Journal of Artificial Intelligence Research 82 (2025) 1267-1292 Submitted 09/2024; published 03/2025

Improving Mutual Information Based Feature Selection by Boosting Unique Relevance

Shiyu Liu SHIYU_LIU@U.NUS.EDU
Department of Electrical and Computer Engineering,
College of Design and Engineering,
National University of Singapore, Singapore

Mehul Motani MOTANI@NUS.EDU.SG
Department of Electrical and Computer Engineering,
College of Design and Engineering, Institute of Data Science,
N.1 Institute for Health, Institute for Digital Medicine,
National University of Singapore, Singapore

Abstract

Mutual Information (MI) based feature selection makes use of MI to evaluate each feature and eventually shortlists a relevant feature subset, in order to address issues associated with high-dimensional datasets. Despite the effectiveness of MI in feature selection, we notice that many state-of-the-art algorithms disregard the so-called unique relevance (UR) of features, which is a necessary condition for the optimal feature subset. In our study of five representative MI based feature selection (MIBFS) algorithms, we find that all of them underperform as they ignore the UR of features and arrive at a suboptimal selected feature subset. We point out that the heart of the problem is that all these MIBFS algorithms follow the criterion of Maximize Relevance with Minimum Redundancy (MRwMR), which does not explicitly target UR. This motivates us to augment the existing criterion with the objective of boosting unique relevance (BUR), leading to a new criterion called MRwMR-BUR. Depending on the task being addressed, MRwMR-BUR has two variants, termed MRwMR-BUR-KSG and MRwMR-BUR-CLF, which estimate UR differently. MRwMR-BUR-KSG estimates UR via a nearest-neighbor based approach called the KSG estimator and is designed for three major tasks: (i) Classification Performance (i.e., higher classification accuracy). (ii) Feature Interpretability (i.e., a more precise selected feature subset for practitioners to explore the hidden relationship between features and labels). (iii) Classifier Generalization (i.e., the selected feature subset generalizes well to various classifiers). MRwMR-BUR-CLF estimates UR via a classifier based approach. It adapts UR to different classifiers, further improving the competitiveness of MRwMR-BUR for classification performance oriented tasks. The performance of MRwMR-BUR-KSG and MRwMR-BUR-CLF is validated via experiments using six public datasets and four popular classifiers. Specifically, as compared to MRwMR, the proposed MRwMR-BUR-KSG improves the test accuracy by 2% – 3% with 25% – 30% fewer features being selected, without increasing the algorithm complexity. MRwMR-BUR-CLF further improves the classification performance by 3.8% – 5.5% (relative to MRwMR), and it also outperforms three popular classifier dependent feature selection methods.

1. Introduction

High-dimensional datasets tend to contain irrelevant and redundant features, leading to extra computation, larger storage, and degraded performance (Bengio, Courville, & Vincent,

2013; Gao et al., 2016; El-Hasnony et al., 2020; Zhou, Wang, & Zhu, 2022; Salem et al., 2022; Fan et al., 2024). Mutual Information (MI) (Cover & Thomas, 2006) based feature selection, which is a classifier independent filter method in the field of dimensionality reduction, attempts to address those issues by selecting a relevant feature subset. We start this work by discussing the advantage of MI based feature selection (MIBFS) over other types of dimensionality reduction methods.

(1) Feature Interpretability. Dimensionality reduction methods can be divided into two classes: feature extraction and feature selection. Feature extraction transforms original features into new features with lower dimensionality while preserving the key information in the original features. As an example, the principal component analysis (PCA) (Jolliffe, 2005) projects data points onto only the first few principal components while preserving as much of the data's variation as possible. Feature extraction may perform well in dimensionality reduction, but the extraction process (e.g., projection) loses the physical meaning of original features (Chandrashekar & Sahin, 2014; Sun et al., 2014; Nguyen et al., 2014; Lamba, Gulati, & Jain, 2022). In contrast, feature selection preserves the feature interpretability by selecting a relevant feature subset. This helps to explore the hidden relationship between variables and makes techniques such as MIBFS preferred in various domains (Kim et al., 2015; Chandrashekar & Sahin, 2014; Hassan et al., 2022; Tripathi, Hemachandra, & Trivedi, 2020; Htun, Biehl, & Petkov, 2023; Theng & Bhoyar, 2024; Fan et al., 2024; Ma & Lu, 2024). Examples of such studies are (El-Kenawy et al., 2020; Sun et al., 2020; Too & Mirjalili, 2021; Luque-Rodriguez & et al, 2022), which make use of feature selection to shortlist a set of medical features associated with the coronavirus. The selected feature subset could provide new insights on the diagnosis of coronavirus and further advance the research on it.

(2) Classifier Generalization. Feature selection methods are either classifier dependent or classifier independent (Guyon & Elisseeff, 2003; Chandrashekar & Sahin, 2014). Examples of the former type include the wrapper method (e.g., forward feature selection (Marcano-Cedeño et al., 2010)) and the embedded method which performs feature selection during the training of a pre-defined classifier (e.g., LASSO with the L1 penalty (Hastie et al., 2015)). The classifier dependent method tends to provide good performance as it directly makes use of the interaction between features and accuracy. However, the selected features are optimized for the pre-defined classifier and may not perform well for other classifiers (Solorio-Fernández et al., 2020; Zebari et al., 2020; Bommert et al., 2020). The filter method, which is classifier independent, scores each feature according to its relevance with the label. As a filter method, MIBFS quantifies relevance using MI as MI can capture the dependencies between random variables (i.e., feature and label). Consequently, the feature subset selected by MIBFS is not tied to the bias of the classifier and is easier to generalize to different classifiers (Bengio et al., 2013; Meyer et al., 2008; Cai et al., 2018; Khaire & Dhanalakshmi, 2019; Gu et al., 2022; He et al., 2024).

(3) Classification Performance. The objective of MIBFS is to find the minimal feature subset with maximum MI with respect to the label (Brown et al., 2012). Mathematically, the goal can be written as

$$S^* = \arg\min f(\arg\max_{S\subseteq\Omega} I(S;Y)), \tag{1}$$

where $f(A, B, \cdots) = (|A|, |B|, \cdots)$, $|A|$ represents the number of features in A and Ω is the set of all features, $S \subseteq \Omega$ is the selected feature subset and S^* is the optimal feature subset. Several works (Liu & Motani, 2020; Brown et al., 2012) have proved that maximizing $I(S;Y)$ is equivalent to maximizing the likelihood ($p(Y|S)$) and suggest that minimizing the size of S helps to improve the generalization performance on the unseen data (more details in Sections 3.1 and 3.2). Finding the optimal feature subset through exhaustive search is computationally intractable. Therefore, numerous MIBFS algorithms (Meyer et al., 2008; Yang & Moody, 2000; Peng, Long, & Ding, 2005) attempt to select the optimal feature subset following the criterion of Maximize Relevance with Minimum Redundancy (MRwMR) (Peng et al., 2005), where Maximize Relevance corresponds to the requirement of maximizing $I(S;Y)$ and Minimum Redundancy is to reduce the size of S. Those algorithms have provided competitive performance in dimensionality reduction (see recent survey works (Zebari et al., 2020; Venkatesh et al., 2019)).

In this paper, we explore a promising feature property, called Unique Relevance (UR), which is the key to select the optimal feature subset in (1). Specifically, UR is the unique relevant information with respected to the label, that is not shared by another features (See more details in Section 2.4). We note that UR has been defined for a long time and it is also known as strong relevance (John, Kohavi, & Pfleger, 1994). While many works (Yu & Liu, 2004; Kohavi & John, 1997) had defined UR and highlighted its importance, only (Liu, Yao, Zhou, & Motani, 2018) utilized UR to develop a new MIBFS algorithm called SURI. The use of UR for feature selection remains largely unexplored. We fill in this gap and improve the performance of MIBFS by exploring the utility of UR. We summarize the flow of the remaining paper together with our contributions as follows.

1. We shortlist five representative MIBFS algorithms and uncover the fact that all of them ignore UR and end up underperforming, namely they select a non-negligible number of redundant features, contradicting the objective of minimal feature subset in (1) (see Section 3.4).
2. We point out that the heart of the problem is that existing MIBFS algorithms follow the criterion of MRwMR (Peng et al., 2005), which lacks a mechanism to identify the UR of features. This motivates us to augment MRwMR and include the objective of boosting UR, leading to a new criterion for MIBFS, called MRwMR-BUR (see Section 3.5).
3. We estimate UR via a nearest neighbor based approach called KSG estimator (Kraskov, Stögbauer, & Grassberger, 2004), resulting in the first variant of MRwMR-BUR, called MRwMR-BUR-KSG. The MRwMR-BUR-KSG is designed to improve MRwMR for three major tasks: (i) Classification Performance (an improvement of 2 – 3% in test accuracy). (ii) Feature Interpretability (i.e., 25 – 30% fewer unnecessary features are selected). (iii) Classifier Generalization (i.e., the selected feature subset generalizes well to four popular classifiers studied) (see Section 4).
4. We propose a classifier based approach to estimate UR, resulting in the second variant of MRwMR-BUR, called MRwMR-BUR-CLF. The MRwMR-BUR-CLF adapts UR to different classifiers, further improving the competitiveness of MRwMR-BUR for classification performance oriented tasks. Through extensive experiments, we observe that MRwMR-BUR-CLF further improves the classification performance of MRwMR by 3.8% – 5.5% and it also outperforms three popular classifier dependent feature selection methods (see Section 5).

We note that a short version of this work has been published in (Liu & Motani, 2020), and this work extends (Liu & Motani, 2020) in three aspects: (i) We evaluate the proposed MRwMR-BUR criterion from the perspective of achieving the goal of MIBFS stated in (1). Our experimental results in Section 3.5 demonstrate that the proposed MRwMR-BUR can better achieve the goal of MIBFS than MRwMR. This serves as a more solid motivation for the proposed MRwMR-BUR. (ii) In addition to the classification accuracy, the performance of MRwMR-BUR is also evaluated in terms of Feature Interpretability (i.e., if the selected feature subset is precise for practitioners to explore the hidden relationship between features and labels) and Classifier Generalization (i.e., if the selected feature subset generalizes well to various classifiers). (iii) We improve the competitiveness of MRwMR-BUR for performance oriented tasks and propose a classifier based approach to estimate UR, leading to another variant of MRwMR-BUR, called MRwMR-BUR-CLF. The MRwMR-BUR-CLF further improves the performance of MRwMR by 3.8% – 5.5% and it also outperforms three popular classifiers based feature selection methods.

2. Background

We first provide a brief introduction to information theoretic concepts in Section 2.1, followed by a definition of the notation in Section 2.2. Next, in Section 2.3, we review existing works and shortlist five MRwMR based algorithms. In Section 2.4, we present three types of information content (unique relevance, conditional relevance, irrelevance). In Section 2.5, we have a discussion on the estimation of MI.

2.1 Entropy and Mutual Information

(1) Entropy. Let X be a discrete random variable with alphabet $\mathcal{X}$ and probability mass function $p(x) = Pr(X = x), x \in \mathcal{X}$. The entropy $H(X)$ measures the uncertainty present in the distribution of X (Cover & Thomas, 2006) and is defined as

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x). \tag{2}$$

To compute it, one possible approach is to estimate the distribution of $p(X)$ by the frequency counts from data (i.e., the fraction of observations taking on value x from the total N). We will discuss more details in Section 2.5. If the distribution of X is highly biased toward one particular event $x \in \mathcal{X}$, then the entropy is low. This makes sense as we have less uncertainty about the outcome. If all events are equally likely, we have maximum uncertainty over the outcome, then $H(X)$ is maximal.

(2) Conditional Entropy. The conditional entropy $H(Y|X)$ of a pair of discrete random variables (X, Y) with a joint distribution $p(x, y)$ is defined as

$$H(Y|X) = -\sum_{x \in \mathcal{X}} p(x) H(Y|x) \tag{3}$$

$$= -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(y|x). \tag{4}$$

This can be thought of as the amount of uncertainty remained in Y after learning the outcome of X.

(3) Mutual Information. The mutual information (Cover & Thomas, 2006) between two discrete random variables X and Y with a joint probability mass function $p(x, y)$ and marginal probability mass functions $p(x)$ and $p(y)$ is defined as

$$I(X;Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}. \tag{5}$$

By chain rule for mutual information, (5) can be rewritten as

$$I(X;Y) = H(Y) - H(Y|X). \tag{6}$$

This can be interpreted as the reduction in the uncertainty of Y due to the knowledge of X. Intuitively, it is the amount of information that one variable provides about another one. We note that the mutual information is symmetric (i.e., $I(X;Y) = I(Y;X)$), and is zero if and only if the variables are statistically independent. In contrast to correlation, MI captures non-linear relationship between variables, and thus can act as a measure of true dependence.

(4) Conditional Mutual Information. The conditioned form of mutual information is given by

$$I(X;Y|Z) = H(X|Z) - H(X|YZ) \tag{7}$$

$$= \sum_{z\in\mathcal{Z}} p(z)\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(xy|z)\log\frac{p(xy|z)}{p(x|z)p(y|z)}. \tag{8}$$

The conditional mutual information captures the information still shared between X and Y after knowing Z, which is a very important property to be considered in feature selection.

2.2 Notation

We now define the notation used in this paper. We denote the set of all features by $\Omega = \{X_k, k = 1, \cdots, M\}$, where M is the number of features. The feature $X_k \in \Omega$ and the label Y are both vectors of length N, where N is the number of samples. Let $S \subseteq \Omega$ be the set of selected features and $\tilde{S} \subseteq \Omega$ be the set of unselected features, i.e., $\Omega = \{S, \tilde{S}\}$.

2.3 Prior MIBFS Works

In MIBFS, MI and its variants (e.g., conditional MI, joint MI) are used as the core of a scoring function to measure how potentially useful a feature or a feature subset could be. Generally, the scoring function will assign a score to each unselected feature and sequentially selects the feature with the maximum score in the current iteration. The procedure for MIBFS using the scoring function $J_{ABC}(\cdot)$ is summarized in Algorithm 1. We now describe the scoring functions of five representative MIBFS algorithms as follows.

The Mutual Information Maximization (MIM) (Lewis, 1992) was the first work to use MI as a scoring function. It scores each feature independently of the others, which is simple but a limitation in itself. The scoring function is given by

$$J_{\text{MIM}}(X_k) = I(X_k;Y). \tag{9}$$

Algorithm 1 MI based Feature Selection via J_{ABC}

Input: Scoring function $J_{ABC}(\cdot)$;
$\Omega \leftarrow$ Set of M features;
$Y \leftarrow$ Label;
Output: the set S with the selected features.
1: Initialization: $S \leftarrow \{\emptyset\}$; $K \leq M$.
2: **repeat**
3: Choose the feature F that
4: $F = \underset{X \in \Omega \setminus S}{\operatorname{argmax}} J_{ABC}(X)$;
5: $S \leftarrow S \cup F$;
6: **until** $|S| = K$:

The minimal Redundancy Maximum Relevance (mRMR) (Peng et al., 2005) introduced an additional penalty term to reduce redundancy within the selected feature set S. The term is divided over the cardinality of S to adaptively vary the penalty. The scoring function is

$$J_{\text{mRMR}}(X_k) = I(X_k; Y) - \frac{1}{|S|} \sum_{X_j \in S} I(X_k, X_j). \tag{10}$$

The Joint Mutual Information (JMI) (Yang & Moody, 2000; Meyer et al., 2008) was proposed to increase complementary information between features. The scoring function is given by

$$J_{\text{JMI}}(X_k) = \sum_{X_j \in S} I(X_k, X_j; Y). \tag{11}$$

The Joint Mutual Information Maximization (JMIM) (Bennasar, Hicks, & Setchi, 2015) built upon JMI and attempted to alleivate the problem of overestimation of the feature significance, so as to optimize the relationship between relevance and redundancy. The scoring function is given by

$$J_{\text{JMIM}}(X_k) = \min_{X_j \in S} I(X_k, X_j; Y). \tag{12}$$

The Greedy Search Algorithm (GSA) (Brown et al., 2012) is a forward searching method that works by selecting, at each step, the candidate feature with the largest joint MI with the label. The scoring function is given by

$$J_{\text{GSA}}(X_k) = I(X_k, S; Y). \tag{13}$$

Along the way, many interesting works (Dai, Huang, Zhang, & Liu, 2024; Yin et al., 2024; Han, Hu, & Gao, 2024) have proposed to improve the performance of MIBFS. As an example, (Wang, Nie, & Huang, 2015) made use of conditional MI and attempted to minimize the global redundancy (i.e., the overall redundancy within the selected feature subset). (Wang et al., 2017) proposed a MI based term called independent classification information, aiming to strike a balance between redundancy and relevance. (Zhang & Parhi, 2018) introduced a term called uncertainty score, evaluated by conditional entropy, and this

term is minimized during feature selection, leading to a method called minimum uncertainty and feature sample elimination. More recently, (Gao, Hu, & Zhang, 2020) penalized the relevance between a candidate feature and the selected feature within the same class, leading to a method called min-redundancy and max-dependency. (Song et al., 2021) proposed bar bones particle swarm optimization to effectively estimate the value of MI during feature selection.

Overall, all these heuristics aim to find the optimal feature subset S^* in (1) via the criterion of Maximize Relevance with Minimum Redundancy (Peng et al., 2005), where Maximize Relevance corresponds to the requirement of maximizing $I(S;Y)$ and Minimum Redundancy is to reduce the size of S.

2.4 Information Content: UR, CR, Irrelevance

In feature selection, a feature may contain three types of information with respected to the label. They are Unique Relevance (UR), Conditional Relevance (CR) and Irrelevance.

The Unique Relevance (UR) of a feature X_k is defined as the unique relevant information which is not shared by any other features in Ω. Mathematically, UR can be calculated as the MI loss when removing that feature from Ω. By the chain rule for MI (Cover & Thomas, 2006), UR can be written as

$$\text{UR } (X_k) = I(\Omega;Y) - I(\Omega \backslash X_k;Y) = I(X_k;Y|\Omega \backslash X_k). \tag{14}$$

Features with non-zero UR are called unique relevant features (also known as strong relevant features in (Kohavi & John, 1997; Brown et al., 2012)).

The Conditional Relevance (CR) of a feature X_k to the label Y with respect to a feature set W can be written as

$$\text{CR} = I(X_k;Y|W). \tag{15}$$

Features with non-zero CR are called conditionally relevant features while features with zero CR are called conditional irrelevant features (Brown et al., 2012; Yu & Liu, 2004). We note that there is another definition called weak relevant feature in the literature (Kohavi & John, 1997; Brown et al., 2012). A conditional relevant feature X_k (i.e., $I(X_k;Y|W) > 0$) is equivalent to the weak relevant feature if it satisfies two conditions: (i) feature X_k has zero UR. (ii) $W \subseteq \{\Omega \backslash X_k\}$.

Irrelevance can be understood as the noise in the signal. Overfitting to the irrelevant aspects of the data will confuse the classifier, leading to decreased accuracy (John et al., 1994; Song, Ni, & Wang, 2011). Mathematically, we define irrelevance of feature X_k as follows,

$$\text{Irrelevance } (X_k) = H(X_k) - I(X_k;Y) = H(X_k|Y). \tag{16}$$

We note that a feature X_k can be completely irrelevant with respect to the label Y if $I(X_k;Y) = 0$.

There is another popular type of decomposition called partial information decomposition (PID) (Williams & Beer, 2010) which decomposes the total mutual information of a system into three parts: unique information, redundancy, synergy and a follow-up work (Bertschinger et al., 2014) attempts to quantify each term based on ideas from decision theory. We note that the definition of UR is similar to unique information, but is estimated differently.

2.5 Estimation of Mutual Information

MI is considered to be very powerful and, therefore, has inspired many works on its estimation. There are two basic approaches to estimate MI – parametric and non-parametric.

The parametric approach is a given form for the density function which assumes that the data are from a known family of distributions (e.g., Gaussian) and the parameters of the function (i.e., mean and variance) are then optimized by fitting the model to the data set (Walters-Williams & Li, 2009; Batina et al., 2011). As an example, (Belghazi et al., 2018) proposed Mutual Information Neural Estimation (MINE) based on dual representations of the KL-divergence and optimize its parameters via back-propagation. More recently, (Choi & Lee, 2021) introduced a regularized version of MINE which aims to enhance the stability of the original MINE while (Wang et al., 2021) attempted to improve MINE with an additional part of label smoothing.

The non-parametric approach utlizes the geometry of the underlying sample to estimate the local density. Examples include histogram-based estimator (Moddemeijer, 1999; Walters-Williams & Li, 2009), adaptive partitioning (Darbellay et al., 1999), kernel density estimator (Batina et al., 2011) and K nearest neighbor based estimator (Kraskov et al., 2004). Compared to the parametric approach, the non-parametric approach makes no assumption about the distribution of data and hence, is more practical.

In this paper, we estimate MI using the KSG estimator (Kraskov et al., 2004) which uses the K nearest neighbors of points in the dataset to detect structure in the underlying probability distribution. The main reason of choosing KSG estimator is two-fold: (i) In terms of performance, KSG estimator is well known for its superior performance. Several survey papers (Doquire & Verleysen, 2012; Khan et al., 2007) have done extensive experiments, suggesting that KSG estimator outperforms a bunch of parametric (e.g., Gaussian) and non-parametric (e.g., KDE estimator) approaches. Specifically, the parametric approaches often estimate the assumed joint densities from a limited number of samples, which is often infeasible in many practical settings. The KSG estimator is a nonparametric estimator, which is more practical for MIBFS where data samples are often limited. On the other hand, among non-parametric estimators, the KSG estimator is a better choice as its KNN structure is more stable and less affected by noise (Walters-Williams & Li, 2009).(ii) In terms of computational efficiency, the KSG estimator is significantly simpler than most approaches(e.g., neural network based approaches), which is more suitable for the intensive estimation of MI in the context of MIBFS. It is worth noting that the KSG estimator is not applicable when the random variable being studied is a mixture of continuous and discrete values. For that case, we can apply the mixed KSG estimator (Gao et al., 2017), which demonstrates good performance at handling mixed variables. We note that the features of all datasets studied in this paper are either purely discrete (real-valued) or continuous while all labels are purely discrete (real-valued) (see Table 3 rows 1 – 5). Therefore, we use the KSG estimator to compute MI quantities.

3. A New Criterion for MIBFS

In this section, we first decompose the goal of MIBFS (stated in (1)) into two parts: (i) Why Maximum MI?; (ii) Why Minimal Feature Subset? and answer these two questions in Section 3.1 and Section 3.2, respectively. In Section 3.3, we show the crucial role of UR in

selecting the optimal feature subset in (1). Next, in Section 3.4, we conduct experiments to uncover the fact that all studied MIBFS algorithms are underperforming in achieving the goal of MIBFS. Lastly, in Section 3.5, we motivate MRwMR-BUR as a new criterion for MIBFS and demonstrate that MRwMR-BUR could better achieve the goal of MIBFS.

3.1 Goal of MIBFS: Why Maximum MI?

We denote the dataset by $D = \{(\Omega^i, Y^i) : i = 1..N\}$, where Ω^i, Y^i, S^i denote all features, label and selected features for the i_{th} sample, respectively. Recall the goal of MIBFS is to identify the minimal feature subset S to maximize the MI with the label. The latter part of the goal can be written as

$$l^* = \max_{S \subseteq \Omega} I(S; Y). \tag{17}$$

By chain rule for MI, (17) can be rewritten as

$$l^* = \max_{S \subseteq \Omega} I(S; Y) = \max_{S \subseteq \Omega} H(Y) - H(Y|S). \tag{18}$$

The H(Y) is constant as it quantifies the uncertainty in Y and is not going to change during feature selection. Therefore, (18) can be further rewritten as

$$l^* = \max_{S \subseteq \Omega} -H(Y|S) = \max_{S \subseteq \Omega} \mathbb{E}_{sy} \left\{ \log p(Y|S) \right\}. \tag{19}$$

By the definition of conditional entropy, (19) can be approximated as

$$\begin{aligned} l^* &= \max_{S \subseteq \Omega} \mathbb{E}_{sy} \left\{ \log p(Y|S) \right\} \\ &\approx \max_{S \subseteq \Omega} \frac{1}{N} \sum_{i=1}^{N} \log p(y^i|S^i) = \max_{S \subseteq \Omega} \prod_{i=1}^{N} p(y^i|S^i). \end{aligned} \tag{20}$$

It can be seen that maximizing $I(S; Y)$ is equivalent to maximizing the conditional likelihood of the label Y given the selected feature subset S. The conditional likelihood is a well-studied statistical principle and maximizing conditional likelihood is the objective of may statistical models (e.g., Naive Bayes Classifier (Friedman, Geiger, & Goldszmidt, 1997)). Therefore, selecting the feature subset S that maximizes MI with the label can help to improve the classification performance.

3.2 Goal of MIBFS: Why Minimal Feature Subset?

Including more terms than necessary is essentially an overfitting issue defined in (Hawkins, 2004) and it may cause two problems:

(i) Efficiency. In terms of maximizing joint MI $I(S; Y)$, if K features can maximize the joint MI. Including more redundant features will cause extra computation and storage.

(ii) Overfitting to irrelevance. More importantly, more features may contain more dominant characteristics of other domains (Ding et al., 2005). These dominant characteristics of other domains are often known as irrelevance and overfitting to the irrelevance may confuse the classifier, leading to degraded performance (John et al., 1994; Song et al., 2011). The following proposition demonstrates that including extra features will increase the irrelevance presented in the selected feature subset.

Proposition 1. *Assuming that $H(X_k|S^*) \neq 0$, including an additional feature X_k to the optimal feature subset S^* in (1) will increase the irrelevance presented in the selected feature subset $\{X_k, S^*\}$.*

Proof. Since S^* is the optimal feature subset in (1), we have

$$I(S^*; Y) = I(S^*, X_k; Y). \tag{21}$$

By chain rule for MI, (21) can be derived as

$$H(S^*) - H(S^*|Y) = H(S^*, X_k) - H(S^*, X_k|Y).$$

After manipulation on both sides, we have

$$\begin{aligned} H(S^*, X_k|Y) - H(S^*|Y) &= H(S^*, X_k) - H(S^*) \\ &= H(X_k|S^*) \geq 0. \end{aligned} \tag{22}$$

□

Referring to our definition of irrelevance in (16), the irrelevance of feature subset $\{X_k, S^*\}$ is $H(S^*, X_k|Y)$ while the irrelevance of feature subset S^* is $H(S^*|Y)$. In (22), we show that, after including feature X_k, the irrelevance with respect to the label Y will increase as long as the entropy of feature X_k does not fully overlap with the joint entropy of S^* (i.e., $H(X_k|S^*) > 0$). Only when the entropy of feature X_k fully overlaps with the joint entropy of S^* (i.e., $H(X_k|S^*) = 0$), including feature X_k will not increase the irrelevance. However, it is unlikely to happen in practice.

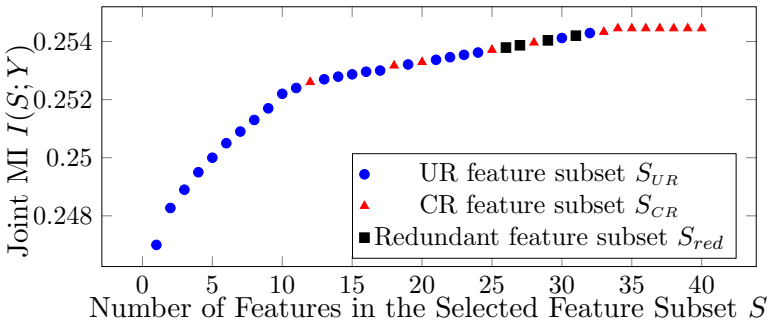
3.3 Goal of MIBFS: A Crucial Condition for Optimality

Recall the goal of MIBFS in (1) is to find the minimal feature subset with maximum MI with respect to the label (Brown et al., 2012). Several works (Yu & Liu, 2004; John et al., 1994) have pointed out that UR is a necessary condition for the optimal solution in (1). This can be simply proved as follows.

Proposition 2. *The optimal feature subset S^* in (1), which has maximum $I(S;Y)$ with minimum $|S|$, must contain all features with UR.*

Proof. Assume there exists a feature $X_k \in \Omega$ with non-zero UR. Suppose we have a feature subset $S \subseteq \Omega \backslash X_k$ which has maximum $I(S;Y)$ with minimum $|S|$. Since X_k has non-zero UR, we have $I(\Omega;Y) > I(\Omega \backslash X_k;Y)$ by definition. Therefore, $I(\Omega;Y) > I(S;Y)$ as $I(\Omega \backslash X_k;Y) \geqslant I(S;Y)$ given $S \subseteq \Omega \backslash X_k$. But this contradicts the initial assumption that $I(S;Y)$ is maximum. □

As shown in Prop. 2, it is compulsory for the optimal feature subset S^* to contain all features with UR. While at certain situations, S^* may also contain features with zero UR. For example, consider a feature subset T which contains all features with UR, it is possible to have another feature $X_m \notin T$ which contributes to higher joint MI (i.e., $I(X_m, T;Y) > I(T;Y)$). The reason for X_m's zero UR is that X_m's relevance with respect to Y overlaps with other features.



algorithm	redundancy rate (γ)
MIM	23.1% (9/39)
JMI	14.3% (5/35)
JMIM	21.1% (8/38)
mRMR	18.9% (7/37)
GSA	11.8% (4/34)

Figure 1: (left) Illustration of feature selection using GSA on the Sonar dataset (Dua & Graff, 2017). (right) Redundancy rates ($\gamma = |S_{red}|/|S_{sat}|$) for various MIBFS algorithms on the Sonar dataset. The numbers in parentheses are $|S_{red}|$ and $|S_{sat}|$, respectively.

3.4 Evaluation of MRwMR Based Algorithms

In this subsection, we evaluate the performance of five representative MIBFS algorithms (described in Section 2.3) in achieving the goal of MIBFS described in (1). Specifically, we aim to count how many redundant features are selected by each of them when the joint MI $I(S;Y)$ firstly saturates. We simulate their feature selection process using the Sonar dataset (Dua & Graff, 2017) and evaluate the variation of joint MI $I(S;Y)$ as more features are selected. Let S_{sat} be the selected feature subset when $I(S;Y)$ firstly reaches saturation. We further divide S_{sat} into two subsets as follows: $S_{sat} = \{S_{UR}, S_{ZUR}\}$. S_{UR} is the UR feature subset and contains selected features with non-zero UR. S_{ZUR} is the zero UR feature subset contains selected features with zero UR.

Since S_{UR} is necessarily a subset of the optimal subset S^* in (1), we only conduct exhaustive search on S_{ZUR} and find the largest feature subset S_{red} which can be removed without decreasing the joint MI. In such a manner, we evaluate the feature subset selected by various MIBFS algorithms. Mathematically, the problem is formulated as follows.

$$S_{red} = \arg\max f(\arg\max_{\bar{S}\subseteq S_{ZUR}} I(S_{UR}, S_{ZUR}\backslash\bar{S}; Y)), \tag{23}$$

where $f(A, B, \cdots) = (|A|, |B|, \cdots)$, $|A|$ represents the number of features in A. By obtaining S_{red}, we further divide S_{ZUR} into two subsets (i.e., $S_{ZUR} = \{S_{CR}, S_{red}\}$):

1. The redundant feature subset S_{red}: the maximal feature subset which can be removed from S_{ZUR} without decreasing the joint MI (i.e., $I(S_{UR}, S_{ZUR}; Y) = I(S_{UR}, S_{ZUR}\backslash S_{red}; Y)$).
2. The CR feature subset S_{CR}: the minimal feature subset which provides the maximum MI (joint with S_{UR}) with the label Y. The feature subset S_{CR} contains selected features which have CR with the label Y given S_{UR}.

To quantify the redundancy of a selected feature subset, we introduce a term called redundancy rate (γ), represented as

$$\gamma = |S_{red}|/|S_{sat}|. \tag{24}$$

In Fig. 1 (left), we evaluate the variation of joint MI $I(S;Y)$ as more features are selected by GSA. The performance of all selected MRwMR based algorithms summarized using redundancy rate is shown in Fig. 1 (right). A higher redundancy rate is undesirable as it indicates that more redundant features are selected.

MRwMR based algorithms	MRwMR-BUR based algorithms
$J_{\text{MIM}}(X_i) = I(X_i;Y)$	$J_{\text{MIM-BUR}}(X_i) = (1-\beta) \times I(X_i;Y) + \beta \times J_{\text{UR}}(X_i)$
$J_{\text{JMI}}(X_i) = \sum_{X_j \in S} I(X_i, X_j; Y)$	$J_{\text{JMI-BUR}}(X_i) = (1-\beta) \times \sum_{X_j \in S} I(X_i, X_j; Y) \times \frac{1}{\lvert S \rvert} + \beta \times J_{\text{UR}}(X_i)$
$J_{\text{mRMR}}(X_i) = I(X_i;Y) - \frac{1}{\lvert S \rvert} \sum_{X_j \in S} I(X_i, X_j)$	$J_{\text{mRMR-BUR}}(X_i) = (1-\beta) \times \big(I(X_i;Y) - \frac{1}{\lvert S \rvert} \sum_{X_j \in S} I(X_i, X_j)\big) + \beta \times J_{\text{UR}}(X_i)$
$J_{\text{JMIM}}(X_i) = \min_{X_j \in S} I(X_i, X_j; Y)$	$J_{\text{JMIM-BUR}}(X_i) = (1-\beta) \times (\min_{X_j \in S} I(X_i, X_j; Y)) + \beta \times J_{\text{UR}}(X_i)$
$J_{\text{GSA}}(X_i) = I(X_i, S; Y)$	$J_{\text{GSA-BUR}}(X_i) = (1-\beta) \times I(X_i, S; Y) + \beta \times J_{\text{UR}}(X_i)$

Table 1: The scoring functions of various MRwMR based feature selection algorithms and their corresponding MRwMR-BUR forms. Depending on the task of MRwMR-BUR, UR can be estimated differently, leading to two variants of MRwMR-BUR: (i) MRwMR-BUR-KSG; (ii) MRwMR-BUR-CLF.

In Fig. 1 (left), we observe that GSA can perform well at the beginning and select features with UR. However, as more features are selected, the joint distribution of the selected feature subset becomes more complex, and then GSA tends to select redundant features. Similarly, a non-negligible number of redundant features are also selected by other algorithms (see Fig. 1 (right)). We note that redundant features contribute nothing, but they do increase the size of the selected feature subset, undermining the objective of minimal feature subset in (1). Surprisingly, all of the representative algorithms studied select a non-negligible number of redundant features, which uncovers the fact that all of them are underperforming. Lastly, we note that we are able to find S_{red} on S_{ZUR} via an exhaustive search because the Sonar dataset is a relatively low dimensional dataset. In general, most real world datasets tend to contain more features and it is computationally infeasible to conduct such an exhaustive search.

3.5 The Proposed MRwMR-BUR Criterion

(1) Motivation for MRwMR-BUR. We point out that the heart of the problem is that features with UR are not prioritized during the selection process as all these algorithms follow the MRwMR criterion which lacks a mechanism to identify UR. Furthermore, based on our analysis of various datasets (see Table 3 row 5), features with UR usually make up a very small fraction of the total features and hence, is difficult to select them without explicitly targeting them. This motivates us to augment MRwMR and include the objective of boosting unique relevance (BUR), leading to a new criterion, called MRwMR-BUR.

(2) The MRwMR-BUR Criterion. The augmented MRwMR-BUR based algorithms are

$$J_{new}(X_i) = (1-\beta) \times J_{org}(X_i) + \beta \times J_{UR}(X_i), \tag{25}$$

where J_{org} is the original MRwMR based algorithm (e.g., MIM), J_{new} is corresponding MRwMR-BUR form and $J_{UR}(X_i)$ returns the UR of feature X_i. Depending on the task being addressed, MRwMR-BUR has two variants, termed **MRwMR-BUR-KSG** and **MRwMR-BUR-CLF**, which estimate UR differently. MRwMR-BUR-KSG directly estimates UR using (14) via a nearest neighbor based approach called KSG estimator and

	$\beta = 0$	$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.5$	$\beta = 0.9$
MIM-BUR-KSG	23.1% (9/39)	**18.9%** (7/37)	21.1% (8/38)	23.1% (9/39)	23.1% (9/39)
JMI-BUR-KSG	14.3% (5/35)	16.7% (6/36)	**11.8%** (4/34)	18.9% (7/37)	21.1% (8/38)
JMIM-BUR-KSG	21.1% (8/38)	**16.7%** (6/36)	18.9% (7/37)	23.1% (9/39)	23.1% (9/39)
mRMR-BUR-KSG	18.9% (7/37)	**14.3%** (5/35)	16.7% (6/36)	23.1% (9/39)	25% (10/40)
GSA-BUR-KSG	11.8% (4/34)	**6.25%** (2/32)	14.3% (5/35)	16.7% (6/36)	16.7% (6/36)

Table 2: Redundancy Rate ($\gamma = |S_{red}| \ / \ |S_{sat}|$) of MRwMR based Algorithms before ($\beta = 0$) and after Boosting UR (see (25)) on the Sonar dataset. All the MI quantities are estimated using the KSG estimator. The numbers in parentheses are $|S_{red}|$ and $|S_{sat}|$, respectively. The numbers in bold represent the lowest redundancy rate over different values of β.

is designed for three major tasks: (i) Classification Performance (i.e., higher classification accuracy). (ii) Feature Interpretability (i.e., a more precise feature subset to explore the hidden relationship between features and labels). (iii) Classifier Generalization (i.e., the selected feature subset generalizes well to various classifiers). MRwMR-BUR-CLF estimates UR via a classifier based approach, aiming the further improve the competitiveness of MRwMR-BUR for classification performance oriented tasks. More details of MRwMR-BUR-CLF are discussed in Section 5.

We note that the augmentation in (25) is slightly different for JMI as JMI is not bounded and the score increases as more features are selected. Therefore, we divide the original JMI by the size of the selected feature subset and include BUR. Moreover, the details for each algorithm with and without BUR are provided in Table 1. We denote the algorithm that extends XYZ as XYZ-BUR (e.g., MIM and MIM-BUR). Furthermore, if the UR is estimated via the KSG estimator, we further extend XYZ-BUR as XYZ-BUR-KSG. Likewise, XYZ-BUR-CLF indicates that the UR is estimated via the classifier based approach.

(3) MRwMR-BUR reduces the redundancy rate. In Table 2, we estimate all the MI quantities using the KSG estimator and present the redundancy rate before and after boosting UR using the same Sonar dataset in Section 3.4. We note that the MRwMR-BUR based algorithm degenerates to the original MRwMR form when $\beta = 0$ (see (25)). It can be seen that the redundancy rate is reduced after slightly boosting UR (e.g., $\beta = 0.1$). For example, the number of redundant features selected by GSA is reduced from 4 to 2 after boosting features with UR ($\beta = 0.1$), leading to a lower redundancy rate of 6.25%. On the other hand, when we heavily boost features with UR by increasing the value of β to 0.5 and 0.9, it causes an increase in the redundancy rate. We posit it is because that heavily boosting features with UR could severely destroy the mechanism of the original MRwMR based algorithm. Specifically, heavily boosting features with UR will only select the candidate feature with UR and overlook its CR with respect to the label Y given S.

(4) Complexity of MRwMR-BUR. In terms of the computational complexity, MRwMR-BUR based algorithms have the same complexity as the corresponding MRwMR based algorithms. This is because the value of UR only needs to be calculated once and can be used in subsequent calculations.

	Colon	Sonar	Madelon	Leukemia	Isolet	Gas sensor
Features	2000	60	500	7070	617	128
Instances	62	208	2600	72	1560	13874
Classes	2	2	2	2	26	6
Data Type	Discrete	Continuous	Continuous	Discrete	Continuous	Continuous
UR (%)	4.3%	28.3%	29.6%	34.9%	37.1%	2.34%

Table 3: Information of the six public datasets used in the experiments. The last row computes the percentage of features with UR among all features for each dataset.

4. Performance Evaluation

In Section 3.3, we motivate the idea of MRwMR-BUR and demonstrate that MRwMR-BUR could significantly reduce the redundancy rate. In this section, we estimate UR using the KSG estimator and compare the performance of each algorithm before and after boosting UR, so as to evaluate the MRwMR-BUR criterion. In Section 4.1, we describe the experiment setup. Next, in Section 4.2, we conduct the performance comparison and analyze the results.

4.1 Experiment Setup

(1) Experiment Details. To examine the performance of the proposed MRwMR-BUR criterion, we conduct experiments using six public datasets (Alon et al., 1999; Golub et al., 1999; Dua & Graff, 2017; Guyon et al., 2003; Alexander et al., 2012) (see descriptions in Table 3) and compare the performance of MRwMR-BUR-KSG (estimate UR via the KSG estimator) to MRwMR via four popular classifiers: Support Vector Machine (SVM) (Cortes et al., 1995), K-Nearest Neighbors (KNN) (Larose & Larose, 2014), Random Forest (RF) (Breiman, 2001) and Multilayer Perceptron (MLP) (Haykin, 1994). Five representative MRwMR based algorithms: MIM (Lewis, 1992), JMI (Yang & Moody, 2000), JMIM (Bennasar et al., 2015), mRMR (Peng et al., 2005) and GSA (Brown et al., 2012) are shortlisted for performance evaluation.

For each run, the dataset is randomly split into three subsets: training dataset (60%), validation dataset (20%), testing dataset (20%). We apply MRwMR and MRwMR-BUR-KSG (with $\beta = 0.1$) based algorithms on the same training dataset to select features and evaluate them using the same testing dataset. We compute the validation accuracy for gradually selecting up to k features, resulting in a vector of $[\theta_1, \theta_2, \cdots, \theta_k]$. Next, we shortlist the first n features which provide the highest validation accuracy and use them to compute the test accuracy. The test accuracy averaged over 20 runs are shown for RF (in rows 1 – 10), KNN (in rows 11 – 20), and SVM (in rows 21 – 30) in Table 6. The performance of MLP can be found at Table 7. Furthermore, the round-up average number of features chosen (i.e., the value in parentheses) is also an indication of the performance in terms of the objective of minimal feature subset in (1).

(2) Parameter Tuning. To ensure fair comparison, the parameters of all classifiers are tuned using the validation dataset via grid search and all algorithms share the same grid searching range and step size. Some key parameters are tuned as follows. (i) the

	KNN	SVM	RF
MM-BUR-KSG $\{S_1\}$	85.0%	82.2%	84.8%
MIM-BUR-KSG $\{S_1 \setminus \{\text{Hsa.8, Hsa.1132}\}\}$	83.2%	80.7%	84.1%
GSA-BUR-KSG $\{S_2\}$	85.5%	74.5%	86.3%
GSA-BUR-KSG $\{S_2 \setminus \{\text{Hsa.8, Hsa.1132}\}\}$	84.1%	73.8%	85.0%

Table 4: Test Accuracy of MIM-BUR-KSG and GSA-BUR-KSG with and without two important features (Hsa.8 and Hsa.1132) on the Colon dataset via KNN, SVM and RF. S_1 and S_2 represent the feature subset with maximum validation accuracy selected by MIM-BUR-KSG and GSA-BUR-KSG, respectively.

Dataset	Feature (Description)
Leukemia	U22376 (C-myb), M31523 (E2A), M69043 (MAD-3), U46751 (p62).
Colon	Hsa.3307 (Human Gps2 mRNA), Hsa.2598 (H.sapiens B-cam mRNA)

Table 5: A List of features identified by MRwMR-BUR-KSG which could contribute to better feature interpretability.

number of nearest neighbors K for KNN is tuned from 3 to 50 with step size of 2. (ii) the regularization coefficient c for SVM is chosen from $\{0.001, 0.01, 0.1, 1, 10\}$. (iii) the number of decision trees in the RF is chosen from $\{10, 15, ..., 100\}$. We directly estimate the UR of features using (14) via the KSG estimator. This is the same as how we estimate UR in Sections 3.2 and 3.3. Following the same notation, we denote the algorithm that extends XYZ as XYZ-BUR-KSG (e.g., MIM and MIM-BUR-KSG).

(3) Source Code. We have released the source code, which can be found at https://github.com/kentridgeai/MRwMR-BUR.

4.2 Performance Comparison

In Table 6, we compare the performance between MRwMR based algorithms and their corresponding MRwMR-BUR-KSG algorithms. The results are presented as average test accuracy $\pm$ standard deviation over 20 runs and the number in the parentheses represents the number of features chosen.

(1) Performance of MRwMR based Algorithms. In terms of MRwMR based algorithms, we observe that MIM tends to provide the worst performance in terms of average test accuracy and number of features required (e.g., see Colon dataset). We suspect it is because that MIM assumes features are independent from each other, leading to degraded performance. For other MRwMR based algorithms, JMI and GSA generally perform better than other MRwMR based algorithms (e.g., see Madelon dataset). This finding agrees with (Brown et al., 2012; Liu et al., 2018) as GSA is greedy in nature and JMI can increase the complementary information between features.

(2) Performance of MRwMR VS MRwMR-BUR-KSG. Comparing the performance of MRwMR to MRwMR-BUR-KSG, we observe that most of the MRwMR based algorithms improve their performance after BUR. As an example, the accuracy of GSA on Colon dataset using RF is increased from 84.7% to 86.5% (compare row 9 to row 10 in

Table 6), resulting an improvement of 2.1%. In addition, for the dataset that we have been able to obtain extremely high accuracy (i.e., the Gas sensor dataset with 99%+ accuracy), MRwMR-BUR can significantly reduce the number of features required while maintaining comparable performance. As an example, the number of features required for mRMR on the Gas sensor dataset using RF (compare row 3 to row 4 in Table 6) is decreased from 93 to 69, a reduction of 25.8%. We also note that, for several cases, MRwMR-BUR utilizes more features than its original counterpart. For example, comparing JMI to JMI-BUR-KSG using SVM on the Leukemia dataset (i.e., row 27 to row 28 in Table 6). We posit that is mainly due to the original algorithm. Specifically, the original algorithm may not highly rank features with UR. As a result, even if MRwMR-BUR prioritizes features with UR, it fails to select them, resulting in a larger subset of selected features. In this case, a larger β value could be helpful.

(3) Generalization to Other Classifiers. In addition to the performance using RF mentioned above, the superior performance of MRwMR-BUR-KSG tends to generalize to KNN and SVM as well. As an example, the accuracy of mRMR on the Colon dataset using KNN is increased from 82.3% to 84.7% after BUR (compare row 13 to row 14), resulting in an improvement of 3%. The number of features needed is decreased from 43 to 30, a reduction of 30.2%. Similar performance can also be observed using Multilayer Perceptron (MLP) (Haykin, 1994). We summarize its results in Table 7.

(4) Feature Interpretability. As compared to MRwMR, MRwMR-BUR-KSG helps to better explore hidden relationships between features and labels in two aspects:

(i) The feature subset selected by MRwMR-BUR-KSG based algorithms contains fewer number of features and excludes more noise from redundant or irrelevant features. This can be seen from above where MRwMR-BUR-KSG based algorithms select a much smaller number of features.

(ii) MRwMR-BUR slightly prioritizes features with UR, leading to a more precise feature subset for study. As a detailed example, MRwMR-BUR-KSG helps to identify two important features in the Colon dataset with 2000 gene features: Hsa.8 (Human mRNA for ORF) and Hsa.1132 (Human mRNA for hepatoma-derived growth factor). Both of these two features have a relatively lower MI, which cause them not to be selected by any of the five representative algorithms studied when those algorithms achieve the peak accuracy. However, both of them are features with UR, which can be identified and selected by MRwMR-BUR-KSG algorithms. This leads to a higher test accuracy on all classifiers studied (i.e., in Table 4, compare rows 1 and 3 to rows 2 and 4, respectively), suggesting the crucial role of these two features in diagnosis (tumoral and non-tumoral). We have discussed our findings with domain experts. The feedback is that such findings are useful which could provide medical staff with new insights and further advance relevant research. In the Table 5, we summarize similar features, i.e., features with UR that can lead to better diagnosis, but are often ignored by MRwMR algorithms due to their relatively low MI.

(5) Algorithm Complexity. We highlight that the value of UR only needs to be calculated once and can be used in subsequent calculations. Hence, the algorithm complexity after BUR is comparable to the original algorithm.

	Colon	Sonar	Madelon	Leukemia	Isolet	Gas sensor
			Random Forest (RF)			
1) MIM	83.7 ± 7.3 (48)	79.5 ± 3.4 (43)	71.5 ± 1.3 (86)	95.5 ± 4.3 (84)	85.1 ± 0.8 (142)	99.27 ± 0.13 (86)
2) MIM-BUR-KSG	**85.1 ± 6.1 (23)**	**80.1 ± 2.8 (39)**	**72.4 ± 1.5 (76)**	**96.5 ± 2.3 (45)**	**86.1 ± 1.5 (134)**	**99.43 ± 0.05 (85)**
3) mRMR	85.0 ± 5.1 (14)	79.4 ± 2.5 (46)	72.0 ± 0.8 (76)	96.1 ± 3.3 (76)	85.8 ± 0.3 (121)	99.44 ± 0.04 (93)
4) mRMR-BUR-KSG	**87.0 ± 3.1 (8)**	**81.0 ± 2.0 (39)**	**72.4 ± 0.3 (74)**	**96.5 ± 3.1 (61)**	**86.3 ± 0.5 (125)**	**99.46 ± 0.03 (69)**
5) JMI	72.8 ± 2.1 (31)	80.5 ± 3.1 (42)	**72.5 ± 0.3 (60)**	95.6 ± 1.3 (59)	87.0 ± 0.2 (130)	99.45 ± 0.06 (79)
6) JMI-BUR-KSG	**74.7 ± 6.0 (37)**	**81.0 ± 3.9 (46)**	71.7 ± 0.3 (61)	**97.1 ± 1.4 (63)**	**87.8 ± 0.5 (133)**	**99.46 ± 0.04 (65)**
7) JMIM	73.2 ± 4.7 (33)	79.6 ± 1.3 (46)	72.8 ± 0.5 (70)	95.6 ± 3.3 (58)	85.3 ± 0.9 (123)	99.44 ± 0.03 (86)
8) JMIM-BUR-KSG	**75.7 ± 2.9 (20)**	**80.6 ± 1.4 (36)**	**73.1 ± 0.4 (59)**	**95.9 ± 1.0 (59)**	**85.7 ± 0.3 (119)**	**99.48 ± 0.09 (75)**
9) GSA	84.7 ± 2.3 (19)	80.3 ± 3.1 (25)	73.0 ± 0.6 (64)	**95.6 ± 3.1 (60)**	86.2 ± 0.3 (118)	99.42 ± 0.04 (95)
10) GSA-BUR-KSG	**86.5 ± 1.5 (14)**	**81.7 ± 2.9 (27)**	**73.4 ± 0.4 (57)**	95.5 ± 1.3 (55)	**87.2 ± 0.3 (115)**	**99.47 ± 0.04 (85)**
			K-Nearest Neighbors (KNN)			
11) MIM	84.0 ± 5.1 (60)	82.6 ± 1.3 (24)	73.0 ± 0.8 (55)	95.3 ± 1.2 (44)	78.1 ± 0.9 (117)	99.04 ± 0.03 (85)
12) MIM-BUR-KSG	**85.2 ± 3.9 (40)**	**83.2 ± 1.1 (24)**	**74.5 ± 1.2 (49)**	**96.0 ± 1.5 (38)**	**78.7 ± 0.6 (129)**	**99.09 ± 0.06 (83)**
13) mRMR	82.3 ± 3.7 (43)	84.8 ± 2.0 (33)	75.3 ± 1.5 (59)	97.5 ± 0.7 (56)	80.8 ± 1.3 (123)	98.94 ± 0.02 (80)
14) mRMR-BUR-KSG	**84.7 ± 2.9 (30)**	**85.6 ± 1.3 (33)**	**76.0 ± 2.0 (65)**	**97.6 ± 0.8 (52)**	**81.9 ± 1.6 (109)**	**99.02 ± 0.09 (78)**
15) JMI	70.0 ± 2.5 (45)	83.2 ± 1.7 (45)	79.0 ± 1.5 (49)	95.4 ± 1.1 (44)	**79.6 ± 1.1 (127)**	99.08 ± 0.06 (70)
16) JMI-BUR-KSG	**70.4 ± 2.7 (40)**	**84.8 ±1.3 (42)**	**79.8 ± 1.2 (51)**	**96.3 ± 0.7 (44)**	79.3 ± 1.0 (127)	**99.10 ± 0.05 (63)**
17) JMIM	72.0 ± 2.3 (81)	**83.9 ± 2.1 (43)**	77.1 ± 0.6 (55)	95.6 ± 0.5 (32)	79.0 ± 0.7 (135)	**99.08 ± 0.06 (83)**
18) JMIM-BUR-KSG	**75.9 ± 2.7 (62)**	83.7 ± 1.3 (42)	**78.2 ± 1.1 (64)**	**95.7 ± 0.9 (33)**	**79.3 ± 1.2 (135)**	99.05 ± 0.03 (78)
19) GSA	84.8 ± 3.1 (70)	83.3 ± 1.7 (46)	77.5 ± 1.4 (51)	95.4 ± 1.2 (55)	79.7 ± 1.5 (130)	98.7 ± 0.06 (53)
20) GSA-BUR-KSG	**86.1 ± 2.4 (55)**	**84.4 ± 1.3 (38)**	**78.3 ± 1.5 (43)**	**95.8 ± 0.7 (54)**	**80.1 ± 1.1 (107)**	**99.0 ± 0.04 (53)**
			Support Vector Machine (SVM)			
21) MIM	81.0 ± 3.9 (65)	72.8 ± 1.3 (34)	61.2 ± 0.7 (33)	96.4 ± 1.1 (97)	88.0 ± 0.7 (138)	96.5 ± 0.07 (91)
22) MIM-BUR-KSG	**82.0 ± 2.4 (61)**	**73.6 ± 1.5 (35)**	**61.6 ± 0.5 (34)**	**97.6 ± 1.5 (94)**	**88.6 ± 1.0 (117)**	**96.6 ± 0.04 (91)**
23) mRMR	**83.0 ± 2.5 (33)**	75.0 ± 1.8 (44)	61.5 ± 0.9 (37)	97.5 ± 1.1 (73)	89.2 ± 1.1 (123)	96.5 ± 0.05 (95)
24) mRMR-BUR-KSG	82.8 ± 2.1 (40)	**75.3 ± 1.3 (40)**	**61.5 ± 0.9 (37)**	**97.5 ± 1.1 (73)**	**89.6 ± 1.3 (127)**	**96.8 ± 0.06 (95)**
25) JMI	73.1 ± 2.2 (15)	74.0 ± 1.1 (39)	61.0 ± 1.3 (33)	96.5 ± 1.4 (83)	**89.8 ± 1.8 (127)**	96.7 ± 0.06 (88)
26) JMI-BUR-KSG	**73.9 ± 2.9 (15)**	**74.2 ± 1.6 (37)**	**62.0 ± 1.2 (35)**	**96.8 ± 1.4 (86)**	89.4 ± 1.4 (144)	**96.8 ± 0.09 (89)**
27) JMIM	76.6 ± 2.9 (20)	73.1 ± 1.9 (40)	62.0 ± 0.7 (40)	96.7 ± 0.8 (92)	88.3 ± 1.6 (135)	96.5 ± 0.09 (95)
28) JMIM-BUR-KSG	**77.5 ± 2.1 (24)**	**73.1 ± 2.4 (36)**	**62.3 ± 0.9 (43)**	**97.1 ± 0.8 (93)**	**88.6 ± 1.7 (115)**	**96.6 ± 0.09 (96)**
29) GSA	73.6 ± 2.4 (33)	73.7 ± 1.4 (35)	**63.3 ± 1.2 (37)**	96.4 ± 1.1 (65)	89.1 ± 1.3 (132)	96.2 ± 0.04 (93)
30) GSA-BUR-KSG	**74.1 ± 2.7 (28)**	**74.0 ± 1.8 (30)**	63.1 ± 0.8 (32)	**97.0 ± 1.5 (53)**	**89.5 ± 1.7 (125)**	**96.4 ± 0.05 (91)**

Table 6: Performance comparison between MRwMR based algorithms and MRwMR-BUR-KSG based algorithms (with $\beta = 0.1$). The results are presented as follows: average test accuracy (%) ± standard deviation over 20 runs (%) (number of features required). The results are shown for RF (row 1 - row 10), KNN (row 11 - row 20) and SVM (row 21 - row 30). All the MI quantities (including UR) are estimated using the KSG estimator. The bold indicates higher performance between MRwMR and MRwMR-BUR-KSG.

	Colon	Sonar	Madelon	Leukemia	Isolet	Gas sensor
1) MIM	82.3 ± 2.3 (51)	74.4 ± 0.9 (30)	65.8 ± 0.9 (39)	96.8 ± 1.4 (68)	85.0 ± 0.9 (109)	99.1 ± 0.05 (80)
2) MIM-BUR-KSG	**84.1 ± 3.7 (48)**	**75.6 ± 1.1 (33)**	**69.3 ± 0.5 (34)**	**97.5 ± 0.8 (61)**	**88.8 ± 1.1 (89)**	**99.1 ± 0.06 (76)**
3) mRMR	83.5 ± 2.7 (31)	76.8 ± 2.0 (42)	63.3 ± 0.7 (33)	**97.3 ± 0.9 (71)**	87.1 ± 0.8 (113)	97.2 ± 0.03 (68)
4) mRMR-BUR-KSG	**83.8 ± 1.9 (34)**	**77.9 ± 1.8 (33)**	**65.8 ± 0.9 (35)**	97.1 ± 1.1 (61)	**87.7 ± 0.9 (110)**	**97.7 ± 0.04 (65)**
5) JMI	74.2 ± 1.1 (20)	77.2 ± 1.3 (40)	67.2 ± 1.5 (29)	96.2 ± 1.1 (60)	89.1 ± 0.9 (117)	97.5 ± 0.05 (77)
6) JMI-BUR-KSG	**73.7 ± 1.5 (25)**	**78.9 ± 1.5 (37)**	**68.0 ± 1.1 (27)**	**96.8 ± 0.9 (65)**	**89.4 ± 0.8 (101)**	**97.8 ± 0.06 (74)**
7) JMIM	77.2 ± 2.5 (18)	**76.9 ± 2.0 (33)**	65.8 ± 0.9 (41)	96.8 ± 0.9 (70)	**88.5 ± 1.5 (131)**	97.0 ± 0.06 (89)
8) JMIM-BUR-KSG	**77.9 ± 1.9 (20)**	76.2 ± 2.1 (39)	**66.2 ± 0.5 (39)**	**96.8 ± 0.7 (55)**	87.9 ± 1.1 (129)	**97.5 ± 0.05 (79)**
9) GSA	**74.2 ± 3.0 (31)**	75.0 ± 0.9 (36)	69.9 ± 1.0 (36)	96.6 ± 0.8 (55)	87.8 ± 0.9 (115)	96.8 ± 0.07 (71)
10) GSA-BUR-KSG	73.8 ± 1.5 (14)	**77.3 ± 1.9 (30)**	**69.7 ± 0.7 (31)**	**96.9 ± 0.5 (40)**	**88.9 ± 1.1 (120)**	**98.1 ± 0.06 (55)**

Table 7: Performance comparison between MRwMR and MRwMR-BUR using the MLP classifier. The experimental setup is the same as in Table 6

5. Adapting UR to Different Classifiers

In Section 4, we have shown the superior performance of MRwMR-BUR-KSG in accomplishing three major tasks: (i) Classification Performance (ii) Feature Interpretability (iii) Classifier Generalization. In this section, we propose a classifier based approach to estimate UR. This approach of estimating UR adapts UR to different classifier, further improving the competitiveness of MRwMR-BUR for classification performance oriented tasks.

In Section 5.1, we introduce the classifier based approach to estimate UR which could adapt UR to different classifiers. In Sections 5.2 and 5.3, we conduct experiments and compare the classification performance of the classifier based approach to estimating UR to that of using the KSG estimator and three popular classifier based feature selection method.

5.1 Estimate UR via the Classifier

By the chain rule for MI, the UR of feature X_k in (14) can be equivalently expressed as

$$I(X_k; Y|\Omega\backslash X_k) = H(Y|\Omega\backslash X_k) - H(Y|\Omega). \tag{26}$$

The term $H(Y|\Omega)$ on the R.H.S of (26) is constant for every candidate feature during the selection process. Therefore, boosting UR is equivalent to boosting $H(Y|\Omega\backslash X_k)$, which is a function of the likelihood $p(Y|\Omega\backslash X_k)$ and this likelihood can be estimated using a classifier $\mathbb{Q}$ with parameter θ. Assuming all samples are i.i.d, the estimated UR of feature X_k can be rewritten as

$$\mathrm{UR}(X_k) \equiv H(Y|\Omega\backslash X_k) = \mathbb{E}\left\{\log p(Y|\Omega\backslash X_k)\right\}, \tag{27}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} \log \mathbb{Q}(y^i|(\Omega^i\backslash x_k^i), \theta), \tag{28}$$

where N is the number of samples.

The reason for estimating UR via a classifier is two-fold:

1. It is known that the estimation of high-dimensional MI is challenging and arguably suffers from the curse of dimensionality (Bellman, 1966). Our approach provides an alternative way to estimate UR, which will be close to the true value as the number of samples N grows given the classifier $\mathbb{Q}$ is a consistent estimator (Lehmann & Casella, 2006).
2. We note that different classifiers may treat the UR of features differently due to their working mechanisms and assumptions. For example, MI quantifies a non-linear relationship between random variables and this non-linear relationship may not be of much help to linear classifiers (e.g., SVM with linear kernel). Estimating UR via a classifier attempts to adapt UR to different classifiers, further improving the competitiveness of MRwMR-BUR for classification performance oriented tasks.

5.2 Experiment Setup

We now compare the performance of MRwMR-BUR using the classifier based approach to estimate UR to that of using the KSG estimator via KNN, SVM and RF. We highlight that the UR is estimated using the classifier being tested and all estimated URs are first normalized from 0 to 1 using min-max normalization. Furthermore, estimating UR in such a manner changes MRwMR-BUR based algorithms to a classifier dependent method. Therefore, in addition to the performance of MRwMR-BUR using the KSG estimator, three popular classifier dependent feature selection methods: (i) Recursive Feature Elimination (RFE) (Guyon, Weston, Barnhill, & Vapnik, 2002), (ii) Backward Feature Elimination (BFE) (Kohavi & John, 1997) and (iii) Forward Feature Selection (FFS) (Marcano-Cedeño et al., 2010), are also used for performance evaluation.

The average test accuracy $\pm$ standard deviation over 20 runs and the corresponding number of features required (i.e., the value in the parentheses) are shown in Table 8 with $\beta = 0.1$. The CLF named algorithms estimate UR via the classifier based approach. Since KNN is a non-parametric classifier contains no parameters related to the feature importance, RFE is not applicable to KNN. Therefore, the results of RFE on KNN is not shown.

5.3 Performance Evaluation

(1) MRwMR-BUR-CLF VS MRwMR. In Table 8, we find that the classifier based approach further improves the performance of MRwMR based algorithms. For example, the accuracy of MIM on the Madelon dataset using RF is increased from 71.5% to 74.2% (compare row 1 in Table 6 to row 1 in Table 8), with an improvement of 3.8%. Similar trends can be observed using other classifiers. For example, the accuracy of MIM using KNN on the Madelon dataset is increased from 73.0% to 77.0% (compare row 11 in Table 6 to row 9 in Table 8), with an improvement of 5.5%. Note that the improvement is much higher than MRwMR-BUR-KSG, which improves the accuracy of MIM from 73.0% to 74.5% (compare row 11 to row 12 in Table 6). This verifies our goal of adapting UR to different classifiers so as to further improve the performance of MRwMR-BUR based algorithms.

(2) MRwMR-BUR-CLF VS Classified Dependent Feature Selection Methods. As compared to the classifier dependent methods, MRwMR-BUR-CLF also obtains a better performance. For example, the accuracy of GSA-BUR-CLF on the Colon dataset using KNN is 85.4%, which is 1.5% higher than BFE. Similarly, the accuracy of mRMR-BUR-CLF on the Isolet dataset using KNN is 1.8% higher than FFS.

	Colon	Sonar	Madelon	Leukemia	Isolet	Gas sensor
			Random Forest (RF)			
1) MIM-BUR-CLF	86.3 ± 2.9 (18)	80.3 ± 1.1 (33)	74.2 ± 0.8 (55)	98.4 ± 1.4 (37)	87.5 ± 1.0 (119)	99.47 ± 0.06 (70)
2) mRMR-BUR-CLF	**88.7 ± 1.6 (30)**	81.4 ± 1.0 (35)	73.1 ± 0.5 (54)	98.8 ± 1.6 (50)	88.3 ± 0.3 (127)	99.49 ± 0.03 (65)
3) JMI-BUR-CLF	75.5 ± 1.1 (34)	81.4 ± 2.0 (35)	72.3 ± 0.9 (58)	**99.2 ± 0.8 (62)**	87.9 ± 0.4 (119)	99.48 ± 0.03 (70)
4) JMIM-BUR-CLF	76.2 ± 1.9 (17)	81.0 ± 0.9 (43)	73.4 ± 0.3 (54)	99.2 ± 1.1 (70)	85.9 ± 0.8 (107)	99.49 ± 0.04 (70)
5) GSA-BUR-CLF	88.0 ± 0.9 (20)	**81.9 ± 1.4 (33)**	74.5 ± 0.3 (55)	99.0 ± 1.4 (59)	**88.8 ± 0.4 (96)**	**99.50 ± 0.06 (81)**
6) RFE	87.1 ± 1.4 (23)	80.9 ± 1.5 (31)	**74.6 ± 0.5 (51)**	98.8 ± 0.7 (42)	88.1 ± 0.5 (102)	99.49 ± 0.05 (68)
7) FFS	87.7 ± 0.8 (25)	81.2 ± 1.2 (36)	74.4 ± 0.3 (54)	98.9 ± 0.6 (48)	87.9 ± 0.5 (107)	99.43 ± 0.04 (63)
8) BFE	86.9 ± 1.1 (27)	80.8 ± 0.7 (39)	74.5 ± 0.3 (49)	98.7 ± 0.9 (53)	88.3 ± 0.5 (111)	99.46 ± 0.03 (72)
			K-Nearest Neighbors (KNN)			
9) MIM-BUR-CLF	86.3 ± 1.8 (41)	84.6 ± 0.7 (41)	77.0 ± 0.6 (63)	97.7 ± 0.4 (31)	80.1 ± 0.8 (123)	99.09 ± 0.05 (78)
10) mRMR-BUR-CLF	86.3 ± 1.9 (29)	84.8 ± 1.4 (30)	76.3 ± 1.4 (65)	98.5 ± 0.8 (46)	**82.6 ± 1.1 (107)**	99.17 ± 0.06 (71)
11) JMI-BUR-CLF	71.2 ± 1.6 (38)	84.5 ± 0.9 (46)	**79.9 ± 1.0 (57)**	98.8 ± 0.7 (34)	80.8 ± 1.3 (119)	**99.28 ± 0.06 (76)**
12) JMIM-BUR-CLF	77.2 ± 1.4 (51)	84.1 ± 0.9 (40)	79.3 ± 0.6 (52)	**99.2 ± 0.7 (28)**	80.1 ± 1.0 (121)	98.78 ± 0.03 (75)
13) GSA-BUR-CLF	**88.3 ± 1.5 (50)**	**85.4 ± 1.0 (44)**	78.4 ± 1.1 (40)	98.4 ± 0.9 (51)	81.0 ± 0.9 (107)	99.2 ± 0.05 (65)
14) FFS	86.8 ± 1.1 (33)	83.7 ± 1.2 (41)	78.8 ± 0.8 (57)	98.9 ± 0.6 (32)	81.3 ± 0.8 (105)	99.12 ± 0.04 (62)
15) BFE	87.4 ± 1.5 (37)	84.1 ± 0.8 (44)	79.2 ± 1.0 (61)	98.5 ± 0.9 (35)	81.1 ± 0.7 (109)	99.07 ± 0.07 (71)
			Support Vector Machine (SVM)			
16) MIM-BUR-CLF	82.4 ± 2.3 (59)	74.1 ± 0.6 (45)	62.8 ± 0.6 (40)	98.1 ± 0.9 (78)	89.1 ± 0.5 (104)	97.1 ± 0.04 (87)
17) mRMR-BUR-CLF	**84.2 ± 0.8 (34)**	**75.5 ± 0.8 (45)**	62.1 ± 0.7 (43)	**98.7 ± 0.7 (59)**	89.9 ± 0.7 (120)	96.4 ± 0.05 (83)
18) JMI-BUR-CLF	73.3 ± 1.6 (20)	75.4 ± 1.0 (39)	62.2 ± 0.6 (34)	98.0 ± 0.9 (75)	**90.1 ± 1.3 (117)**	97.1 ± 0.05 (92)
19) JMIM-BUR-CLF	79.5 ± 1.3 (26)	73.7 ± 1.4 (35)	62.5 ± 0.6 (42)	98.3 ± 0.6 (70)	89.4 ± 1.0 (129)	**97.3 ± 0.09 (93)**
20) GSA-BUR-CLF	76.2 ± 1.1 (29)	74.1 ± 1.1 (43)	63.4 ± 0.5 (36)	98.5 ± 0.9 (66)	90.1 ± 1.2 (122)	96.5 ± 0.03 (86)
21) RFE	83.8 ± 0.9 (33)	74.7 ± 1.5 (41)	63.5 ± 0.9 (35)	98.6 ± 0.6 (68)	89.7 ± 0.9 (118)	96.1 ± 0.1 (85)
22) FFS	83.1 ± 1.2 (31)	74.5 ± 1.2 (36)	63.8 ± 0.7 (37)	98.5 ± 0.7 (62)	89.4 ± 0.5 (109)	96.4 ± 0.07 (82)
23) BFE	83.4 ± 1.1 (35)	74.8 ± 1.0 (39)	**63.9 ± 0.6 (38)**	98.6 ± 0.8 (66)	89.6 ± 0.8 (119)	96.3 ± 0.06 (88)

Table 8: Performance of MRwMR-BUR-CLF based algorithms and three popular classifier dependent feature selection methods (RFE, FFS and BFE). The results are presented as: average test accuracy (%) ± standard deviation over 20 runs (number of features required).The bold indicates the best performance over all methods studied for a given classifier.

5.4 MRwMR-BUR-KSG VS MRwMR-BUR-CLF

Three factors are usually considered when choosing a feature selection method, i.e., (i) classification performance, (ii) feature interpretability and (iii) classifier generalization. Depending on the task being addressed, we need to apply different variants of MRwMR-BUR. For the classification performance oriented task, MRwMR-BUR-CLF should be used as it tends to provide better classification performance than MRwMR-BUR-KSG, as well as classifier dependent feature selection methods. If either of the other two factors (feature interpretability, classifier generalization) also needs to be considered, we recommend using MRwMR-BUR-KSG. This is because MRwMR-BUR-KSG is classifier independent and the selected feature subset is not tied to the bias of classifier. As a result, the feature subset

selected by MRwMR-BUR-KSG does not vary from classifier to classifier, and thus is generally more accurate in terms of feature interpretability. Furthermore, the feature subset is selected from an information-theoretic perspective and tends to generalize well to various classifiers, since the mechanisms of most classifiers are closely related to information theory.

6. Reflections

We now conclude the paper by presenting some reflections and suggestions for future work.

(1) Advantage of MRwMR-BUR over MRwMR. In this paper, we propose MRwMR-BUR as a new criterion to design MIBFS algorithms. The MRwMR-BUR criterion equips the existing MRwMR criterion with a mechanism to explicitly target the features with UR. The MRwMR based algorithms can be easily modified to the corresponding MRwMR-BUR form, without increasing the complexity. The results suggest that the MRwMR-BUR better achieves the goal of MIBFS by selecting fewer number of redundant features, leading to better classification performance, interpretability and generalization.

(2) Should We always Select Features with UR First? The results in Section 3.1 show that the minimal feature subset S^* in (1) must include all features with UR. Does this mean that we should only select features with UR first? The answer is no, especially when we have a feature budget, i.e., the number of features that can be selected is limited. This is because features with UR (e.g., $I(X_k;Y|\Omega\backslash X_k) >0$) may not contribute to higher relevance in the short term conditioned on $S \subseteq \Omega\backslash X_k$ (i.e., $I(X_k;Y|S) = 0$) as reducing conditional features (from $\Omega\backslash X_k$ to $S \subseteq \Omega\backslash X_k$) may decrease the conditional MI.

(3) Factors Affecting the Performance of MRwMR-BUR. We note that two main factors that affect the performance of MRwMR-BUR. **(i) The Value of** β which balances the original objective of maximizing relevance with prioritizing features with UR. From extensive experiments, we have consistently found that $\beta = 0.1$ is a good choice to balance UR and relevance. Alternatively, β can be thought of as a hyper-parameter and tuned via a validation dataset. A small value of β will help to slightly prioritize features with UR while a large value may undermine the selection mechanism of the original algorithm. Based on our experience, a value range of [0.05 - 0.3] seems to be a reasonable searching range for β. The theoretical determination of the optimal value β is clearly worth deeper thought. This could help to further minimize redundancy by prioritizing more important features. **(ii) The estimation of UR**. For the performance of MRwMR-BUR-KSG, we estimate UR using the KSG estimator as features of all datasets studied in this paper are either purely discrete (real-valued) or continuous while all labels are purely discrete (real-valued) (more details in Section 2.5). For the case of continuous-discrete mixture, we suggest using the mixed KSG estimator (Gao et al., 2017). As for MRwMR-BUR-CLF, it heavily relies on the classifier used. The UR estimated by classifier A may not be favored by classifier B as their mechanisms could be significantly different. We strongly recommend practitioners use the same classifier for both UR estimnation and classification tasks.

(4) Potentially Better MIBFS Algorithms. In this paper, we are not proposing a new MIBFS algorithm. Instead, we explore a new criterion for MIBFS algorithms that incorporate UR into the objective. Our results demonstrate that the MRwMR-BUR criterion has superior performance over the existing MRwMR criterion. We believe this new insight can inspire better MIBFS algorithms that optimally use UR.

References

Alexander, V., et al. (2012). Chemical gas sensor drift compensation using classifier ensembles, Sensors and Actuators B..

Alon, U., et al. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, *96*(12), 6745–6750.

Batina, L., et al. (2011). Mutual information analysis: a comprehensive study. *Journal of Cryptology*, *24*(2), 269–291.

Belghazi, M. I., et al. (2018). MINE: mutual information neural estimation..

Bellman, R. (1966). Dynamic programming. *Science*, *153*(3731), 34–37.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828.

Bennasar, M., Hicks, Y., & Setchi, R. (2015). Feature selection using joint mutual information maximisation. *Expert Systems with Applications*, *42*(22), 8520–8532.

Bertschinger, N., et al. (2014). Quantifying unique information. *Entropy*, *16*(4), 2161–2183.

Bommert, A., et al. (2020). Benchmark for filter methods for feature selection in high-dimensional classification data. *Computational Statistics & Data Analysis*, *143*, 106839.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Brown, G., et al. (2012). Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *Journal of Machine Learning Research*, *13*(1), 27–66.

Cai, J., et al. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, *300*, 70–79.

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, *40*(1), 16–28.

Choi, K., & Lee, S. (2021). Regularized mutual information neural estimation..

Cortes, C., et al. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Cover, T. M., & Thomas, J. A. (2006). *Elements of Information Theory, 2nd edition.* John Wiley & Sons.

Dai, J., Huang, W., Zhang, C., & Liu, J. (2024). Multi-label feature selection by strongly relevant label gain and label mutual aid. *Pattern Recognition*, *145*, 109945.

Darbellay, G. A., et al. (1999). Estimation of the information by an adaptive partitioning of the observation space. *IEEE Transactions on Information Theory*, *45*(4), 1315–1321.

Ding, C., et al. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, *3*(02), 185–205.

Doquire, G., & Verleysen, M. (2012). A comparison of multivariate mutual information estimators for feature selection. In *International Conference on Pattern Recognition Applications and Methods*, Vol. 2, pp. 176–185.

Dua, D., & Graff, C. (2017). UCI machine learning repository..

El-Hasnony, I. M., et al. (2020). Improved feature selection model for big data analytics. *IEEE Access*, *8*, 66989–67004.

El-Kenawy, et al. (2020). Novel feature selection and voting classifier algorithms for covid-19 classification in ct images. *IEEE Access*, *8*, 179317–179335.

Fan, Y., et al. (2024). Learning correlation information for multi-label feature selection. *Pattern Recognition*, *145*, 109899.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, *29*(2-3), 131–163.

Gao, S., et al. (2016). Variational information maximization for feature selection. In *Proceedings of the 30th Advances in Neural Information Processing Systems*, pp. 487–495.

Gao, W., Hu, L., & Zhang, P. (2020). Feature redundancy term variation for mutual information-based feature selection. *Applied Intelligence*, *50*(4), 1272–1288.

Gao, W., et al. (2017). Estimating mutual information for discrete-continuous mixtures. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 5986–5997.

Golub, T. R., et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, *286*(5439), 531–537.

Gu, X., et al. (2022). A feature selection algorithm based on equal interval division and conditional mutual information..

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*(Mar), 1157–1182.

Guyon, I., et al. (2003). Result analysis of the NIPS 2003 feature selection challenge..

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, *46*(1), 389–422.

Han, Q., Hu, L., & Gao, W. (2024). Feature relevance and redundancy coefficients for multi-view multi-label feature selection. *Information Sciences*, *652*, 119747.

Hassan, K. M., et al. (2022). Epileptic seizure detection in eeg using mutual information-based best individual feature selection..

Hastie, T., et al. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC.

Hawkins, D. M. (2004). The problem of overfitting. *Journal of Chemical Information and Computer Sciences*, *44*(1), 1–12.

Haykin, S. (1994). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.

He, J., et al. (2024). An oscillatory particle swarm optimization feature selection algorithm for hybrid data based on mutual information entropy. *Applied Soft Computing*, *152*, 111261.

Htun, H. H., Biehl, M., & Petkov, N. (2023). Survey of feature selection and extraction techniques for stock market prediction. *Financial Innovation*, *9*(1), 26.

John, G. H., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Proceedings of the 8th International Conference on Machine Learning*, pp. 121–129.

Jolliffe, I. (2005). Principal component analysis..

Khaire, U. M., & Dhanalakshmi, R. (2019). Stability of feature selection algorithm: A review..

Khan, S., et al. (2007). Relative performance of mutual information estimation methods for quantifying the dependence among short and noisy data. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, *76*(2), 026209.

Kim, B., et al. (2015). Mind the gap: A generative approach to interpretable feature selection and extraction. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 2260–2268.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence*, *97*(1-2), 273–324.

Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. *Physical Review E*, *69*(6), 066–138.

Lamba, R., Gulati, T., & Jain, A. (2022). A hybrid feature selection approach for parkinson's detection based on mutual information gain and recursive feature elimination..

Larose, D. T., & Larose, C. D. (2014). *Discovering knowledge in data: an introduction to data mining*. John Wiley & Sons.

Lehmann, E. L., & Casella, G. (2006). *Theory of point estimation*. Springer Science & Business Media.

Lewis, D. D. (1992). Feature selection and feature extraction for text categorization. In *Proceedings of the workshop on Speech and Natural Language*, pp. 212–217.

Liu, S., & Motani, M. (2020). Exploring unique relevance for mutual information based feature selection. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 2747–2752.

Liu, S., Yao, J., Zhou, C., & Motani, M. (2018). SURI: Feature selection based on unique relevant information for health data. In *Proceedings of IEEE International Conference on Bioinformatics and Biomedicine*, pp. 687–692.

Luque-Rodriguez, M., & et al (2022). Initialization of feature selection search for classification. *Journal of Artificial Intelligence Research*, *75*, 953–983.

Ma, X.-A., & Lu, K. (2024). Class-specific feature selection using neighborhood mutual information with relevance-redundancy weight. *Knowledge-Based Systems*, *300*, 112212.

Marcano-Cedeño, A., et al. (2010). Feature selection using sequential forward selection and classification applying artificial metaplasticity neural network. In *36th Conference on IEEE Industrial Electronics Society*, pp. 2845–2850. IEEE.

Meyer, P. E., et al. (2008). Information-theoretic feature selection in microarray data using variable complementarity. *IEEE Journal of Selected Topics in Signal Processing*, *2*(3), 261–274.

Moddemeijer, R. (1999). A statistic to estimate the variance of the histogram-based mutual information estimator based on dependent pairs of observations. *Signal Processing*, pp. 51–63.

Nguyen, X. V., et al. (2014). Effective global approaches for mutual information based feature selection. In *Proceedings of the 20th International Conference on Knowledge Discovery and Data Mining*, pp. 512–521.

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

Salem, O. A., et al. (2022). Fuzzy joint mutual information feature selection based on ideal vector..

Solorio-Fernández, S., et al. (2020). A review of unsupervised feature selection methods. *Artificial Intelligence Review*, *53*(2), 907–948.

Song, Q., Ni, J., & Wang, G. (2011). A fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE Transactions on Knowledge and Data Engineering*, *25*(1), 1–14.

Song, X.-f., et al. (2021). Feature selection using bare-bones particle swarm optimization with mutual information. *Pattern Recognition*, *112*, 107804.

Sun, L., et al. (2020). Adaptive feature selection guided deep forest for covid-19 classification with chest ct. *IEEE Journal of Biomedical and Health Informatics*, *24*(10), 2798–2805.

Sun, L., et al. (2014). Feature selection using mutual information based uncertainty measures for tumor classification. *Bio-medical Materials and Engineering*, *24*(1), 763–770.

Theng, D., & Bhoyar, K. K. (2024). Feature selection techniques for machine learning: a survey of more than two decades of research. *Knowledge and Information Systems*, *66*(3), 1575–1637.

Too, J., & Mirjalili, S. (2021). A hyper learning binary dragonfly algorithm for feature selection: A covid-19 case study. *Knowledge-Based Systems*, *212*, 106553.

Tripathi, S., Hemachandra, N., & Trivedi, P. (2020). Interpretable feature subset selection: A shapley value based approach. In *IEEE International Conference on Big Data*, pp. 5463–5472. IEEE.

Venkatesh, B., et al. (2019). A review of feature selection and its methods. *Cybernetics and Information Technologies*, *19*(1), 3–26.

Walters-Williams, J., & Li, Y. (2009). Estimation of mutual information: A survey. In *International Conference on Rough Sets and Knowledge Technology*, pp. 389–396. Springer.

Wang, D., Nie, F., & Huang, H. (2015). Feature selection via global redundancy minimization. *IEEE Transactions on Knowledge and Data Engineering*, *27*(10), 2743–2755.

Wang, J., et al. (2017). Feature selection by maximizing independent classification information. *IEEE Transactions on Knowledge and Data Engineering*, *29*(4), 828–841.

Wang, X., et al. (2021). Adaptive label smoothing for classifier-based mutual information neural estimation. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 1035–1040.

Williams, P. L., & Beer, R. D. (2010). Nonnegative decomposition of multivariate information..

Yang, H. H., & Moody, J. (2000). Data visualization and feature selection: New algorithms for nongaussian data. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 687–693.

Yin, T., et al. (2024). Exploiting feature multi-correlations for multilabel feature selection in robust multi-neighborhood fuzzy β covering space. *Information Fusion*, *104*, 102150.

Yu, L., & Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, *5*(Oct), 1205–1224.

Zebari, R., et al. (2020). A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *Journal of Applied Science and Technology Trends*, *1*(2), 56–70.

Zhang, Z., & Parhi, K. K. (2018). Muse: Minimum uncertainty and sample elimination based binary feature selection. *IEEE Transactions on Knowledge and Data Engineering*, *31*(9), 1750–1764.

Zhou, H., Wang, X., & Zhu, R. (2022). Feature selection based on mutual information with correlation coefficient. *Applied Intelligence*, *52*(5), 5457–5474.