criterion based on some measure such as correlation coefficient [4] or mutual information [5] to asses the goodness of a feature subset. The evaluation is independent of the learning machine and is less prone to over-fitting, but may fail to find the optimal feature subset for a given learning machine.

In addition to the relevance criterion, feature selection must also employ a certain search process, which drives the feature selection. Optimally, exhaustive search evaluates all possible feature subsets and selects the best one. This is usually computationally prohibitive, so greedy approaches like sequential search or random search [6] are used in practice.

In this work, we focus on the filter approach to feature selection and present a novel method based on quadratic mutual information. The method's criterion has its roots in Cauchy–Schwarz divergence and quadratic Renyi entropy. Our motivation is the straightforward estimation of Cauchy–Schwarz divergence [7] for discrete features, continuous features, or their combination, which makes it suitable to use without any preprocessing dependent on expert knowledge about the data. Moreover, it avoids the use of parameters, which are inconvenient for non-experts in the field. It is possible to use it as a precursor to classification and regression problems in order to avoid over-fitting and to improve the learning machine performance.

The paper is organized as follows. Section 2 briefly reviews previous work on feature selection using information-theoretic measures and their generalizations. Section 3 presents the proposed measure and search organization for the task of finding relevant features. Section 4 presents the experimental setting and the results obtained on classification and regression problems. Lastly, Section 5 gives conclusions and possible future research directions.

2. Related Work

There are many information-theoretic feature selection methods that have been proposed in the last two decades. Brown et al. [8] and Vergara et al. [2] unified most of them in the mutual information feature selection framework. There are also a few cases of using mutual information derived from Renyi [9] and Tsallis entropy [10] showing promising results. Chown and Huang [11] proposed the use of a data compression algorithm along with quadratic mutual information to perform feature selection, but their method is prone to over-fitting, due to the estimation of the criterion in high-dimensional space. Here we mention a few of the most well-known information-theoretic measures and criteria reviewed in [8], since they are all based on similar ideas.

2.1. Information-Theoretic Measures

Information-theoretic measures offer means to rank feature subsets according to the information they provide about the output. A finite set of features $\mathbf{X} = \{X_1, \ldots, X_N\}$, which can acquire values $\mathbf{x}_1, ..., \mathbf{x}_{m_1}$ with probabilities $p_1(\mathbf{x}_1), ..., p_1(\mathbf{x}_{m_1})$, has the Shannon entropy

$$H(\mathbf{X}) = -\sum_{i=1}^{m_1} p_1(\mathbf{x}_i) \log p_1(\mathbf{x}_i) \,. \tag{1}$$

Similarly, we can calculate the entropy of the output $H(Y)$ given the possible values $Y = \{y_1, \ldots, y_{m_2}\}$ with probabilities $p_2(y_1), \ldots, p_2(y_{m_2})$ and the joint entropy $H(\mathbf{X}, Y)$ given the joint probabilities $p_{12}(\mathbf{x}_i, y_j)$.

Another important information-theoretic measure is the Kullback–Leibler divergence (KL), which measures discrepancy between two probability distributions p and p'

$$D_{\mathrm{KL}}(p, p') = \sum_{i=1}^{m_1} p(\mathbf{x}_i) \log \frac{p(\mathbf{x}_i)}{p'(\mathbf{x}_i)} \,. \tag{2}$$

The Kulback–Leibler divergence between the joint probability distribution $p_{12}(\mathbf{x}, y)$ and the distribution $p_1(\mathbf{x})p_2(y)$ is mutual information (MI):

$$I(\mathbf{X}; Y) = D_{\mathrm{KL}}(p_{12}(\mathbf{x}, y), p_1(\mathbf{x})p_2(y)) \,. \tag{3}$$

We can usually estimate the probability distributions using one of the histogram-based methods. When the features are continuous, one option is to apply a discretization step beforehand (equal width binning, equal frequency binning) [12]. The manual selection of the number of bins can affect the estimation of MI and can lead to spurious results by shrouding some properties of the probability distribution. A better approach is to perform the discretization using an adaptive technique like minimum description length (MDL) [13], but this does not work for continuous output and thus cannot be used in a regression problem.

To avoid discretization, we can compute the differential mutual information directly from continuous data using the differential Kullback–Leibler divergence

$$D^d_{\mathrm{KL}}(p, p') = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{p'(\mathbf{x})} d\mathbf{x} \,, \tag{4}$$

$$I^d(\mathbf{X}; Y) = D^d_{\mathrm{KL}}(p_{12}(\mathbf{x}, y), p_1(\mathbf{x})p_2(y)) \,, \tag{5}$$

but we must estimate the probability density functions $p_1(\mathbf{x})$, $p_2(y)$, and $p_{12}(\mathbf{x}, y)$ beforehand.

The non-parametric Parzen-window method [14] is the most straightforward approach to density function estimation. The estimate is obtained by spanning kernel functions around the data samples, $p(\mathbf{x}) = \sum_{i=1}^{n} G(\mathbf{x} - \mathbf{x}_i, \mathbf{h})/n$. The most commonly used kernel function is the product of Gaussians $G(\mathbf{x}, \mathbf{h}) = \prod_{d=1}^{D} G(x_d, h_d)$, where D is the size of the feature set. The estimate depends on the choice of kernel width h_d, for which there are several recipes in the literature [15]. However, the numerical computation of differential MI for a set of features is computationally quite expensive and prone to error. Another approach to differential MI estimation is the k-nearest neighbors (kNN) estimator [16] of MI, which in certain situations provides better results than the Parzen-window, but is still computationally expensive and not suitable to use directly on data sets comprised of discrete and continuous data [17]. A more recent approach is to estimate the density ratio in (4) directly. However, due to the logarithm in (4), this approach becomes computationally expensive and susceptible to outliers [18]. To alleviate this problem, the authors [18] propose a squared-loss mutual information measure which makes the computation more robust.

Besides the classical Shannon entropy, there exists a range of information entropy generalizations [19]. One of the more widely known is the Renyi entropy [20]

$$H_{\mathrm{R}_q}(\mathbf{X}) = \frac{1}{1-q} \log \sum_{i=1}^{m_1} p_1(\mathbf{x}_i)^q \,, \tag{6}$$

which extends the original concept by introducing an additional parameter q. It should be noted that Renyi entropy converges to Shannon entropy as q approaches 1 in the limit. Renyi also defined the differential Renyi entropy, where the integral $\int p_1(\mathbf{x})^q d\mathbf{x}$ substitutes the sum $\sum_{i=1}^{m_1} p_1(\mathbf{x}_i)^q$ in (6).

Usually, the estimation of differential entropy includes a probability density function (PDF) estimation from the data followed by the integral estimation from the PDF, which is challenging in high-dimensional problems. Erdogmus et al. [21] showed that quadratic Renyi entropy ($q = 2$) can be directly estimated from the data, bypassing the explicit need to estimate the PDF. Namely, the information potential $V(\mathbf{X}) = \int p_1(\mathbf{x})^2 d\mathbf{x}$ can be estimated as

$$V(\mathbf{x}) = \frac{1}{n^2} \sum_{k=1}^{n} \sum_{j=1}^{n} G(\mathbf{x}_k - \mathbf{x}_j, \sqrt{2}\mathbf{h}) \,, \tag{7}$$

replacing the numerical integration of the PDF with sums over the data samples. The Renyi differential quadratic entropy estimator thus becomes

$$\hat{H}_{\mathrm{R}_2}(\mathbf{X}) = -\log V(\mathbf{x}) \,. \tag{8}$$

There exist many proposals on how to compute mutual information with regards to Renyi entropy, but each lacks some of the properties that the Shannon mutual information exhibits [22]. One of the proposed measures—the Cauchy–Schwarz divergence

$$D_{\mathrm{CS}}(p_1, p_2) = -\log \frac{(\int p_1(\mathbf{x}) p_2(\mathbf{x}) d\mathbf{x})^2}{\int p_1^2(\mathbf{x}) d\mathbf{x} \int p_2^2(\mathbf{x}) d\mathbf{x}} \tag{9}$$

by Principe et al. [7]—is especially suitable as a substitute for Kullback–Liebler divergence, as it enables assessment of the dependence between variables directly from the data samples. By rearranging the above equation, we obtain

$$D_{\mathrm{CS}}(p_1, p_2) = 2H_{\mathrm{R}_2}(\mathbf{X}; Y) - H_{\mathrm{R}_2}(\mathbf{X}) - H_{\mathrm{R}_2}(Y) \,, \tag{10}$$

where the first term $H_{\mathrm{R}_2}(\mathbf{X}; Y)$ is the quadratic Renyi cross-entropy [7], which can be directly estimated from the data using a similar approach as in the case of $\hat{H}_{\mathrm{R}_2}(\mathbf{X})$

$$\hat{H}_{\mathrm{R}_2}(\mathbf{X}; Y) = -\log \frac{1}{n^{D+2}} \sum_{i=1}^{n} \left[\left(\sum_{k=1}^{n} G(y_i - y_k, \sqrt{2}h) \right) \times \prod_{d=1}^{D} \left(\sum_{j=1}^{n} G(x_{di} - x_{dj}, \sqrt{2}h_d) \right) \right] . \tag{11}$$

On the basis of Cauchy–Schwarz divergence (10), Principe et al. [7] proposed quadratic mutual information (QMI)

$$I_{\mathrm{CS}}(\mathbf{X}; Y) = D_{\mathrm{CS}}(p_{12}(\mathbf{x}, y), p_1(\mathbf{x}) p_2(y)) \tag{12}$$

as a candidate for measuring dependence. They prove that $I_{\mathrm{CS}}(\mathbf{X}; Y) = 0$ if and only if $\mathbf{X}$ and Y are independent of each other and positive otherwise, similar to the Kullback–Liebler divergence.

2.2. Information-Theoretic Feature Selection Methods

Given a set of already-selected features $\mathbf{X}_S = \{X_1, .., X_M\}$ and a set of candidate features $\mathbf{X}_C = \{X_{M+1}, \ldots, X_N\}$, Battiti [23] proposed to compute a mutual information feature selection criterion (MIFS) for each candidate feature X_c

$$S_{\mathrm{MIFS}}(X_c) = I(X_c; Y) - \beta \sum_{s=1}^{M} I(X_c; X_s) \tag{13}$$

and add the feature with the maximum value to the set of already selected features. The criterion is a heuristic which takes into account first order relevancy $I(X_c; Y)$ and first order redundancy $I(X_c; X_s)$. It includes the parameter β, which greatly affects performance [24].

Peng et al. [25] improved on the MIFS idea and proposed the minimum redundancy maximum relevance criterion (MRMR), which uses MIFS with automatic setting of parameter β

$$S_{\mathrm{MRMR}}(X_c) = I(X_c; Y) - \frac{1}{M} \sum_{s=1}^{M} I(X_c; X_s) \,. \tag{14}$$

MRMR avoids using parameters, but still considers only first-order interactions.

Yang and Moody [26] used joint mutual information (JMI) as a criterion for feature selection

$$S_{\mathrm{JMI}}(X_c) = \sum_{s=1}^{M} I(X_c, X_s; Y) \,. \tag{15}$$

This criterion considers second-order interactions between features and the output, thus increasing computational costs on one hand, but on the other hand also allowing detection of features which, when taken in pairs, provide more information about the output than the sum of both features' individual contributions.

Several methods have been developed which go beyond second-order interactions [27–29]. The joint search for multiple features is difficult, as multidimensional probability distributions are hard to estimate, and becomes especially problematic when the number of samples is small. However, this is a favourable approach when questing for a small number of features, as some subtle interactions can be revealed. When using filter methods as a pre-processing stage for a machine learning task, it is usually better to select more features and give the learning machine more options to choose from and possibly find higher-order interactions during the learning phase [30].

These methods are usually used on discrete/discretized data for classification problems. Frenay et al. [31] examined the adequacy of MI for feature selection in regression tasks and argue that in most cases it is a suitable criterion. However, regardless of feature selection being a precursor to classification or regression task, most problems arise from the difficulty of estimating the MI.

3. The Proposed Method

The quadratic mutual information (12) works as the basis for our feature selection method, because it can be computed directly from the data samples and works for both discrete and continuous features. Optimally, the method should assess every possible subset of feature candidates and select the subset with maximum QMI. However, evaluating all possible subsets of features is prohibitively time-consuming. Another problem is that the estimation of I_{CS} is prone to over-fitting, especially if the number of samples is not much larger than the number of features in the subset. This is a common problem in machine learning when dealing with high-dimensional data. To cope with it, feature selection methods usually rank or select features iteratively one by one. Even if the features are added to the relevant set one by one, it is still important to consider possible interactions between them to prevent adding redundant features, or to include those that are not informative about the output on their own, but are useful when taken with other features.

The proposed method (Algorithm 1) selects features iteratively until it reaches an ending criterion—the number of features we want to have. At each step, the algorithm considers all possible candidates from the set of candidate features $\mathbf{X}_C$. It checks each candidate feature X_c against the set of already selected features $X_s \in \mathbf{X}_S$ from the previous steps

$$S_{\mathrm{QMIFS}}(X_c) = S_{\mathrm{QMIFS}}(X_c, \mathbf{X}_S, Y) = \begin{cases} I_{\mathrm{CS}}(X_c; Y) & \text{if } M = 0 \\ \sum_{s=1}^{M} \left(I_{\mathrm{CS}}(X_c, X_s; Y) - I_{\mathrm{CS}}(X_c; X_s) \right) & \text{if } M > 0 \end{cases} . \tag{16}$$

It adds the candidate feature X_c with maximum S_{QMIFS} to the set of already selected features. In the beginning, $\mathbf{X}_S$ is empty, so the algorithm considers only quadratic mutual information between candidates and output. For later steps, the criterion function (16) is composed of sums of pairs of terms. The first term rewards the candidate features that are the most informative about the output when taken along with an already selected feature. The second term penalizes the features that have a strong correlation with already selected features. On one hand, this ensures the detection of features which work better in pairs—they provide more information about the output when taken together than the sum of both features' individual contributions. On the other hand, it avoids selecting redundant features—the information they provide about the output is present in one of the already selected

features. Extension of the criterion to include higher-order interactions between features is possible, but considerably increases the computational time and is more prone to over-fitting.

Algorithm 1: Quadratic mutual information feature selection—QMIFS

Data: Set of candidate features $\mathbf{X}_C$ and output Y
Result: Set of selected features indices $\mathbf{S}$

Standardize $\mathbf{X}_C$ and Y
$\mathbf{X}_S \longleftarrow \emptyset$
$\mathbf{S} \longleftarrow \emptyset$
while *ending condition not met* **do**
  $S_{\max} = 0$
  for $X_c \in \mathbf{X}_C$ **do**
    $S_c \longleftarrow S_{\text{QMIFS}}(X_c, \mathbf{X}_S, Y)$
    if $S_c > S_{\max}$ **then**
      $S_{\max} \longleftarrow S_c$
      $X_{\max} \longleftarrow X_c$
      $c_{\max} \longleftarrow c$
    end
  end
  $\mathbf{X}_C \longleftarrow \mathbf{X}_C / X_{\max}$
  $\mathbf{X}_S \longleftarrow \mathbf{X}_S \cup X_{\max}$
  $\mathbf{S} \longleftarrow \mathbf{S} \cup c_{\max}$
end

There are a few considerations we must take into account before using this method to select the features. Firstly, the estimation of I_{CS} depends heavily on the kernel width $\mathbf{h}$ [7]. The Silverman rule [32] is a common way to estimate it, but the width (h_d) must be the same across all features. Neglecting this, the value of the criterion function will vary even if all candidate features are equally relevant to the output [7], and will fail to choose the correct ones. We take care of this problem by standardizing the data, which in turn causes the Silverman rule to produce the same h_d for every feature. Secondly, the magnitude of I_{CS} has no meaning [7] due to the dependence on the choice of window width. However, correct identification of the most relevant features requires only relative difference among them. That is, given two features X_a, X_b, and the output, and knowing that feature X_a is more informative about the output than X_b, the S_{QMIFS} estimate is acceptable when $S_{\text{QMIFS}}(X_a) > S_{\text{QMIFS}}(X_b)$. The following small-scale experiment nicely presents some of the important properties of the proposed criterion.

We generate correlated data composed from two features X_s, X_c, and an output Y. All three are continuous with 2000 samples drawn from normal distribution with zero mean and unit variance. We assume that feature X_s is already in the set of selected features, and treat X_c as the current candidate. Figure 1a shows how $S_{\text{QMIFS}}(X_c)$ changes while keeping correlation corr(X_c,X_s) fixed at 0.1, corr(X_s,Y) at 0.6, and varying the correlation between X_c and output Y from 0 to 1. As the correlation increases, $S_{\text{QMIFS}}(X_c)$ also increases, but non-linearly. This behaviour is expected, since correlation is not comparable to quadratic mutual information. Figure 1b shows the opposite, how increasing the correlation between features affects the criterion value. We fix correlations corr(X_c,Y) and corr(X_c,Y) to 0.6 and vary the corr(X_c,X_s) from 0 to 1. The result shows that S_{QMIFS} penalizes redundant features—the higher the redundancy (represented here as inter-feature correlation), the lower the criterion value. These findings demonstrate that S_{QMIFS} follows the aforementioned propriety of guaranteeing the correct ordering of features.

Authors in [33] present an efficient approach to speeding up the computation of I_{CS} with an insignificant loss to precision. The basic algorithm for computing I_{CS} has a time complexity of $O(n^2)$. They use a greedy incomplete Cholesky decomposition algorithm in order to achieve the computational complexity of $O(nd^2)$, where d depends on the data. This approach is useful only when

$d^2 < n$. In their work, they achieve substantial time savings when dealing with common data sets, so we adopted their approach in the computation of I_{CS}.

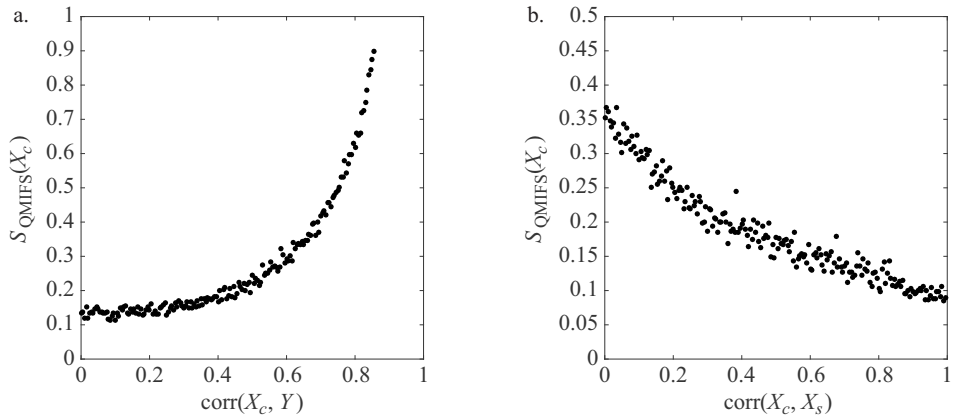

Figure 1. Properties of quadratic mutual information feature selection (QMIFS) criterion: (**a**) relevance of feature X_c as its correlation with the output Y increases; and (**b**) redundancy of feature X_c as its inter-feature correlation with already selected feature X_s increases.

4. Results and Discussion

For our experiments, we use ten data sets; nine are from the UCI machine learning repository [34], and one is from a company which deals with web advertisement placement. To compare the methods over a wide variety of scenarios, we choose the data sets so that some include only discrete data, some only continuous, and some mixed. The experiments cover two problem domains: one is dealing with classification, and the other with regression. Table 1 briefly summarizes the information about the data sets. For each data set it lists number of instances, number of features and their type, the type of the output, and the problem domain.

Table 1. Properties of used data sets. All but the Web Advertisement data set are from the UCI collection.

Data Set	Instances	Features		Output	Problem Domain
		Discrete	Continuous		
Chess	1000	36	0	binary	Classification
Breast Cancer	569	0	30	binary	
Ionosphere	351	0	34	binary	
Sonar	208	0	60	binary	
Wine	178	0	13	ternary	
Communities	1993	0	100	continuous	Regression
Parkinson Telemonitoring	1000	0	16	continuous	
Wine Quality	1599	0	11	continuous	
Housing	506	12	1	continuous	
Web Advertisement	950	38	8	continuous	

4.1. Experimental Methodology

We compare our method QMIFS to three other common and comparable methods which use an information-theoretic approach to feature selection: MIFS with $\beta = 1$, MRMR, and JMI. These three methods all need discretization of the continuous features before using them. The results are obtained using Matlab R2016a running on an Intel i7-6820HQ processor (Intel, Santa Clara, CA, USA) with 16 GB of main memory.

Classification tree from the Matlab Statistics and Machine Learning Toolbox serves as the indirect performance evaluation tool on the classification problem domain. The MDL discretization procedure from WEKA [35]—which promises better results than the usual approach of equal frequency or equal width binning [12]—acts as the preprocessing step, where needed. We evaluate the performance of the methods using the classification accuracy (CA), the area under the curve (AUC), Youden index (Y-index)—the difference between true positive rate (TPR) and false positive rate (FPR)—calculated in the optimal receiver operating characteristic (ROC) point, and the execution time.

In the regression problem domain, we asses the performance using the regression tree from the Matlab Statistics and Machine Learning Toolbox and measure the root-mean-square-error (RMSE) along with the execution time. As the output is continuous, MDL discretization is useless. Instead, equal frequency binning is used, with five bins for every feature and output. Equal frequency binning usually works better than equal width binning [12], and the empirical evidence from experimenting with MDL discretization shows that the number of bins per feature is often between three and seven.

In both problem domains, one thousand hold-out validations are performed on each data set. Each time, two thirds of randomly sampled instances act as the training set to build the model and the rest as the validation set to measure the performance. For each method, we vary the number of selected features: 3, 5, 7, or 10, and compare the results against the baseline performance where all features are used to train the model.

To get a clearer representation of result in both problem domains, we rank the methods according to the measures CA, AUC, Y-index, and RMSE. Each method obtains a rank from 1 (best) to 4 (worst). The ranked values get the same (average) rank if their 95% confidence intervals overlap.

4.2. Classification Performance

Table 2 shows which features are selected by each method, and Table 3 summarizes the ranking of the methods for each test scenario for measures CA, AUC, and Y-index and their average ranks. The ranks imply that all three measures behave similarly, which is expected since the data sets are well balanced with respect to the number of class values. Table 4 shows a more detailed insight into the performance of the methods for seven selected features. It includes only the maximum standard error of the performance indexes, as standard errors across different methods are practically the same. Additionally, Figure 2 reveals how different numbers of selected features affect the performance in terms of CA.

Chess data set: Baseline performs better in this case—looks like the learning machine can handle all 36 features. CA drops by about 0.03 after reducing the number of features to seven, and all the methods show similar behaviour in prioritizing features. According to Tables 3 and 4, our method is better than the others when selecting five, seven, and ten features, with regards to all three performance indexes. The time measurements in Table 4 show that it is also at least seven times faster at selecting seven features.

Breast Cancer data set: In this case, classification tree benefits from feature selection—even with only three features selected. Table 3 shows that all methods perform similarly; the largest discrepancy among them being at five selected features, where JMI overcomes the others in all three performance indexes. Again, QMIFS is the fastest method, with a three-to-four times lower running time.

Ionosphere data set: All methods improve the performance compared to the baseline. Our method does not perform very well in terms of CA, AUC, or Y-index, even though Table 2 shows that five out of seven selected features are the same as in the best performing method—JMI. It ranks second at three selected features, but then falls behind when selecting more of them. However, in terms of execution time, it is again three-to-four times faster.

Sonar data set: Only a few features are common to all the methods, so the performance varies substantially between them. At three selected features, JMI and QMIFS work the best and offer similar performance, having the same AUC ranks, with CA and Y-index being worse for QMIFS. At five features, CA improves for all methods but is overall still worse than the baseline. Using seven features

selected by MIFS, JMI, or QMIFS offers a considerable improvement in comparison to the baseline (3% better CA). The methods achieve the same ranks since the differences between them are small, causing the confidence intervals to overlap. At 10 selected features, all the methods offer improvement over the baseline, with JMI and MRMR having 2% better CA than QMIFS and MIFS. Our method again has the lowest execution time when selecting seven features.

Wine data set: Feature selection improves performance in comparison to the baseline, even though there are only 11 features in the data set. All methods select similar features, manifesting in similar performance. This can be seen in the rankings and in Table 4. QMIFS achieves the best ranks when selecting five, seven, or ten features, and is also at least 1.5 times faster than the other three.

Table 4 reveals that CA, AUC, and Y-index behave similarly because the data sets used are well balanced in terms of class values. In all cases except the Chess data set, the classification tree benefits from the feature selection with a 0.01–0.03 increase in CA. The differences between methods in terms of CA, AUC, and Y-index are small—relative difference is mainly less than 1%. The execution times clearly show that our method is the fastest. Due to the similarity of the first-order methods MRMR and MIFS, their execution times are equal and smaller in comparison to the second order method (JMI). Even though the time measurements are given only for seven selected features, the behaviour is similar in all test cases. What causes the other methods to be considerably slower is the MDL discretization done beforehand, which produces a large amount of computational overhead.

Overall, QMIFS offers performance similar to the other methods in terms of CA, AUC, and Y-index. Its average ranks shown in Table 3 across all data sets and number of selected features are 2.4/2.3/2.4, placing it somewhere in the middle—better than MIFS and MRMR, but lagging behind JMI. The subtle differences in the rankings can be attributed to the fact that both QMIFS and JMI are second-order methods and can detect some more peculiar relations between features. The difference between QMIFS and JMI could be attributed to the superiority of MDL discretization compared to the direct estimation in the case of QMIFS.

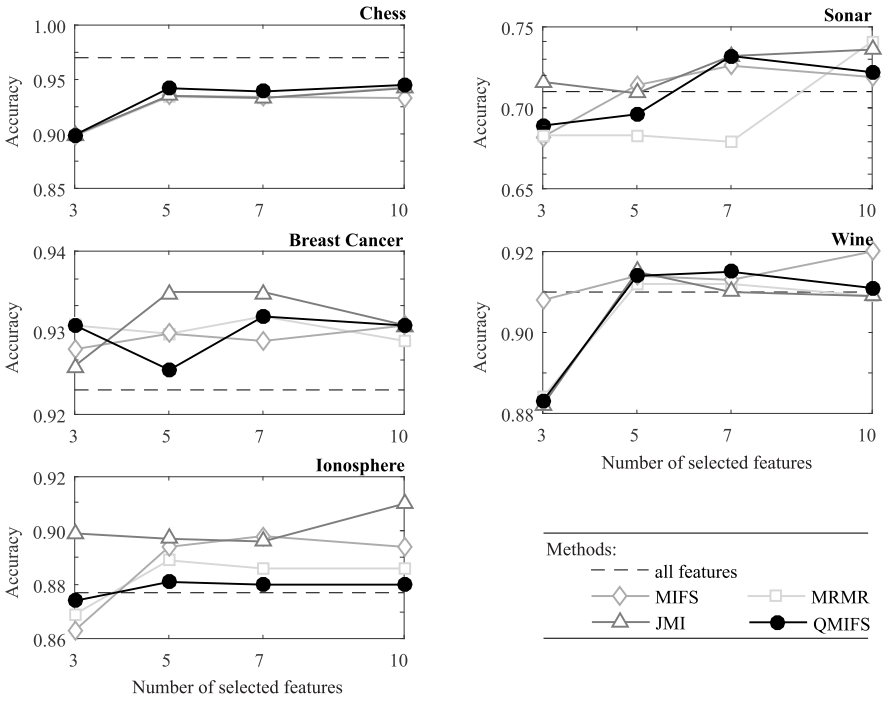

Figure 2. Performance of feature selection methods in the classification problem domain given in terms of classification accuracy of a classification tree. Higher values mean better performance. JMI: joint mutual information; MIFS: mutual information feature selection; MRMR: minimum redundancy maximum relevance; QMIFS: quadratic MIFS.

Table 2. Classification problem domain: selected features.

Data Set	Method	Selected Features			
		3	**5**	**7**	**10**
Chess	MIFS	21, 10, 33	32, 9	1, 2	3, 4, 5
	MRMR	21, 10, 33	32, 15	8, 16	18, 6, 27
	JMI	21, 10, 33	32, 15	8, 16	18, 6, 7
	QMIFS	33, 10, 21	35, 6	8, 18	7, 15, 13
Breast cancer	MIFS	23, 22, 5	19, 10	12, 15	30, 18, 29
	MRMR	23, 22, 28	14, 27	21, 8	29, 25, 24
	JMI	23, 28, 24	8, 22	21, 4	7, 27, 14
	QMIFS	28, 2, 23	8, 27	21, 22	3, 7, 1
Ionosphere	MIFS	5, 16, 2	18, 1	10, 30	32, 7, 24
	MRMR	5, 16, 18	27, 3	6, 7	4, 34, 31
	JMI	5, 6, 3	33, 8	17, 21	4, 13, 29
	QMIFS	5, 6, 33	15, 8	7, 21	28, 31, 24
Sonar	MIFS	11, 51, 36	44, 4	1, 2	3, 6, 7
	MRMR	11, 51, 36	48, 12	9, 54	45, 4, 21
	JMI	11, 4, 12	48, 9	21, 45	10, 36, 49
	QMIFS	12, 27, 11	48, 10	16, 9	13, 49, 28
Wine	MIFS	7, 1, 11	5, 3	4, 9	8, 10, 2
	MRMR	7, 1, 13	11, 10	12, 6	5, 2, 4
	JMI	7, 1, 13	11, 10	12, 6	2, 5, 4
	QMIFS	7, 1, 13	12, 10	11, 6	5, 9, 4

Table 3. Classification problem domain: ranking of feature selection methods. Ranks calculated from the measures classification accuracy (CA), area under the curve (AUC), and Y-index (the difference between true positive rate, TPR, and false positive rate, FPR) are presented as triplets CA/AUC/Y-index.

Data Set	Method	Selected Features				Average	
		3	**5**	**7**	**10**		
Chess	MIFS	2.5/2.5/2.5	3/3/3	2/4/3	4/4/4	2.9/3.4/3.1	
	MRMR	2.5/2.5/2.5	3/3/3	4/2.5/3	2.5/1.5/2.5	3.0/2.4/2.8	
	JMI	2.5/2.5/2.5	3/3/3	3/2.5/3	2.5/3/2.5	2.8/2.8/2.8	
	QMIFS	2.5/2.5/2.5	1/1/1	1/1/1	1/1.5/1	1.4/1.5/1.4	
Breast Cancer	MIFS	3.5/4/3.5	2.5/3/3	4/2.5/4	2/4/2.5	3.0/3.4/3.3	
	MRMR	1.5/2/1.5	2.5/3/2	2/2.5/1.5	4/2/2.5	2.5/2.4/1.9	
	JMI	3.5/2/3.5	1/1/1	2/2.5/1.5	2/2/2.5	2.1/1.9/2.1	
	QMIFS	1.5/2/1.5	4/3/4	2/2.5/3	2/2/2.5	2.4/2.4/2.8	
Iono-sphere	MIFS	3.5/3.5/3.5	1.5/3.5/2.5	1.5/2.5/1.5	2/2/2	2.1/2.9/2.4	2.5/2.9/2.7
	MRMR	3.5/3.5/3.5	3.5/2/2.5	3.5/2.5/3	3/3.5/3	3.4/2.9/3.0	3.0/2.8/2.8
	JMI	1/1/1	1.5/1/1	1.5/1/1.5	1/1/1	1.3/1.0/1.1	2.1/2.1/2.1
	QMIFS	2/2/2	3.5/3.5/4	3.5/4/4	4/3.5/4	3.3/3.3/3.5	2.4/2.3/2.4
Sonar	MIFS	3/3/3	1/1/1	2/2/2	3.5/3.5/3.5	2.4/2.4/2.4	
	MRMR	4/4/3	3.5/4/4	4/4/4	1.5/1.5/1.5	3.3/3.4/3.1	
	JMI	1/1.5/1	2/2.5/2	2/2/2	1.5/1.5/1.5	1.6/1.9/1.6	
	QMIFS	2/1.5/3	3.5/2.5/3	2/2/2	3.5/3.5/3.5	2.8/2.4/2.9	
Wine	MIFS	1/2.5/1	2.5/3/3	2.5/2.5/3	1.5/1.5/2.5	1.9/2.4/2.4	
	MRMR	3/2.5/3	2.5/3/3	2.5/2.5/3	3.5/3/4	2.9/2.8/3.3	
	JMI	3/2.5/3	2.5/3/3	2.5/2.5/3	3.5/3/2.5	2.9/2.8/2.9	
	QMIFS	3/2.5/3	2.5/1/1	2.5/2.5/1	1.5/1.5/1	2.4/1.9/1.5	

Table 4. Classification problem domain: values of measures and the execution times for seven selected features. The values obtained with all features are shown for comparison. Maximum standard error for given measure and data set is included in parentheses.

Data Set	Method	Measure				
		CA	**AUC**	**Y-Index**	**FPR, TPR**	**Time (s)**
Chess	MIFS	0.934	0.956	0.865	0.106, 0.970	2.48
	MRMR	0.932	0.964	0.866	0.099, 0.965	2.51
	JMI	0.933	0.965	0.866	0.098, 0.965	2.68
	QMIFS	0.938	0.976	0.886	0.032, 0.918	0.32
	All features	0.969 ($\pm 4 \times 10^{-4}$)	0.985 ($\pm 3 \times 10^{-4}$)	0.943 ($\pm 7 \times 10^{-4}$)	0.032, 0.975	($\pm$0.04)
Breast cancer	MIFS	0.929	0.939	0.858	0.095, 0.953	1.53
	MRMR	0.932	0.942	0.868	0.080, 0.948	1.52
	JMI	0.935	0.944	0.873	0.074, 0.947	1.56
	QMIFS	0.932	0.941	0.866	0.080, 0.947	0.42
	All features	0.923 ($\pm 6 \times 10^{-4}$)	0.927 ($\pm 8 \times 10^{-4}$)	0.849 ($\pm 1 \times 10^{-3}$)	0.093, 0.941	($\pm$0.02)
Ionosphere	MIFS	0.897	0.904	0.794	0.150, 0.944	1.15
	MRMR	0.888	0.903	0.780	0.152, 0.931	1.17
	JMI	0.897	0.915	0.796	0.138, 0.933	1.21
	QMIFS	0.881	0.897	0.764	0.160, 0.924	0.39
	All features	0.877 ($\pm 1 \times 10^{-3}$)	0.878 ($\pm 1 \times 10^{-3}$)	0.755 ($\pm 2 \times 10^{-3}$)	0.164, 0.919	($\pm$0.02)
Sonar	MIFS	0.728	0.755	0.504	0.267, 0.770	1.18
	MRMR	0.681	0.711	0.424	0.301, 0.725	1.20
	JMI	0.731	0.759	0.510	0.280, 0.790	1.23
	QMIFS	0.734	0.759	0.506	0.270, 0.777	0.38
	All features	0.708 ($\pm 2 \times 10^{-3}$)	0.724 ($\pm 2 \times 10^{-3}$)	0.452 ($\pm 3 \times 10^{-3}$)	0.302, 0.754	($\pm$0.02)
Wine	MIFS	0.911	0.919	0.826	0.050, 0.876	0.51
	MRMR	0.910	0.917	0.828	0.070, 0.898	0.49
	JMI	0.911	0.916	0.829	0.069, 0.898	0.54
	QMIFS	0.915	0.923	0.839	0.068, 0.907	0.29
	All features	0.906 ($\pm 1 \times 10^{-3}$)	0.912 ($\pm 1 \times 10^{-3}$)	0.813 ($\pm 3 \times 10^{-3}$)	0.066, 0.878	($\pm$0.02)

4.3. Regression Performance

Table 5 shows which features are selected by each method, and Table 6 summarizes the ranking of the methods for each test scenario and the average ranks. Table 7 and Figure 3 show the RMSE for each method and data set. Additionally, execution times are presented in Table 7.

Table 5. Regression problem domain: selected features.

Data Set	Method	Selected Features			
		3	**5**	**7**	**10**
Communities	MIFS	45, 52, 67	95, 97	48, 36	24, 15, 89
	MRMR	45, 52, 4	41, 51	72, 69	18, 3, 50
	JMI	45, 4, 44	50, 69	51, 46	41, 3, 16
	QMIFS	45, 41, 78	42, 44	4, 68	29, 39, 16
Parkinson Telemonitoring	MIFS	15, 12, 14	16, 8	5, 10	2, 13, 4
	MRMR	15, 12, 14	8, 16	2, 10	5, 13, 7
	JMI	15, 14, 2	6, 13	9, 4	10, 7, 11
	QMIFS	15, 14, 13	2, 10	8, 4	11, 9, 6
Wine Quality	MIFS	11, 10, 6	4, 9	2, 5	8, 7, 3
	MRMR	11, 10, 2	5, 7	8, 4	9, 3, 6
	JMI	11, 10, 8	3, 2	5, 7	1, 4, 6
	QMIFS	11, 10, 2	8, 3	7, 1	9, 6, 5
Housing	MIFS	13, 11, 4	6, 12	7, 9	8, 2, 10
	MRMR	13, 11, 6	12, 7	10, 4	3, 1, 5
	JMI	13, 6, 11	3, 1	10, 5	2, 7, 9
	QMIFS	13, 8, 3	6, 7	11, 5	10, 9, 4
Web Advertisement	MIFS	29, 9, 21	44, 35	15, 13	12, 22, 6
	MRMR	29, 9, 21	44, 35	39, 13	12, 15, 22
	JMI	29, 4, 41	30, 28	10, 39	31, 3, 37
	QMIFS	3, 8, 40	2, 4	46, 16	36, 10, 45

Table 6. Regression problem domain: ranking of feature selection methods.

Data Set	Method	Selected Features				Average	
		3	**5**	**7**	**10**		
Communities	MIFS	1.5	2	4	4	2.9	
	MRMR	1.5	2	1	2	1.6	
	JMI	3	2	2.5	1	2.1	
	QMIFS	4	4	2.5	3	3.4	
Parkinson Telemonitoring	MIFS	3.5	3.5	4	1.5	3.1	
	MRMR	3.5	3.5	1	1.5	2.4	
	JMI	1	1.5	2.5	3.5	2.1	
	QMIFS	2	1.5	2.5	3.5	2.4	
Wine Quality	MIFS	3	4	2.5	2.5	3.0	3.1
	MRMR	1.5	1	2.5	2.5	1.9	2.5
	JMI	4	2.5	2.5	2.5	2.9	2.4
	QMIFS	1.5	2.5	2.5	2.5	2.3	2.1
Housing	MIFS	4	3.5	3.5	2	3.3	
	MRMR	1.5	3.5	3.5	4	3.1	
	JMI	1.5	2	2	3	2.1	
	QMIFS	3	1	1	1	1.5	
Web Advertisement	MIFS	3.5	3.5	3	2	3.0	
	MRMR	3.5	3.5	4	3	3.5	
	JMI	2	2	2	4	2.5	
	QMIFS	1	1	1	1	1.0	

Table 7. Regression problem domain: values of root-mean-square-error (RMSE) measure and execution times for different numbers of selected features. Column *All Features* holds the value of RMSE obtained with all features and maximum standard error given in parentheses.

Data Set	Method	RMSE					Time (s)			
		3	**5**	**7**	**10**	**All Features**	**3**	**5**	**7**	**10**
Communities	MIFS	0.176	0.180	0.186	0.188	0.190	0.97	1.43	1.87	2.49
	MRMR	0.176	0.180	0.180	0.181		0.97	1.42	1.86	2.50
	JMI	0.182	0.181	0.183	0.183		1.53	2.34	3.22	4.49
	QMIFS	0.190	0.190	0.183	0.185	($\pm 3 \times 10^{-4}$)	2.74	4.06	5.46	7.51 ($\pm$0.02)
Parkinson Telemonitoring	MIFS	9.12	8.89	8.93	8.21	8.30	0.32	0.35	0.37	0.40
	MRMR	9.11	8.92	8.23	8.23		0.32	0.35	0.37	0.40
	JMI	8.40	8.32	8.43	8.48		0.38	0.43	0.48	0.54
	QMIFS	8.83	8.34	8.42	8.45	($\pm 2 \times 10^{-2}$)	0.49	0.59	0.66	0.77 ($\pm$0.01)
Wine Quality	MIFS	0.772	0.789	0.772	0.773	0.771	0.32	0.35	0.37	0.39
	MRMR	0.760	0.758	0.773	0.773		0.32	0.35	0.37	0.39
	JMI	0.798	0.775	0.770	0.772		0.36	0.42	0.45	0.48
	QMIFS	0.760	0.774	0.771	0.770	($\pm 1 \times 10^{-3}$)	0.42	0.43	0.46	0.48 ($\pm$0.01)
Housing	MIFS	5.49	5.12	5.12	4.74	4.63	0.29	0.30	0.30	0.32
	MRMR	5.08	5.10	5.06	4.93		0.29	0.30	0.31	0.32
	JMI	5.08	5.03	4.85	4.90		0.31	0.33	0.35	0.37
	QMIFS	5.25	4.99	4.61	4.54	($\pm 3 \times 10^{-2}$)	0.34	0.38	0.41	0.43 ($\pm$0.01)
Web Advertisement	MIFS	3.746	3.757	3.668	3.466	3.59	0.51	0.60	0.69	0.82
	MRMR	3.746	3.755	3.703	3.526		0.51	0.60	0.72	0.84
	JMI	3.562	3.699	3.567	3.582		0.63	0.88	1.12	1.43
	QMIFS	2.953	2.956	3.174	3.175	($\pm 6 \times 10^{-3}$)	3.19	3.49	3.76	4.17 ($\pm$0.02)

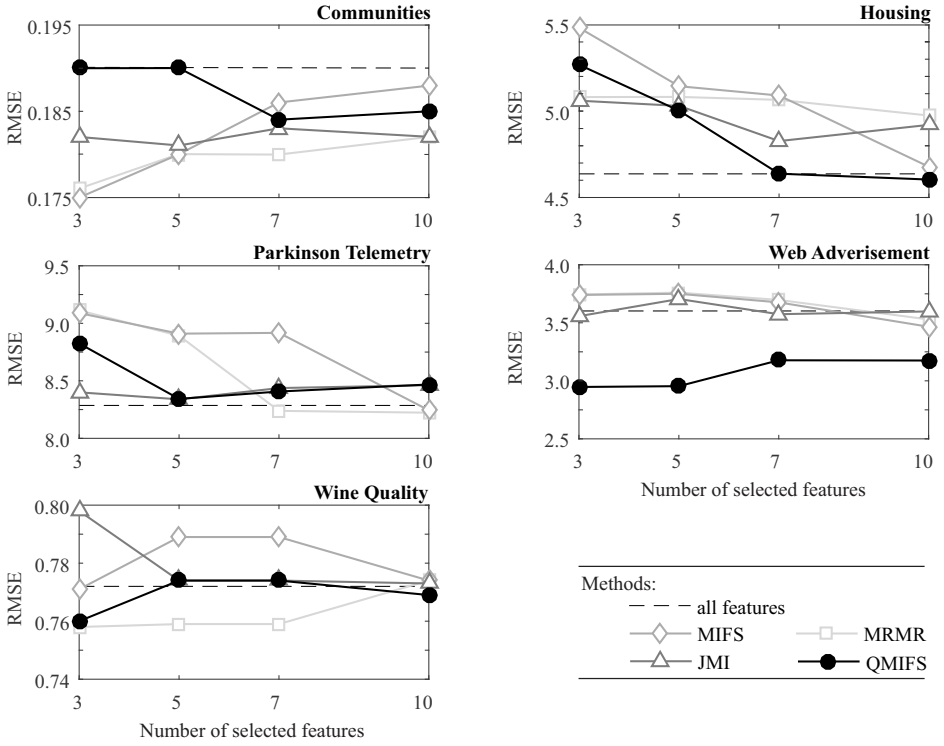

Figure 3. Performance of feature selection methods in the regression problem domain given in terms of RMSE achieved by regression tree. Lower values indicate better performance.

Communities data set: Table 7 shows that the methods improve RMSE in all cases, even if we use only three features to train the model. This is expected, since the number of features in the data set is quite large (100) and difficult for the learning machine to tackle. Our method ranks last when selecting three or five features, but improves afterwards with RMSE comparable to other three methods (second and third best at seven and ten selected features, respectively). The selected features across different methods are much more versatile on this data set, owing to the fact that there are many input features to begin with. It is slower than the other three methods by a factor of 1.5–3.

Parkinson Telemonitoring data set: There is only a small gain in the performance by using at least seven features chosen by MRMR. The top three ranking features across all the methods are very similar, with only JMI offering 6–8% lower RMSE in comparison to others. However, our method performs equally well as JMI for five and more features. The execution times are comparable, with the first-order methods being faster, which is expected.

Wine Quality data set: In some cases, feature selection offers an improvement in the regression performance even though the total number of features in the data set is only 11. Overall, our method and MRMR are superior to MIFS and JMI, selecting similar features and offering improvement over the baseline. The execution times behave similarly as in the previous case.

Housing data set: The baseline performs better here for the most part, but there are only 13 features in the data set, so the learning machine does not have a difficult task in training the model. Only our method shows a small performance (2%) benefit compared to baseline when using the top ten features, and it achieves the best overall performance among the four methods, with an average rank of 1.5. Execution times are roughly 30% higher for the second order methods.

Web Advertisement data set: Our method improves the model's performance dramatically compared to the baseline and other feature selection methods, which all exhibit similar behaviour. The number of input features in the data set is large enough to pose a difficult task to the learning machine, so it benefits considerably from feature selection, at least when QMIFS is used. However, our method is much slower than the other three methods—by a factor of 3–6.

In terms of average RMSE ranks, our method outperforms the other three, achieving a value of 2.1 across all test cases. JMI and MRMR are tied for the second place, with average ranks of 2.4 and 2.5; MIFS is lagging behind, with an average rank of 3.1. These results suggest that without the possibility of using MDL to discretize the data, the other methods lag behind our approach. There are probably not many higher-order relations in the data, since JMI is comparable to MRMR in terms of overall performance. Obviously, the way in which underlying probability densities are estimated has a higher impact on the performance than the order of the method. We believe that QMIFS better distinguishes relations in the data than ad-hoc binning used in the other three methods.

Due to the higher versatility of the dependent variable values in the regression problem domain, incomplete Cholesky decomposition is not so effective, leading to longer execution times for our method. This is especially obvious in the Communities and Web advertisement data sets. Additionally, equal frequency binning causes much less computational overhead than MDL to MIFS, MRMR, and JMI, which consequently outperform QMIFS regarding execution times.

5. Conclusions

In this paper, we propose a quadratic mutual information feature selection method (QMIFS). Our goal was to detect second-order non-linear relations between features and the output, similarly to joint mutual information. Additionally, we focused on the analysis of both discrete and continuous features and outputs, avoiding the intermediate step of estimating underlying probability density functions using histograms or kernel density estimation. To achieve these goals, we employed a quadratic mutual information measure, as it enables direct estimation from the data samples. The measure itself does not exhibit all the properties intrinsic to mutual information measure, and therefore our method was developed to compensate for deficiencies.

We compare our method to three other methods based on information-theoretic measures: mutual information feature selection (MIFS), minimum redundancy maximum relevance (MRMR), and joint mutual information (JMI). The methods are compared indirectly, on the classification problem domain using models built by the classification tree learning machine, and on the regression problem domain using the regression tree learning machine. The results show that our method offers similar performance on the classification problem domain in terms of classification accuracy, area under the curve, and Youden index as the other three, but is considerably faster. When dealing with regression, it compares favourably to the others regarding root-mean-squared-error, but is slower.

We conclude that our method is universal, capable of feature selection on classification or regression problem domain. QMIFS does not need an additional preprocessing step to estimate the probability density function, as is the case in the other three methods. This and the fact that it avoids using parameters makes it simple to use for non-experts in the field. Experiments show that straightforward estimation of QMI from data samples using quadratic Renyi entropy and Gaussian kernels does a good job at identifying the important information in the data. Additionally, it offers considerable execution time savings compared to other feature selection methods coupled with advanced discretization techniques like MDL.

Future research should go towards finding better estimators for the width of the kernel, which importantly affects estimation of QMI. Potential other measures could also be investigated for compatibility with our approach. Moreover, the computational cost of the QMI and other potential measures can be further reduced by using the fast Gauss transform, as proposed in [7].

Acknowledgments: This research was supported by Slovenian Research Agency under grant P2-0241 (National research program Synergetics of complex systems and processes).

Author Contributions: Davor Sluga and Uroš Lotič conceived and designed the experiments, analyzed the data, and wrote the paper; Davor Sluga performed the experiments. Both authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
2. Vergara, J.R.; Estévez, P.A. A review of feature selection methods based on mutual information. *Neural Comput. Appl.* **2014**, *24*, 175–186.
3. Kohavi, R.; John, G.H. Wrappers for feature subset selection. *Artif. Intell.* **1997**, *97*, 273–324.
4. Hall, M.A. Correlation-based feature selection of discrete and numeric class machine learning. In Proceedings of the Seventeenth International Conference on Machine Learning, Stanford, CA, USA, 29 June–2 July 2000; pp. 359–366.
5. Fleuret, F. Fast binary feature selection with conditional mutual information. *J. Mach. Learn. Res.* **2004**, *5*, 1531–1555.
6. Chandrashekar, G.; Sahin, F. A survey on feature selection methods. *Comput. Electr. Eng.* **2014**, *40*, 16–28.
7. Principe, J.C. *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*; Springer Science & Business Media: New York, NY, USA, 2010.
8. Brown, G. A new perspective for information theoretic feature selection. In Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS-09), Clearwater Beach, FL, USA, 16–18 April 2009; pp. 49–56.
9. Gonçalves, L.B.; Macrini, J.L.R. Rényi entropy and Cauchy-Schwarz mutual information applied to mifs-u variable selection algorithm: A comparative study. *Pesqui. Oper.* **2011**, *31*, 499–519.
10. Sluga, D.; Lotric, U. Generalized information-theoretic measures for feature selection. In Proceedings of the International Conference on Adaptive and Natural Computing Algorithms, Lausanne, Switzerland, 4–6 April 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 189–197.
11. Chow, T.W.; Huang, D. Estimating optimal feature subsets using efficient estimation of high-dimensional mutual information. *IEEE Trans. Neural Netw.* **2005**, *16*, 213–224.

12. Garcia, S.; Luengo, J.; Sáez, J.A.; Lopez, V.; Herrera, F. A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 734–750.
13. Irani, K.B. Multi-interval discretization of continuous-valued attributes for classification learning. In Proceedings of the 13th International Joint Conference on Artificial Intelligence, Chambery, France, 28 August–3 September 1993; pp. 1022–1029.
14. Parzen, E. On estimation of a probability density function and mode. *Ann. Math. Stat.* **1962**, *33*, 1065–1076.
15. Katkovnik, V.; Shmulevich, I. Kernel density estimation with adaptive varying window size. *Pattern Recognit. Lett.* **2002**, *23*, 1641–1648.
16. Kraskov, A.; Stögbauer, H.; Grassberger, P. Estimating mutual information. *Phys. Rev. E* **2004**, *69*, 066138.
17. Walters-Williams, J.; Li, Y. Estimation of mutual information: A survey. In Proceedings of the International Conference on Rough Sets and Knowledge Technology, Gold Coast, QLD, Australia, 14–16 July 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 389–396.
18. Sugiyama, M. Machine learning with squared-loss mutual information. *Entropy* **2012**, *15*, 80–112.
19. Beck, C. Generalised information and entropy measures in physics. *Contemp. Phys.* **2009**, *50*, 495–510.
20. Renyi, A. On measures of entropy and information. In Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA, USA, 20 June–30 July 1960; pp. 547–561.
21. Erdogmus, D.; Principe, J.C. Generalized information potential criterion for adaptive system training. *IEEE Trans. Neural Netw.* **2002**, *13*, 1035–1044.
22. Renyi, A. *Some Fundamental Questions About Information Theory*; Akademia Kiado: Budapest, Hungary, 1976; Volume 2.
23. Battiti, R. Using mutual information for selecting features in supervised neural net learning. *IEEE Trans. Neural Netw.* **1994**, *5*, 537–550.
24. Kwak, N.; Choi, C.H. Input feature selection for classification problems. *IEEE Trans. Neural Netw.* **2002**, *13*, 143–159.
25. 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.* **2005**, *27*, 1226–1238.
26. Yang, H.; Moody, J. Feature selection based on joint mutual information. In Proceedings of the International ICSC Symposium on Advances in Intelligent Data Analysis, Rochester, NY, USA, 22–25 June 1999; pp. 22–25.
27. Rajan, K.; Bialek, W. Maximally informative "stimulus energies" in the analysis of neural responses to natural signals. *PLoS ONE* **2013**, *8*, e71959.
28. Fitzgerald, J.D.; Rowekamp, R.J.; Sincich, L.C.; Sharpee, T.O. Second order dimensionality reduction using minimum and maximum mutual information models. *PLoS Comput. Biol.* **2011**, *7*, e1002249.
29. Rowekamp, R.J.; Sharpee, T.O. Analyzing multicomponent receptive fields from neural responses to natural stimuli. *Netw. Comput. Neural Syst.* **2011**, *22*, 45–73.
30. Sánchez-Maroño, N.; Alonso-Betanzos, A.; Tombilla-Sanromán, M. Filter methods for feature selection—A comparative study. In Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning, Birmingham, UK, 16–19 December 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 178–187.
31. Frénay, B.; Doquire, G.; Verleysen, M. Is mutual information adequate for feature selection in regression? *Neural Netw.* **2013**, *48*, 1–7.
32. Silverman, B.W. *Density Estimation for Statistics and Data Analysis*; CRC Press: Boca Raton, FL, USA, 1986; Volume 26.
33. Seth, S.; Príncipe, J.C. On speeding up computation in information theoretic learning. In Proceedings of the International Joint Conference on Neural Networks (IJCNN 2009), Atlanta, GA, USA, 14–19 June 2009; pp. 2883–2887.
34. Lichman, M. UCI Machine Learning Repository. Available online: http://archive.ics.uci.edu/ml (accessed on 1 December 2016).
35. Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I.H. The WEKA data mining software: An update. *ACM SIGKDD Explor. Newsl.* **2009**, *11*, 10–18.