

Searching for Interacting Features in Subset Selection *

Zheng Zhao and Huan Liu[†]
Department of Computer Science & Engineering,
Arizona State University

September 21, 2007

Abstract

The evolving and adapting capabilities of robust intelligence are best manifested in its ability to learn. Machine learning enables computer systems to learn, and improve performance. Feature selection facilitates machine learning (e.g., classification) by aiming to remove irrelevant features. Feature (attribute) interaction presents a challenge to feature subset selection for classification. This is because a feature by itself might have little correlation with the target concept, but when it is combined with some other features, they can be strongly correlated with the target concept. Thus, the unintentional removal of these features may result in poor classification performance. It is computationally intractable to handle feature interactions in general. However, the presence of feature interaction in a wide range of real-world applications demands practical solutions that can reduce high-dimensional data while perpetuating feature interactions. In this paper, we take up the challenge to design a special data structure for feature quality evaluation, and to employ an information-theoretic feature ranking mechanism to efficiently handle feature interaction in subset selection. We conduct experiments to evaluate our approach by comparing with some representative methods, perform a lesion study to examine the critical components of the proposed algorithm to gain insights, and investigate related issues such as data structure, ranking, time complexity, and scalability in search of interacting features.

Keywords: Feature Interaction, Feature Selection, Search, Data Structure, and Classification

*Some preliminary experimental results of this work were reported in a conference paper at IJCAI 2007. This version provides detailed discussions of the employed data structure, presents extended theoretical analysis, and reports comprehensive experimental evaluations of key components of this work. For example, the role of data structure and its details, the effect of ranking, and two different ranking mechanisms with additional, complementary data sets.

[†]Corresponding author: Huan Liu, huan.liu@asu.edu. Brickyard Suite 501, 699 South Mill Avenue, Tempe, AZ, 85287, U.S.A. 480-727-7349 (voice) 480-965-2751 (fax)

1 Introduction

The rapid advance of computer technology and the ubiquitous use of the Web have provided unparalleled opportunities for humans to expand capabilities in production, services, communications, and research. In this process, immense quantities of high-dimensional data are accumulated challenging state-of-the-art machine learning techniques to efficiently produce useful results. Machine learning can benefit significantly from using only relevant data in terms of learning performance (e.g., predictive accuracy and time) and learned results such as improved comprehensibility. A widely applied technique for finding relevant data is *feature selection*, which studies algorithms of finding relevant features among many extant ones. Feature selection finds its pervasive application in many real-world domains: ranging from computational biology, biomedicine, text processing, image analysis, to Web services. Successful applications of feature selection also bring about new challenges, one of which is to search for *interacting* features that often function together and is elusive to efficient solutions. In the case of high-throughput microarray data, for instance, finding interacting features is to search for a small number of pertinent genes from hundreds of thousands of ones, reflecting the immune response mechanism involving both antigen presentation and immunoproteasome pathways. The overarching need for finding interacting features goes beyond the current methods to keep pace with data of increasing dimensionality. The new demands motivate us toward fundamental research in the design and development of novel and integrated methods of intelligent search for interacting features.

The high dimensional data poses a challenge to learning tasks such as classification. Given training data $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \subseteq (X, Y)$, a classification algorithm learns to produce a mapping $h : X \rightarrow Y$, which maps an observation $x \in X$ to its classification label $y \in Y$ [1]. Usually classification algorithms are designed to achieve an optimal or near optimal solution according to some performance metrics (e.g., accuracy, 0-1 loss and hinge loss) to ensure that a meaningful mapping is obtained in the learning process. Some representative classification algorithms include: Decision Trees, Naïve Bayes, k -Nearest Neighbor, Support Vector Machines, Bagging, and Boosting [2, 3, 4, 5, 6]. In the presence of many irrelevant features, classification algorithms tend to overfit training data [7], as too many parameters (corresponding to the number of features) in a learning model would allow the model to adjust to specific random factors in the training data which are not common in the full data [8]. Once overfitting happens, the gap between the estimated and the true accuracy becomes big, and the performance of the classifier can deteriorate significantly. Overfitting poses a serious problem especially for learning on data with high dimensionality [9]. In order to find probably approximately

correct (PAC) hypotheses¹, PAC learning theory [10, 11] gives a theoretic relationship between the number of instances (data points) needed in terms of the size of hypothesis space and the number of dimensions. For example, a binary data set with binary classes has a hypothesis space of $O(2^{2^d})$, where d is the dimensionality, it would require $O(2^d)$ instances to learn a probably and approximately correct hypothesis, which is usually unlikely to obtain in real-world applications when the dimensionality of the data is huge.

Various studies show that features can be removed without performance deterioration [12, 13, 14]. Feature selection is one effective means to remove irrelevant features [15, 7, 16]. The task of a feature selection algorithm is to remove irrelevant features based on the given data $X = (x_i)_{i \in [n]}$ and their labels $Y = (y_i)_{i \in [n]}$, where data point x_i is a vector with d features, and find a feature subset $\{F_{j_1}, F_{j_2}, \dots, F_{j_r}\}$, where $r < d$ and j_i ($i \in \{1 \dots r\}$) are indexes of features (F_1, \dots, F_d) , such that with dimensionally reduced data, a learning algorithm can achieve similar or better performance. To find the smallest k relevant features out of d features requires an exponentially large search space ($O(2^d)$, $0 \leq k \leq d$) [17, 18]. Researchers often resort to various approximation methods to determine relevant features (e.g., correlation between individual features and the class) [19, 20]. One commonly used approach is called sequential forward selection (SFS): First, the best feature is selected from d features; next, the second best features is selected from the remaining $d - 1$ features and combined with the previously selected features; then repeat until adding a new feature cannot improve some performance measure or all features are selected.

However, a single feature may be considered irrelevant based on its correlation with the class, but it may become very relevant if combined with other features. The unintentional removal of these features can result in the loss of useful information and thus may cause poor classification performance. This is studied in [22] as attribute interaction. For example, MONK1 [23] is a data set in which some features interact. There are six features in MONK1 and the target concept of MONK1 is: $(A_1 = A_2)$ or $(A_5 = 1)$. Here A_1 and A_2 are two interacting features. Considered individually, the correlation between A_1 and the class Y (similarly for A_2 and Y) is zero, measured by mutual information [24]. Hence, A_1 or A_2 is irrelevant when each is individually evaluated. However, if we combine A_1 with A_2 , they are strongly relevant in defining the target concept. An intrinsic character of feature interaction is its irreducibility [25], i.e., a feature could lose its relevance due to the absence of any other feature interacting with it.

¹The classification error probability is bounded by some value (i.e., approximately correct) with certain confidence specified by a probability (i.e., probably correct) [3].

Existing efficient feature selection algorithms usually assume feature independence [12, 19]. Because of the irreducible nature of feature interactions, these algorithms cannot select interacting features such as A_1 and A_2 in MONK1. Others attempt to explicitly address feature interactions by finding some low-order interactions (2- or 3-way). For example, methods are proposed in [26] to address feature interaction with Cartesian product for Naïve Bayes Classifier. In [22], the authors suggest to use interaction gain as a practical heuristic for detecting attribute interaction. Using interaction gain, their algorithms can detect if datasets have 2-way (one feature and the class) and 3-way (two features and the class) interactions. They further provide in [25] a justification of the interaction information, replace the notion of ‘high’ and ‘low’ in [22] with statistical significance, and illustrate the significant interactions in the form of interaction graph. In [20], the authors proposed an algorithm (FCBF) that sequentially searches predominant features by evaluating predominant correlations - close to a 3-way interaction: a F_i - F_j correlation is considered together with F_i - Y and F_j - Y correlations. It is reported to be effective and efficient.

To address the feature interaction in subset selection both effectively and efficiently, we design and implement an efficient approach to dealing with feature interactions in subset selection. Feature interactions can be implicitly handled by a carefully designed feature evaluation metric and a search strategy with a specially designed data structure, which together take into account interactions among features when performing feature selection. In the following, we show an example to the challenge of feature interaction in subset selection; we review feature relevance and introduce feature interaction, and propose a feature scoring metric based on data consistency; develop a filter algorithm named INTERACT to select relevant features while implicitly exploring feature interaction; and present an extensive empirical study in comparison with some representative algorithms followed by further analysis and discussion.

2 Motivating Examples

We select some synthetic data sets with known feature interactions and examine how some existing feature selection algorithms fare in finding interacting features. We apply four feature selection algorithms to the synthetic data sets: FCBF (selecting features using normalized mutual information) [20], CFS (selecting features using correlation based measure and Best-First search [27]) [19], ReliefF (selecting features by hypothesis maximization [13]) [28], and FOCUS (selecting features using consistency based measure and exhaustive search) [17]. These algorithms can be found in WEKA [29]. Four synthetic

data sets are used to examine how various algorithms deal with known feature interactions in feature selection. The first data set is Corral [30], having six boolean features A_0, A_1, B_0, B_1, I, R . The class Y is defined by $Y = (A_0 \wedge A_1) \vee (B_0 \wedge B_1)$ and features A_0, A_1, B_0, B_1 are independent of each other. Feature I is irrelevant to Y and its values have a uniform random distribution; and feature R is correlated with Y 75% of the time and is redundant. The other three training data sets are of MONKs [23]. Their target concepts are: (1) MONK1, $(A_1 = A_2)$ or $(A_5 = 1)$; (2) MONK2, exactly two of $A_1 = 1, A_2 = 1, A_3 = 1, A_4 = 1, A_5 = 1, A_6 = 1$; and (3) MONK3, $(A_5 = 3$ and $A_4 = 1)$ or $(A_5 \neq 4$ and $A_2 \neq 3)$ (5% class noise added to the training data).

Results on the synthetic data sets of the four feature selection algorithms are presented in Table 1. For Corral, all four algorithms remove the irrelevant feature I , but only FOCUS removes the redundant feature R . Features A_0, A_1 and B_0, B_1 interact with each other to determine the class label of an instance. CFS, FCBF and ReliefF cannot remove R because R is strongly correlated (75%) with Y . For the three MONKs data sets, we provide results for the full data in the table. ReliefF missed A_1, A_5 for MONK2, A_4 for MONK3 using the full data. As seen in Table 1, FCBF and CFS perform similarly for MONK1, and FCBF finds more features than CFS for MONK2 and MONK3. FOCUS can handle feature interaction when selecting features. However, as an exhaustive search algorithm, FOCUS is impractical because finding moderately high-order interactions can be too expensive, as $\sum_{i=1}^m \binom{d}{i}$ can be too large, even when dimensionality d is moderately large.

3 Relevance, Interaction and Data Consistency

One goal of feature selection is to remove all *irrelevant* features. In the following, we first define feature relevance as in [30]. Let \mathbf{F} be the full set of features, F_i be a feature, $S_i = \mathbf{F} - \{F_i\}$, and $P(Y|S)$ denote the conditional probability of class Y given a feature set S . The relevance of a feature can be formalized below.

Definition 1 (Feature Relevance) *A feature F_i is relevant if and only if*

$$\exists S'_i \subseteq S_i, \text{ such that } P(Y|F_i, S'_i) \neq P(Y|S'_i). \quad (1)$$

Otherwise, feature F_i is said to be irrelevant.

Definition 1 suggests that a feature can be relevant only when its removal from a feature set will reduce its prediction power. From Definition 1, it can be shown that a feature is relevant due to two reasons: (1) it is strongly correlated with the target concept; or (2) it forms a feature subset with other features and the subset is strongly correlated with the target concept. If a feature is relevant because of the second reason, there exists feature interaction. Feature interaction is characterized by its irreducibility [25]. A k th-order feature interaction can be formalized next.

Definition 2 (k th-order Feature Interaction) \mathbf{F} is a feature subset with k features F_1, F_2, \dots, F_k . Let \mathcal{C} denote a metric that measures the relevance of the class label with a feature or a feature subset. Features F_1, F_2, \dots, F_k are said to interact with each other if and only if: for an arbitrary partition $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \dots, \mathcal{F}_l\}$ of \mathbf{F} , where $l \geq 2$ and $\mathcal{F}_i \neq \phi$, we have

$$\forall i \in [1, \dots, l], \quad \mathcal{C}(\mathbf{F}) > \mathcal{C}(\mathcal{F}_i) \quad (2)$$

Identifying either relevant features or a k th-order feature interaction requires exponential time complexity. Hence, definitions 1 and 2 cannot be directly applied to identifying relevant or interacting features if the dimensionality of a data set is high. Some heuristics or approximations are necessary for developing viable algorithms. Many efficient feature selection algorithms identify relevant features based on the evaluation of the correlation between the class and a feature (or a current, selected feature subset). Representative measures used for evaluating feature relevance include: distance measures [28, 31], information measures [14], and consistency measures [17, 32], to name a few. Using these measures, feature selection algorithms usually start with an empty set and successively add “good” features to the selected feature subset, the so-called sequential forward selection (SFS) framework as described earlier. Under this framework, features are deemed relevant mainly based on their individually high correlations with the class, and some relevant interacting features of high order may be removed [19, 33, 34] because the irreducible nature of feature interaction cannot be attained by SFS. Revisiting the MONK1 problem, a feature subset selection algorithm of SFS starts with the empty subset S , and adds one good feature at a time to S . Since neither A_1 nor A_2 can be selected into S with the absence of the other, there will be no chance for features A_1 and A_2 to be evaluated together for their relevance.

Recall that finding high-order feature interaction using relevance (Definitions 1 and 2) entails exhaustive search of all feature subsets. In order to avoid the resulting exponential time complexity, we derive a feature scoring metric based on the consistency hypothesis proposed in [17] to approximate

the relevance measure as in Definition 1. With this metric, we design a filter algorithm that can deal with feature interaction in subset selection.

Let D be a data set of m instances, $D = \{d_1, d_2, \dots, d_m\}$, and \mathbf{F} be the feature space of D with n features, $\mathbf{F} = \{F_1, F_2, \dots, F_n\}$. We give the following definitions.

Definition 3 (Inconsistent Instances) *If two instances d_i, d_j in D have the same values except for their class labels, d_i and d_j are inconsistent instances or the two matching instances are inconsistent.*

Definition 4 (Inconsistent-Instances Set) *For $\mathcal{D} \subseteq D$, \mathcal{D} is an inconsistent-instances set if and only if $\forall d_i, d_j \in \mathcal{D}, i \neq j$, either d_i and d_j are inconsistent or they are duplicate. \mathcal{D} is a **maximal** inconsistent-instances set, if and only if $\forall d \in D$ and $d \notin \mathcal{D}$, $\mathcal{D} \cup \{d\}$ is not an inconsistent-instances set.*

Definition 5 (Inconsistency Count) *Let \mathcal{D} be an inconsistent-instances set with k elements d_1, d_2, \dots, d_k , and y_1, y_2, \dots, y_t are the class labels of \mathcal{D} , we partition \mathcal{D} into t subsets S_1, S_2, \dots, S_t by the class labels, where $S_i = \{d_j | d_j \text{ has label } y_i\}$. The inconsistency count of \mathcal{D} is:*

$$\text{inconsistencyCount}(\mathcal{D}) = k - \max_{1 \leq i \leq t} \|S_i\| \quad (3)$$

Definition 6 (Inconsistency Rate) *Let $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_p$ denote all maximal inconsistent-instances sets of D . The inconsistency rate (ICR) of D is*

$$\text{ICR}(D) = \frac{\sum_{1 \leq i \leq p} \text{inconsistencyCount}(\mathcal{D}_i)}{m} \quad (4)$$

Definition 7 (Consistency Contribution) or c-contribution *Let $\mathbf{\Pi}$ denote the projection operator which retrieves a subset of columns from D according to the feature subset. The c-contribution of feature F_i for \mathbf{F} is defined as*

$$\begin{aligned} \text{c-contribution}(F_i, \mathbf{F}) &= \text{ICR}(\mathbf{\Pi}_{\mathbf{F}-\{F_i\}}(D)) \\ &\quad - \text{ICR}(\mathbf{\Pi}_{\mathbf{F}}(D)) \end{aligned} \quad (5)$$

It is easy to verify that the inconsistency rate is monotonic in terms of the number of features, i.e., $\forall S_i, S_j, S_i \subseteq S_j \Rightarrow ICR(\mathbf{\Pi}_{S_i}(D)) \geq ICR(\mathbf{\Pi}_{S_j}(D))$ [32]. Hence, *c-contribution* of a feature is always a non-negative number with the zero meaning no contribution. *C-contribution* of a feature F_i is a function of $\mathbf{F} - \{F_i\}$, where \mathbf{F} is the set of features for D . *C-contribution* of a feature is an indicator about how significantly the elimination of that feature will affect consistency. For example, *c-contribution* of an irrelevant feature is zero. *C-contribution* can be considered as an approximation of the metric specified in Definition 1 by using inconsistency rate as an approximation of $P(Y|S)$, the conditional probability of class Y given a feature set S .

The monotonic property of inconsistency rate suggests that the backward elimination search strategy should best fit *c-contribution* in feature selection. That is, one can start with the full feature set and successively eliminate features one at a time based on *c-contributions*. Backward elimination allows every feature to be evaluated with the features it may interact with. Hence, backward elimination with *c-contribution* is suitable for finding interacting features. However, backward elimination using inconsistency rate or *c-contribution* has two problems. The first one is that it is *very costly* as it needs to calculate inconsistency rate for each potentially removable feature. As in the work of FOCUS [17] and ABB [32], FOCUS relies on exhaustive search and ABB resorts to complete search. It is impractical to do so when the dimensionality is reasonably large, which separates this work from FOCUS and ABB. We will design a specific data structure next in order to achieve efficient calculation of *c-contribution* in our proposed algorithm. The second problem is that *c-contribution* is sensitive to which feature is selected first in *c-contribution* calculation, the so-called *the feature order problem*. This is because features evaluated first for their consistency are more likely to be eliminated first, particularly in the early stage of feature elimination.

Solutions to the two problems will enable *c-contribution* to be possibly used in building an efficient algorithm of backward elimination that can handle interacting features in feature selection. We present our solutions and the algorithm INTERACT next.

4 INTERACT: Algorithm of Eliminating Irrelevant Features

We first present our solutions that form two pillar components for the algorithm INTERACT, i.e., efficient update of *c-contribution* and ranking features, next discuss the algorithm in detail, and then provide time complexity analysis.

4.1 Efficient update of c -contribution

C -contribution relies on the calculation of inconsistency rate. We exploit the monotonicity of inconsistency rate to accelerate the calculation of c -contribution so that we can reduce n to 1 time of data access. We can show that the following two properties are true after a feature f_i is eliminated from a set $\{f_1, \dots, f_i, \dots, f_n\}$ where $i = 1, 2, \dots, n$ given the monotonicity of inconsistency rate: (I) all *inconsistent-instances sets* are still inconsistent as removing a feature will not make any inconsistent-instances set consistent; and (II) each maximal *inconsistent-instances set* will be either of equal size or bigger. If the eliminated feature is irrelevant, the maximal inconsistent-instances set will remain unchanged; if the eliminated feature is relevant, the maximal set will increase its size since the removal results in more inconsistent instances.

Based on these two properties, we design a hashing mechanism to efficiently calculate c -contribution by avoiding accessing the data set whenever a feature is removed for the recalculation of c -contribution. In other words, with the hashing mechanism, we realize the objective of visiting the data only once to in the process of executing the algorithm. The hashing mechanism is depicted in Figure 1. Each instance is inserted into the hash table using the values of those features in S_{list} as the hash key, where the i th entry corresponds to an inconsistent-instances set with m_i instances, and S_{list} contains the features that are not yet eliminated. In the beginning, S_{list} has the full set of feature. Instances with the same feature value will be hashed into the same entry of the hash table. The information about the labels is recorded. Thus each entry in the hash table corresponds to a maximal inconsistent-instance set of $\Pi_{S_{list}}(D)$. The inconsistency rate of $\Pi_{S_{list}}(D)$ can therefore be obtained by simply scanning the hash table. Property (I) says that in order to generate an entry in the hash table for the new feature list, $S_{list'}$ (after eliminating a feature), it is not necessary to scan the data again, but only the current hash table (shown in Figure 1 (a) and (b)). Property (II) suggests that after each iteration of elimination, the number of entries of the hash table for new S_{list} should decrease as the number of feature is reduced (shown in Figure 1 (b)). Hence, the hashing data structure eliminates the need for repeated access of data and allows for efficient update of c -contribution for *iterative* feature elimination.

4.2 Dealing with the feature order problem

We now consider the feature order problem in applying c -contribution. Without any information about features, randomly selecting a feature for potential elimination seems a sensible way. However, some analysis tells us that doing so can result in the removal of a relevant feature according to c -contribution

calculation. Intuitively, using c -contribution to measure a feature’s usefulness depends on the number of remaining $(n-k)$ features. For a given feature out of n features, its c -contribution generally decreases when n increases. Consider two values of n , 3 and 2001. A relevant feature’s c -contribution in the case of $n = 3$ can be better captured than that in the case of $n = 2001$. This is because in the latter, the remaining 2000 features will likely be sufficient to maintain data consistency. Hence, the more features we have (as in the early stage of feature elimination), the more likely that the first feature considered for elimination will be chosen according to its c -contribution.

If we can remove irrelevant features first, we will likely retain the most relevant ones in the remaining subset of selected features. Assuming that a set of features can be divided into subset $S1$ including all the relevant features, and subset $S2$ containing all the irrelevant ones. By considering to remove features in $S2$ first, features in $S1$ are more likely to remain in the final set of selected features. Since we do not have prior knowledge of $S1$ and $S2$, we apply a heuristic to rank individual features using *symmetrical uncertainty* (SU) in an descending order such that the (heuristically) most relevant feature is positioned at the beginning of the list. SU is often used as a fast correlation measure to evaluate the relevance of individual features [19, 20]. This heuristic attempts to increase the chance for a strongly relevant feature to remain in the selected subset. Let $H(X)$ and $H(X, Y)$ denote entropy [24] and joint entropy respectively, and $M(X, Y) = H(Y) + H(X) - H(X, Y)$ be the mutual information measuring the common information shared between the two variables X and Y . SU between the class label Y and a feature F_i is:

$$SU(F_i, Y) = 2 \left[\frac{M(F_i, Y)}{H(F_i) + H(Y)} \right] \quad (6)$$

This ranking heuristic cannot guarantee that the interacting features be ranked high. For MONK1, for example, $SU(A_1, Y) = SU(A_6, Y) = 0$. Either one can be evaluated first for its c -contribution. Since $c\text{-contribution}(A_1, \mathbf{F}) > c\text{-contribution}(A_6, \mathbf{F})$, A_6 is eliminated. We will experimentally examine the effect of ranking in Section 5.3.

4.3 Handling Interacting Features in Subset Selection

The above solutions pave the way for c -contribution to be used in practical selection of interacting features. We present an algorithm, INTERACT, that efficiently searches for interacting features. It is a filter algorithm that employs backward elimination to remove those features with no or low c -contribution. The details are shown in Algorithm 1.

Given a full set of n features and a class attribute Y , INTERACT finds a feature subset S_{best} for

Algorithm 1: INTERACT

```

Input:  $F_1, F_2, \dots, F_n$ , the full feature set;
          $Y$ , the class label;
          $\delta$ , a predefined threshold;
Output:  $S_{best}$ , the best subset;
1 %Ranking;
2  $S_{list} = \text{NULL}$ ;
3 for  $i=1$  to  $n$  do
4   | calculate  $SU_{F_i, y}$  for  $F_i$ ;
5   | append  $F_i$  to  $S_{list}$ ;
6 end
7 order  $S_{list}$  in descending values of  $SU_{i, y}$ ;
8 %Feature Elimination;
9  $counter = n$ ;
10 repeat
11 |  $F = S_{list}[counter]$ ;
12 |  $p = c\text{-contribution}(F, S_{list})$ ;
13 | if  $p \leq \delta$  then
14 |   | remove  $F$  from  $S_{list}$ ;
15 | end
16 |  $counter = counter - 1$ ;
17 until  $counter = 0$  ;
18  $S_{best} = S_{list}$ ;
19 return  $S_{best}$ ;

```

the class concept. The algorithm consists of two major parts. In the first part (lines 1-7), the features are ranked in descending order based on their SU values. In the second part (lines 8-18), features are evaluated one by one starting from the end of the ranked feature list. $S_{list}[i]$ returns the feature in the position i of the list, S_{list} . If $c\text{-contribution}$ of a feature is less than δ , the feature is removed, otherwise it remains in the list. And the counter is decreased and pointed to the next unchecked feature preceding the F in the ranked feature list (line 11). The algorithm continues until all features in the list are checked. δ is a predefined threshold ($0 < \delta < 1$). Features with their $c\text{-contribution} < \delta$ are considered immaterial and thus removed. A large δ is associated with a high probability of removing relevant features. In this work, a feature's relevance is defined by its $c\text{-contribution}$: the higher value of $c\text{-contribution}(F, S_{list})$ indicates that F is more relevant. We set $\delta = 0.0001$ if not otherwise mentioned. The parameter can also be tuned using the standard cross validation technique.

Simplified backward elimination. In (line 11-15), we use a *simplified backward elimination* search strategy. The normal backward elimination search strategy works as follows: in order to remove one out of n features, it needs to calculate n values of $c\text{-contribution}$ and remove the feature with the smallest $c\text{-contribution}$; to remove k features, we need to repeat the above process k times. That is,

we need to calculate *c-contribution* for $O(kn)$ times, which is very expensive. *C-contribution* allows a feature to be evaluated with all features it potentially interacts with. Therefore, if a feature is not redundant or irrelevant, its *c-contribution* is likely large. Using this clue, we develop the simplified backward elimination search strategy: since the n features are ranked, in order to remove k out of n features, we start from the end of the ranked list of features to check if a feature’s *c-contribution* is below δ : if it is, the feature is removed, otherwise, it is retained. Hence, we only need to calculate *c-contribution* $O(n)$ times, instead of $O(kn)$ times as in the normal backward elimination.

4.4 Time complexity of INTERACT

The first part of the algorithm has a linear time complexity of $O(nm)$, where n is the number of features and m is the number of instances of a given data set. For the second part of the algorithm, the calculation of a feature’s *c-contribution* using a hash table takes also $O(nm)$. For n features, the time complexity of INTERACT is $O(n^2m)$. This is the worst case analysis. Its average time complexity is less because (1) we only use the hash table of current S_{list} in the calculation of *c-contribution*, and (2) the number of the entries in the hash table decreases after each iteration. If we assume it decreases to an α percentage of the initial size, where $0 < \alpha < 1$. Then, in n iterations, the overall time complexity of INTERACT is $O(nm(1 - \alpha^{n+1})/(1 - \alpha))$ since:

$$\sum_{i=1}^n \alpha^{i-1} nm = nm \sum_{i=1}^n \alpha^{i-1} = nm(1 - \alpha^n)/(1 - \alpha).$$

In other words, INTERACT can be expected to be comparable with heuristic algorithms such as FCBF [20].

5 Empirical Study

We now empirically evaluate the performance of INTERACT in search of interacting features. For the four synthetic data sets with known feature interaction (Table 1), INTERACT finds all and only the relevant features using the default $\delta = 0.0001$. In this section, we first describe the experiment setup, then report the standard empirical study in terms of the number of selected features and predictive accuracy using selected features comparing with some representative algorithms as baselines. In addition, we conduct a lesion study by removing one of them from INTERACT at a time to specifically evaluate the effects of the two solutions we present in Section 4. At the end, we provide the

run time and scalability results to verify our time complexity analysis given in the previous section.

5.1 Experiment setup

In the experiments, we choose four representative feature selection algorithms for comparison purposes. They are FCBF [20], CFS [19], ReliefF [28], and FOCUS [17]. They are briefly discussed in Section 2, and can be found in the WEKA environment [29]. INTERACT is also implemented in the WEKA’s framework and available for research purposes upon request. The experiments are conducted in the WEKA environment.

Twenty seven (27) benchmark data sets are used in which 23 data sets were investigated in [35] and are available from the UCI ML Repository [36]. Among the 4 additional benchmark data sets, ‘musk2’, ‘USCensus90’, and ‘internet-ads’ are from the UCI ML Repository, and the ‘45×4026+2C’ data is from [37]. The last 4 data sets have either a relatively large number of instances or high dimensionality. The information about the data sets is summarized in Table 2. The 27 data sets are divided into 3 groups as follows. The 23 data sets² used in [35] are divided into 2 groups based on their results about feature interaction: (1) 16 of them without obvious feature interaction, and (2) the remaining 7 with feature interaction. The 3rd group is composed of the 4 additional large data sets with either a larger number of features or a larger number of instances. The speed and scalability of INTERACT will be presented in Section 5.4. The last data set of Group 3 (‘internet-ad’) has a relatively large number of features (1558) and a large number of instances (3279).

For each data set, we run all 5 feature selection algorithms and obtain selected feature subsets of each algorithm. For data sets containing features with continuous values, if needed, we apply the MDL discretization method [38] (available in WEKA). We remove all index features if any. In order to evaluate whether the selected features are indeed good, we apply two effective classification algorithms C4.5 [2] and a linear Support Vector Machine (SVM) [4]³ (both available from Weka) before and after feature selection and obtain prediction accuracy by 10-fold cross-validation (CV) [3]. Although C4.5 itself evaluates features (one at a time), its performance is sensitive to the given sets of features (e.g., its accuracy rates on MONK1 are 88.88% and 75.69% for the 3 selected features (A_1, A_2 , and A_5) and for all 6 features, respectively). For ReliefF, we use 5 neighbors and 30 instances throughout the experiments as suggested in [31] and 0 is set as the threshold for removing irrelevant features. All

²The additional 3 data sets are MONKs data. We consider them synthetic data and discuss them separately earlier.

³Since Naive Bayes Classifier (NBC) assumes conditional independence between features [39], selecting interacting features or not has limited impact on it. Our experimental results of NBC conform to the analysis.

experiments were conducted on a PENTIUM IV 2.4G PC with 1.5GB RAM. Because of FOCUS’s exponential time complexity, we only provide those results of FOCUS obtained in 3 hours of dedicated use of the PC. INTERACT, FCBF, CFS and ReliefF all complete their runs in seconds. This is consistent with our understanding and expectation of these algorithms.

5.2 Results of Selected Features and Accuracy

We report below two sets of standard results.

Number of selected features. Table 3 presents the numbers of features selected by the five algorithms. All algorithms significantly reduced the number of features in many cases (e.g., from 56 to as few as 6 for ‘lung-cancer’). The average numbers of selected features for the 27 data sets are 11 (INTERACT), 9.37 (FCBF), 7.48 (CFS), 117.48 (ReliefF), and 231.48 (Full Set); and group-wise averages are shown in the table. That is, the average number of features is reduced from 231.48 to around tens after feature selection. For some data sets, FOCUS did not finish after 3 hours (indicated by NA in the table). For the 4 synthetic data sets with known relevant features (Table 1), INTERACT selected only the relevant ones. Next we examine the effect of this reduction on accuracy and we would like to see if the reduction causes any change in accuracy.

Predictive accuracy Tables 4 and 5 evince predictive accuracy rates obtained by the 10-fold cross validation using C4.5 and linear SVM. Results are also organized in three groups. For each group, average accuracy and win/loss numbers are given for easy comparison between the methods.

For the 16 data sets of **Group 1**, we observe that INTERACT performs competitively comparing with the other 4 algorithms. We present *average accuracy* and *number of selected features* (Table 3) in pairs to facilitate the comparison in these two dimensions. **C4.5:** (84.84, 8.56) for INTERACT, (84.12, 5.25) for FCBF, (84.09, 4.05) for CFS, (84.66, 12.69) for ReliefF, (82.54, 9.07) for FOCUS, and (84.77, 15.63) for the Full Set; and **SVM:** (84.19, 8.56) for INTERACT, (83.79, 5.25) for FCBF, (82.89, 4.05) for CFS, (84.74, 12.69) for ReliefF, (81.86, 9.07) for FOCUS, and (84.83, 15.63) for the Full Set. All feature selection algorithms can still achieve comparable accuracy rates as that of using the Full Set set of features. FOCUS did not finish for ‘mushroom’ and ‘adult’ in 3 hours, so there is no result for the two data sets.

For **Group 2** of 7 data sets with known interactions, the average accuracy and number of selected features are: **C4.5:** (81.30, 12.14) for INTERACT, (74.83, 7.14) for FCBF, (75.30, 9.43) for CFS, (76.07, 19.86) for ReliefF, (69.43, 8.2) for FOCUS, and (77.64, 26.14) for the Full Set; and **SVM:**

(79.15, 12.14) for INTERACT, (73.21, 7.14) for FCBF, (77.85, 9.43) for CFS, (74.38, 19.86) for ReliefF, (67.35, 8.2) for FOCUS, and (77.28, 26.14) for the Full Set. For both classifiers, the reduction of features by INTERACT leads to the results better than or comparable with *using all features*: average accuracy 81.30% (INTERACT) vs. 77.64% (Full Set) for C4.5, and 79.15% (INTERACT) vs. 77.28% (Full Set) for SVM. Comparing INTERACT with other feature selection algorithms, INTERACT performs consistently better for C4.5 with better average accuracy. For SVM, INTERACT performs well comparing with other feature selection algorithms with better average accuracy. One exception is the ‘soy-large’ data for the result of SVM. We notice that the data set has 35 features, 306 instances, and 19 classes (Table 2); INTERACT identifies 13 features (Table 3) - the smallest number of selected features (FCBF also selected 13 features). We surmise that it may be too easy for the inconsistency measure to be satisfied with a small feature subset when each class has a small number of instances. In sum, for both classifiers, in general, INTERACT can help achieve better or similar accuracy, and hence, INTERACT is *effective in search of interacting features*.

For **Group 3** of 4 data sets, the average accuracy and number of selected features are: **C4.5**: (94.42, 18.75) for INTERACT, (91.31, 29.75) for FCBF, (91.28, 16) for CFS, (93.32, 707.5) for ReliefF, no result for FOCUS, and (92.32, 597) for the Full Set; and **SVM**: (95.09, 18.75) for INTERACT, (94.66, 29.75) for FCBF, (95.21, 16) for CFS, (94.86, 707.5) for ReliefF, no result for FOCUS, and (95.39, 1454.25) for the Full Set. In short, INTERACT performs comparatively well.

Additional experimental results specific to various designs and solutions of INTERACT are presented in the next two sections.

5.3 Sensitivity study

In Sections 4.1 and 4.2, we presented two solutions to the two problems for using consistency measures, i.e., *costly update* in feature selection with a large number of instances, and *failure to appropriately order features* for backward elimination when there exists an inordinate number of features. In the following, we conduct experiments by removing one solution or component at a time to observe the effect of each of our solutions. First, we study the effect of ranking features and compare the information-theoretic ranking with another ranking mechanism. Then we investigate the effect of our hashing data structure.

5.3.1 The effect of feature ranking

INTERACT ranks features before backward elimination of features. As a part of the lesion study, we remove the ranking component from INTERACT to form a version $\text{INTERACT}_{\setminus R}$. The comparative results are shown in Table 6 and 7, which indicate that ranking is effective. In particular, among those 5 data sets whose accuracy results are significantly different, the average difference is as big as 21.83% ($= 75.00 - 44.17\%$) for ‘lung-cancer’. The average accuracy difference is very distinct for Group 3. The average numbers of selected features for the 27 data sets are similar, 11 (INTERACT) and 11.59 ($\text{INTERACT}_{\setminus R}$).

INTERACT ranks features using Symmetric Uncertainty (SU). Clearly, using SU is a heuristic and other feature ranking methods can also be used in INTERACT. Tables 8 and 9 record the results of using ReliefF as the feature ranking method in INTERACT for C4.5 and SVM, respectively. The results suggest that the two feature ranking methods are similar in terms of accuracy. Clearly, different ranking algorithms can have disparate biases which can result in different orders of features.

5.3.2 The effect of the data structure

We devised a hashing data structure to speed up the time consuming calculation of consistency contribution in search of a feature for elimination among the remaining features. Here we examine how effective the data structure is by comparing the run time of INTERACT with that of $\text{INTERACT}_{\setminus D}$ which does not employ the hashing data structure. Since data size is a determining factor for run time, we first reorganize the 27 data sets according to their sizes for convenient visual inspection: each data set’s size is approximated by $n * m$ - number of features multiplied by number of instances without considering feature types such as nominal or continuous.

Figure 2 (a) shows the results for the 19 data sets with run time less than 1 second. When data sets are small, INTERACT can take more time than $\text{INTERACT}_{\setminus D}$. This is due to the required overhead for the hashing data structure. With the increase in data size, the run time of $\text{INTERACT}_{\setminus D}$ increases from the lowest 0.02 second to 0.5 second; and the run time of INTERACT remains relatively stable (between 0.08 and 0.20 second).

Figure 2 (b) includes the results for the 8 data sets with run time more than 1 second. For the first 3 data sets ‘segment’, ‘kr-vs-kp’, and ‘mushroom’, INTERACT and $\text{INTERACT}_{\setminus D}$ took (1.02, 2.27), (1.31, 6.97), and (0.58, 2.80), respectively. For the remaining 5 data sets, the run time of INTERACT keeps below 149 seconds (for data ‘internet-ad’), while the run time of $\text{INTERACT}_{\setminus D}$ can be as high

as 1889 seconds (for data ‘internet-ad’). Figure 2 (c) plots the run results in logarithmic scale.

In sum, the run time difference between INTERACT and $\text{INTERACT}_{\setminus D}$ is more pronounced for large data sets. With the hashing data structure, we are able to constrict the time complexity within a manageable range. We compare run times about INTERACT with the selected feature selection algorithms and study the scalability issue next.

5.4 Run time comparison and scalability

Table 10 records the run time for each feature selection algorithm for the three groups of data. Except for FOCUS, all algorithms finished their runs within 3 minutes. For the data sets in Group 1, algorithms can be ordered based on the average run time in an ascending order: CFS, FCBF, Relief, INTERACT, and FOCUS, where CFS and FCBF are the fastest. For the data sets in Group 2, the algorithms are ordered as FCBF, CFS, Relief, INTERACT, and FOCUS. CFS and FCBF are the fastest, and FOCUS is the slowest if it can finish the run in 3 minutes. For the data sets in Group 3, Relief and FCBF are the fastest, followed by INTERACT, then by CFS, and FOCUS could not finish within the given time.

Scalability check In order to compare INTERACT with other three algorithms (FOCUS is not included in this experiment) on scalability with an increasing number of instances, we pick 5 data sets with a relatively large number of instances or a large number of features from Figure 2 (b,c). The results are shown in Figure 3. The curves are obtained using an increment of 10% of data. INTERACT scales well comparing with the 3 known scalable algorithms (FCBF, CFS, and ReliefF) - mostly linear patterns. One interesting observation is that for the ‘mushroom’ data, ReliefF works the slowest. Given that ReliefF calculates feature weights using only 30 instances sampled from the training data, its performance is supposed to be invariable to the increasing number of instances. A closer look at ReliefF reveals that for each sampled instance, ReliefF still needs to search all instances in the training data for its nearest neighbors, thus its run time increases with the increasing number of instances. This can be verified also in the results for ‘adult’ and ‘USCensus90’ when comparing ReliefF with FCBF and CFS in Figure 3.

6 Conclusions

The success of many feature selection algorithms allows us to tackle challenging real-world problems. Many applications inherently demand the selection of interacting features. In this work, we show

the challenges of studying feature interaction in subset selection, and propose to search for interacted features using consistency contribution to measure feature relevance. We investigate the key issues that hindered the use of consistency measures in the past, and develop an efficient algorithm, INTERACT that implicitly handles feature interaction and efficiently selects relevant features. The key to our success is the two singular solutions: (1) we design a special hashing data structure so that INTERACT avoids the repeated scanning of a data set by taking advantage of some intrinsic properties of the consistency contribution measure, resulting in a significant speed-up of the algorithm; and (2) INTERACT employs symmetrical uncertainty as a heuristic to rank features, overcoming the feature ordering problem. The efficiency and effectiveness of INTERACT is demonstrated through theoretical analysis as well as extensive experiments comparing INTERACT with representative feature selection algorithms using both synthetic and real-world data sets. Special experiments are designed to verify the effects of the two solutions. Experimental results show that INTERACT can (1) effectively reduce a large number of features, (2) retain competitive accuracy, (3) achieve good time performance, and (4) scale well with the increasing number of instances. Our current work is to explore more competing feature-quality measures for searching for interacting features in order to further advance this line of research.

References

- [1] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. John Wiley & Sons, New York, 2001.
- [2] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [3] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.
- [4] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., 1995.
- [5] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, pp. 123–140, 1996.
- [6] Y. Freund and R. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Computer Systems and Science*, vol. 55, no. 1, pp. 119 – 139, 1997.
- [7] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

- [8] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2001.
- [9] J. Friedman, “On bias, variance, 0/1-loss, and the curse-of-dimensionality,” *Data Mining and Knowledge Discovery*, vol. 1, no. 1, 1997.
- [10] L. Valiant, “A theory of the learnable,” *Communications of the Association for Computing Machinery*, vol. 27, pp. 1134–1142, 1984.
- [11] M. Kearns and U. Vazirani, *An Introduction to Computational Learning Theory*. The MIT Press, 1994.
- [12] M. Dash and H. Liu, “Feature selection for classification,” *Intelligent Data Analysis: An International Journal*, vol. 1, no. 3, pp. 131–156, 1997.
- [13] R. Gilad-Bachrach, A. Navot, and N. Tishby, “Margin based feature selection - theory and algorithms,” in *ICML '04: Twenty-first international conference on Machine learning*. ACM Press, 2004.
- [14] F. Fleuret, “Fast binary feature selection with conditional mutual information,” *J. Mach. Learn. Res.*, vol. 5, pp. 1531–1555, 2004.
- [15] A. L. Blum and P. Langley, “Selection of relevant features and examples in machine learning,” *Artificial Intelligence*, vol. 97, pp. 245–271, 1997.
- [16] H. Liu and L. Yu, “Toward integrating feature selection algorithms for classification and clustering,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 491–502, 2005.
- [17] H. Almuallim and T. G. Dietterich, “Learning boolean concepts in the presence of many irrelevant features,” *Artificial Intelligence*, vol. 69, no. 1-2, pp. 279–305, 1994.
- [18] D. Koller and M. Sahami, “Toward optimal feature selection,” in *Proceedings of the Thirteenth International Conference on Machine Learning*, 1996, pp. 284–292.
- [19] M. A. Hall, “Correlation-based feature selection for discrete and numeric class machine learning,” in *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, pp. 359–366.
- [20] L. Yu and H. Liu, “Feature selection for high-dimensional data: a fast correlation-based filter solution,” in *Proceedings of the twentieth International Conference on Machine Learning*, 2003, pp. 856–863.

- [21] M.-Y. Park and T. Hastie, “Regularization path algorithms for detecting gene interactions,” web, July 2006.
- [22] A. Jakulin and I. Bratko, “Analyzing attribute dependencies,” in *Proc. of Principles of Knowledge Discovery in Data (PKDD)*, ser. LNAI, N. Lavrač, D. Gamberger, H. Blockeel, and L. Todorovski, Eds., vol. 2838. Springer-Verlag, Sept. 2003, pp. 229–240. [Online]. Available: citeseer.ist.psu.edu/jakulin03analyzing.html
- [23] S. Thrun and et al, “The monk’s problems: A performance comparison of different learning algorithms,” Carnegie Mellon University, Tech. Rep. CMU-CS-91-197, 1991.
- [24] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley, 1991.
- [25] A. Jakulin and I. Bratko, “Testing the significance of attribute interactions,” in *ICML ’04: Twenty-first international conference on Machine learning*. ACM Press, 2004.
- [26] M. Pazzani, “Searching for dependencies in bayesian classifiers,” In: *Learning from Data: AI and Statistics V*. Springer-Verlag, 1996.
- [27] L. Xu, P. Yan, and T. Chang, “Best first strategy for feature selection,” in *Proceedings of the Ninth International Conference on Pattern Recognition*, 1988, pp. 706–708.
- [28] I. Kononenko, “Estimating attributes : Analysis and extension of RELIEF,” in *Proceedings of the European Conference on Machine Learning, April 6-8*, F. Bergadano and L. De Raedt, Eds. Catania, Italy: Berlin: Springer-Verlag, 1994, pp. 171–182.
- [29] I. H. Witten and E. Frank, *Data Mining - Practical Machine Learning Tools and Techniques with JAVA Implementations*. Morgan Kaufmann Publishers, 2005, 2nd Edition.
- [30] G. H. John, R. Kohavi, and K. Pfleger, “Irrelevant feature and the subset selection problem,” in *Proceedings of the Eleventh International Conference on Machine Learning*, 1994, pp. 121–129.
- [31] M. Robnik-Sikonja and I. Kononenko, “Theoretical and empirical analysis of Relief and ReliefF,” *Machine Learning*, vol. 53, pp. 23–69, 2003.
- [32] H. Liu, H. Motoda, and M. Dash, “A monotonic measure for optimal feature selection,” in *Machine Learning: ECML-98, April 21 - 23, 1998*, C. Nedellec and C. Rouveirol, Eds. Chemnitz, Germany: Berlin Heidelberg: Springer-Verlag, Apr. 1998, pp. 101–106.

- [33] D. A. Bell and H. Wang, “A formalism for relevance and its application in feature subset selection,” *Machine Learning*, vol. 41, no. 2, pp. 175–195, 2000.
- [34] M. Dash and H. Liu, “Consistency-based search in feature selection,” *Artificial Intelligence*, vol. 151, no. 1-2, pp. 155–176, 2003.
- [35] A. Jakulin, “Machine learning based on attribute interactions,” Ph.D. dissertation, University of Ljubljana, 2005.
- [36] C. Blake and C. Merz, “UCI repository of machine learning databases,” 1998. [Online]. Available: <http://www.ics.uci.edu/~mlearn/MLRepository.html>
- [37] A. Alizadeh and et al., “Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling,” *Nature*, vol. 403, pp. 503–511, 2000.
- [38] U. M. Fayyad and K. B. Irani, “Multi-interval discretization of continuous-valued attributes for classification learning,” in *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, 1993, pp. 1022–1027.
- [39] J. T. Irina Rish, Joseph L. Hellerstein, “An analysis of data characteristics that affect naive bayes performance,” IBM T.J. Watson Research Center, Tech. Rep., 2001.

Table 1: Features selected by each algorithm on synthetic data. ‘_’ indicates a missing relevant feature.

	Relevant Features	FCBF	CFS	ReliefF	FOCUS
Corral Data					
Corral	A_0, A_1, B_0, B_1	$A_0, A_1, B_0, B_1, \mathbf{R}$	$A_0, -, -, -, \mathbf{R}$	$A_0, A_1, B_0, B_1, \mathbf{R}$	A_0, A_1, B_0, B_1
MONKS Data (Full Data)					
Monk1	A_1, A_2, A_5	$-, -, A_5$	$-, -, A_5$	A_1, A_2, A_5	A_1, A_2, A_5
Monk2	$A_1, A_2, A_3, A_4,$ A_5, A_6	$A_1, A_2, A_3, A_4,$ A_5, A_6	$-, -, -, -,$ $A_5, -$	$-, A_2, A_3, A_4,$ $-, A_6$	$A_1, A_2, A_3, A_4,$ A_5, A_6
Monk3	A_2, A_4, A_5	A_2, A_4, A_5	$A_2, -, -$	$A_2, -, A_5$	A_2, A_4, A_5

Table 2: Summary of the benchmark data sets. #F: number of features, #I: number of instances, #C: number of classes.

Data Set	#F	#I	#C	Data Set	#F	#I	#C
Group 1				Group 2			
soy-small	35	46	4	lung-cancer	56	32	3
heart	13	269	2	zoo	16	101	7
ecoli	7	336	8	wine	13	178	3
glass	9	214	7	soy-large	35	306	19
wisc-canc	9	699	2	cmc	9	1,473	3
lymph	18	148	4	vehicle	18	846	4
pima	8	767	2	kr-vs-kp	36	3,196	2
austral	14	689	2	Group 3			
breast-cancer	9	286	2	musk2	166	6,598	2
credit	15	689	2	USCensus90	67	9,338	3
vote	16	435	2	45×4026+2C	4026	45	2
colic	22	368	2	internet-ads	1,558	3,279	2
german	20	999	2				
segment	19	2,310	7				
mushroom	22	8,124	2				
adult	14	32,560	2				

Table 3: Number of selected features for each feature selection algorithms on benchmark data sets.

Data Set	INTERACT	FCBF	CFS	ReliefF	FOCUS	Full Set
Group 1						
soy-small	2	5	9	18	3	35
heart	11	7	7	10	11	13
ecoli	6	5	4	7	7	7
glass	7	6	7	9	7	9
wisc-canc	6	8	9	9	6	9
lymph	7	8	10	14	7	18
pima	8	4	3	6	8	8
austral	13	6	1	12	13	14
breast-cancer	7	2	5	6	8	9
credit	13	6	1	11	13	15
vote	9	4	1	16	9	16
colic	9	5	1	17	10	22
german	14	3	3	16	13	20
segment	8	6	5	18	9	19
mushroom	5	4	1	20	NA	22
adult	12	5	5	14	NA	14
Average	8.56	5.25	4.50	12.69	9.07	15.63
Group 2						
lung-cancer	6	6	11	34	5	56
zoo	5	8	9	15	5	16
wine	5	10	8	13	5	13
soy-large	13	13	21	35	NA	35
cmc	9	2	3	6	9	9
vehicle	18	4	11	17	17	18
kr-vs-kp	29	7	3	19	NA	36
Average	12.14	7.14	9.43	19.86	8.20	26.14
Group 3						
musk2	19	2	10	162	NA	166
USCensus90	15	4	1	62	NA	67
45×4026+2C	3	64	42	2321	NA	4026
internet-ads	38	49	11	285	NA	1558
Average	18.75	29.75	16.00	707.50	NA	1454.25

Table 4: Accuracy of C4.5 on selected features: ‘Acc’ denotes 10-fold CV accuracy(%) and p -val obtained from a two-tailed t -test. The symbols “+” and “-” respectively identify statistically significant (at 0.05 level) if INTERACT wins over or loses to the compared algorithm. NA denotes the result is not available.

Data Set	INTERACT	FCBF		CFS		ReliefF		FOCUS		Full Set	
	Acc	Acc	p -val	Acc	p -val	Acc	p -val	Acc	p -val	Acc	p -val
Group 1											
soy-small	100.00	100.00	1.00	97.50	0.34	98.00	0.34	100.00	1.00	97.50	0.34
heart	76.98	79.96	0.02-	78.85	0.18	79.59	0.04-	76.98	1.00	76.98	1.00
ecoli	82.42	82.72	0.67	77.98	0.05+	82.42	1.00	82.42	1.00	82.41	0.98
glass	71.45	69.61	0.49	73.31	0.35	73.77	0.06	71.45	1.00	73.31	0.17
wisc-canc	95.43	95.43	1.00	95.28	0.73	95.28	0.73	94.71	0.82	95.28	0.73
lymph	75.76	72.43	0.10	73.10	0.31	73.05	0.37	73.65	0.54	75.71	0.99
pima	74.19	75.23	0.25	75.49	0.10	74.33	0.86	74.19	1.00	74.19	1.00
austral	86.51	84.33	0.01+	85.49	0.24	85.78	0.05+	85.34	0.73	85.93	0.10
breast-cancer	71.34	71.02	0.89	72.36	0.53	71.33	1.00	69.23	1.00	75.21	0.04-
credit	85.06	84.77	0.77	85.48	0.65	85.35	0.77	86.21	0.18	85.20	0.34
vote	96.78	95.86	0.04+	95.64	0.14	96.78	1.00	96.78	1.00	96.78	1.00
colic	85.31	80.16	0.01+	81.52	0.08	85.32	0.99	85.31	1.00	85.32	0.99
german	73.48	72.77	0.55	72.77	0.55	70.97	0.01+	69.57	0.06	69.87	0.01+
segment	96.62	96.97	0.18	96.36	0.47	96.41	0.54	96.62	0.62	96.45	0.66
mushroom	100.00	99.02	0.00+	98.52	0.00+	100.00	1.00	NA	NA	100.00	1.00
adult	86.06	85.70	0.04+	85.71	0.07	86.24	0.17	NA	NA	86.24	0.17
average	84.84	84.12		84.09		84.66		82.54		84.77	
Win/Loss		5/1		2/0		2/1		0/0		1/1	
Group 2											
lung-cancer	75.00	50.00	0.01+	43.75	0.00+	65.63	0.06	37.50	0.01+	46.88	0.02+
zoo	91.09	92.08	0.72	91.09	1.00	91.09	1.00	91.09	1.00	92.08	0.78
wine	96.63	93.82	0.05+	93.82	0.05+	83.15	0.00+	93.26	0.05+	93.82	0.05+
soy-large	82.03	84.31	0.25	84.31	0.17	84.31	0.31	NA	NA	85.95	0.11
cmc	52.14	51.93	0.89	54.11	0.08	50.17	0.10	52.27	0.17	52.27	0.17
vehicle	73.05	57.68	0.00+	69.62	0.00+	67.73	0.00+	73.05	1.00	73.05	1.00
kr-vs-kp	99.19	94.02	0.00+	90.43	0.00+	90.43	0.00+	NA	NA	99.44	0.31
average	81.30	74.83		75.30		76.07		69.43		77.64	
Win/Loss		4/0		4/0		3/0		2/0		2/0	
Group 3											
musk2	96.30	91.27	0.00+	95.76	0.25	96.79	0.12	NA	NA	96.85	0.16
USCensus90	98.25	98.12	0.16	97.99	0.01+	98.30	0.48	NA	NA	98.25	1.00
45×4026+2C	86.67	80.00	0.23	77.78	0.14	86.67	1.00	NA	NA	77.78	0.19
internet-ads	96.46	95.85	0.08	95.76	0.03+	91.52	0.00+	NA	NA	96.40	0.79
average	94.42	91.31		91.82		93.32		NA		92.32	
Win/Loss		1/0		2/0		1/0		NA		0/0	

Table 5: Accuracy of SVM on selected features: ‘Acc’ denotes 10-fold CV accuracy(%) and p -val obtained from a two-tailed t -test. The symbols “+” and “-” respectively identify statistically significant (at 0.05 level) if INTERACT wins over or loses to the compared algorithm. NA denotes the result is not available.

Data Set	INTERACT	FCBF		CFS		ReliefF		FOCUS		Full Set	
	Acc	Acc	p -val	Acc	p -val	Acc	p -val	Acc	p -val	Acc	p -val
Group 1											
soy-small	100.00	98.00	0.34	95.50	0.17	98.00	0.34	84.78	0.01+	100.00	1.00
heart	84.37	85.50	0.08	83.63	0.34	84.74	0.34	84.37	1.00	83.97	0.66
ecoli	83.33	83.33	1.00	78.57	0.00+	83.04	0.34	83.04	0.34	83.04	0.34
glass	56.17	56.62	0.73	52.90	0.02+	59.39	0.04-	56.17	1.00	59.39	0.04-
wisc-canc	96.14	96.71	0.22	96.57	0.34	96.57	0.34	96.14	1.00	96.57	0.34
lymph	81.76	81.71	0.99	84.38	0.12	87.14	0.01-	85.14	0.02-	88.43	0.01-
pima	77.18	77.31	0.83	76.26	0.21	76.92	0.51	77.18	1.00	77.18	1.00
austral	84.76	85.20	0.20	85.49	0.10	84.76	1.00	84.47	0.34	84.76	1.00
breast-cancer	70.62	70.33	0.93	67.13	0.31	68.89	0.60	71.68	0.34	70.27	0.89
credit	84.62	85.20	0.10	85.49	0.26	85.20	0.06	85.05	1.00	84.62	1.00
vote	95.63	95.63	1.00	95.63	1.00	94.70	0.38	95.63	1.00	94.70	0.38
colic	83.68	80.71	0.05+	81.52	0.07	83.15	0.70	83.42	0.17	82.06	0.17
german	75.67	71.67	0.00+	71.67	0.00+	75.78	0.92	75.38	1.00	74.97	0.33
segment	88.36	89.39	0.02-	89.00	0.21	92.73	0.01-	82.47	0.00+	92.77	0.01-
mushroom	99.90	99.02	0.05+	98.52	0.03+	100.00	0.31	NA	NA	100.00	0.31
adult	84.81	84.27	0.00+	84.06	0.00+	84.84	0.44	NA	NA	84.84	0.34
average	84.19	83.79		82.89		84.74		81.86		84.83	
Win/Loss		4/1		5/1		0/3		2/1		0/3	
Group 2											
lung-cancer	62.5	53.13	0.32	62.5	1.00	62.5	1.00	31.25	0.01+	37.5	0.03+
zoo	93.07	92.08	0.72	96.04	0.19	96.04	0.19	93.07	1.00	96.04	0.19
wine	96.63	98.31	0.08	96.63	1.00	84.83	0.01+	89.89	0.04+	97.75	0.17
soy-large	83.33	87.91	0.01-	92.16	0.00-	92.16	0.00-	NA	NA	91.18	0.00-
cmc	48.74	44.87	0.05+	49.97	0.37	44.33	0.02+	48.68	0.34	48.68	0.34
vehicle	73.88	42.08	0.00+	57.21	0.00+	50.35	0.00+	73.88	1.00	73.88	1.00
kr-vs-kp	95.9	94.06	0.00+	90.43	0.00+	90.43	0.00+	NA	NA	95.93	0.94
average	79.15	73.21		77.85		74.38		67.35		77.28	
Win/Loss		3/0		2/1		4/1		2/0		1/1	
Group 3											
musk2	90.19	84.59	0.00+	86.91	0.00+	94.97	0.00-	NA	NA	94.88	0.00-
census	98.27	98.08	0.03+	97.99	0.02+	98.36	0.25	NA	NA	98.29	0.79
45×4026+2C	95.56	100.00	0.12	100.00	0.12	95.56	1.00	NA	NA	91.11	0.39
internet-ads	96.34	95.97	0.08	95.94	0.02+	90.57	0.00+	NA	NA	97.28	0.01-
average	95.09	94.66		95.21		94.86		NA		95.39	
Win/Loss		2/0		3/0		1/1		NA		0/2	

Table 6: Comparison of INTERACT and INTERACT \setminus_R (INTERACT without feature ranking), accuracy of C4.5 on selected features. The symbols “+” and “-” respectively identify statistically significant (at 0.05 level) if INTERACT wins over or loses to the compared algorithm.

Group 1				Group 2			
Data Set	INTERACT	INTERACT \setminus_R	$p - val$	Data Set	INTERACT	INTERACT \setminus_R	$p - val$
soy-small	100.00	100.00	1.00	lung-cancer	75.00	44.17	0.04+
heart	76.98	76.98	1.00	zoo	91.09	91.09	1.00
ecoli	82.42	82.42	1.00	wine	96.63	96.63	1.00
glass	71.45	71.45	1.00	soy-large	82.03	82.03	1.00
wisc-canc	95.43	95.57	0.34	cmc	52.14	52.14	1.00
lymph	75.76	75.76	1.00	vehicle	73.05	73.05	1.00
pima	74.19	74.19	1.00	kr-vs-kp	99.19	99.19	1.00
austral	86.51	86.51	1.00	average	81.30	76.90	1.00
breast-cancer	71.34	71.34	1.00	Win/Loss			1/0
credit	85.06	85.06	1.00	Group 3			
vote	96.78	89.42	0.00+	Data Set	INTERACT	INTERACT \setminus_R	$p - val$
colic	85.31	85.59	0.58	musk2	96.30	95.59	0.84
german	73.48	72.47	0.12	USCensus90	98.25	97.02	0.03+
segment	96.62	96.62	1.00	45×4026+2C	86.67	78.00	0.00+
mushroom	100.00	98.97	0.00+	internet-ads	96.46	96.49	0.91
adult	86.06	86.06	1.00	average	94.42	91.78	
average	84.84	84.28		Win/Loss			2/0
Win/Loss			2/0				

Table 7: Comparison of INTERACT and INTERACT \setminus_R (INTERACT without feature ranking), accuracy of SVM on selected features. The symbols “+” and “-” respectively identify statistically significant (at 0.05 level) if INTERACT wins over or loses to the compared algorithm.

Group 1				Group 2			
Data Set	INTERACT	INTERACT \setminus_R	$p - val$	Data Set	INTERACT	INTERACT \setminus_R	$p - val$
soy-small	100.00	78.00	0.00+	lung-cancer	62.5	51.67	0.05+
heart	84.37	84.37	1.00	zoo	93.07	96.00	0.59
ecoli	83.33	83.33	1.00	wine	96.63	93.27	0.05+
glass	56.17	58.03	0.10	soy-large	83.33	85.66	0.03-
wisc-canc	96.14	96.42	0.51	cmc	48.74	48.74	1.00
lymph	81.76	78.95	0.45	vehicle	73.88	73.88	1.00
pima	77.18	77.18	1.00	kr-vs-kp	95.9	95.9	1.00
austral	84.76	84.76	1.00	Average	79.15	77.87	1.00
breast-cancer	70.62	70.28	0.61	Win/loss		2/1	
credit	84.62	84.62	1.00	Group 3			
vote	95.63	91.47	0.10	Data Set	INTERACT	INTERACT \setminus_R	$p - val$
colic	83.68	83.14	0.17	musk2	90.19	87.06	0.00+
german	75.67	72.77	0.01+	USCensus90	98.27	97.97	0.02+
segment	88.36	88.40	0.34	45×4026+2C	95.56	79.00	0.07
mushroom	99.90	100.00	0.10	internet-ads	96.34	95.30	0.01+
adult	84.81	84.81	1.00	average	95.09	89.83	
Average	84.19	82.28		Win/loss		3/0	
Win/loss		2/0					

Table 8: Comparison of INTERACT and INTERACT_{RF} (ranking using ReliefF), accuracy of C4.5 on selected features. The symbols “+” and “-” respectively identify statistically significant (at 0.05 level) if INTERACT wins over or loses to the compared algorithm.

Group 1				Group 2			
Data Set	INTERACT	INTERACT _{RF}	<i>p</i> - <i>val</i>	Data Set	INTERACT	INTERACT _{RF}	<i>p</i> - <i>val</i>
soy-small	100.00	100.00	1.00	lung-cancer	75.00	58.33	0.5+
heart	76.98	76.98	1.00	zoo	91.09	91.09	1.00
ecoli	82.42	82.42	1.00	wine	96.63	96.63	1.00
glass	71.45	71.45	1.00	soy-large	82.03	82.03	1.00
wisc-canc	95.43	95.57	0.34	cmc	52.14	52.14	1.00
lymph	75.76	75.76	1.00	vehicle	73.05	73.05	1.00
pima	74.19	74.19	1.00	kr-vs-kp	99.19	99.19	1.00
austral	86.51	86.51	1.00	Average	81.30	78.92	
breast-cancer	71.34	71.34	1.00	Win/loss			1/0
credit	85.06	85.06	1.00	Group 3			
vote	96.78	96.78	1.00	Data Set	INTERACT	INTERACT _{RF}	<i>p</i> - <i>val</i>
colic	85.31	85.59	0.58	musk2	96.30	95.30	0.05+
german	73.48	72.47	0.12	USCensus90	98.25	98.37	0.50
segment	96.62	96.62	1.00	45×4026+2C	86.67	94.00	0.17
mushroom	100.00	98.97	0.00+	internet-ads	96.46	96.67	0.13
adult	86.06	86.06	1.00	average	94.42	96.09	
Average	84.84	84.73		Win/loss			1/0
Win/loss			1/0				

Table 9: Comparison of INTERACT and INTERACT_{RF} (ranking using ReliefF), accuracy of SVM on selected features. The symbols “+” and “-” respectively identify statistically significant (at 0.05 level) if INTERACT wins over or loses to the compared algorithm.

Group 1				Group 2			
Data Set	INTERACT	INTERACT _{RF}	<i>p</i> - <i>val</i>	Data Set	INTERACT	INTERACT _{RF}	<i>p</i> - <i>val</i>
soy-small	100.00	100.00	1.00	lung-cancer	62.5	52.50	0.21
heart	84.37	84.37	1.00	zoo	93.07	93.07	1.00
ecoli	83.33	83.33	1.00	wine	96.63	96.63	1.00
glass	56.17	57.08	0.34	soy-large	83.33	83.33	1.00
wisc-canc	96.14	96.14	1.00	cmc	48.74	48.74	1.00
lymph	81.76	81.76	1.00	vehicle	73.88	73.88	1.00
pima	77.18	77.18	1.00	kr-vs-kp	95.9	96.09	0.54
austral	84.76	84.76	1.00	Average	79.15	77.75	1.00
breast-cancer	70.62	70.97	0.34	Win/loss		0/0	
credit	84.62	84.62	1.00	Group 3			
vote	95.63	95.63	1.00	Data Set	INTERACT	INTERACT _{RF}	<i>p</i> - <i>val</i>
colic	83.68	83.42	0.69	musk2	90.19	86.28	0.00+
german	75.67	75.28	0.70	USCensus90	98.27	98.22	0.42
segment	88.36	88.36	1.00	45×4026+2C	95.56	95.56	1.00
mushroom	99.90	96.60	0.00+	internet-ads	96.34	95.91	0.20
adult	84.81	84.82	0.44	average	95.09	93.99	
Average	84.19	84.02		Win/loss		1/0	
Win/loss		1/0					

Table 10: Run time (in second) for each feature selection algorithms over benchmark data sets.

Time	INTERACT	FCBF	CFS	ReliefF	FOCUS
Group 1					
soy-small	0.16	0.03	0.05	0.05	0.08
heart	0.11	0.02	0.02	0.11	6.77
ecoli	0.11	0.02	0.02	0.03	0.09
glass	0.11	0.02	0.02	0.02	0.23
wisc-canc	0.13	0.02	0.02	0.05	0.64
lymph	0.11	0.02	0.00	0.02	10.3
pima	0.13	0.02	0.02	0.06	0.53
austral	0.16	0.03	0.03	0.06	48.27
breast-cancer	0.09	0.00	0.00	0.02	0.38
credit	0.14	0.03	0.03	0.06	98.17
vote	0.11	0.02	0.00	0.03	47.89
colic	0.13	0.02	0.00	0.05	618.63
german	0.20	0.03	0.03	0.13	4938.16
segment	0.88	0.38	0.39	0.31	1097.13
mushroom	0.63	0.17	0.27	0.94	NA
adult	10.94	2.11	1.92	3.24	NA
Average	0.88	0.18	0.17	0.32	490.52
Group 2					
lung-cancer	0.09	0.00	0.05	0.03	2.31
zoo	0.09	0.02	0.02	0.02	0.81
wine	0.13	0.02	0.02	0.02	0.33
soy-large	0.14	0.03	0.06	0.09	NA
cmc	0.16	0.02	0.02	0.08	2.81
vehicle	0.22	0.05	0.06	0.11	976.58
kr-vs-kp	1.19	0.19	0.28	0.58	NA
Average	0.29	0.04	0.07	0.13	196.57
Group 3					
musk2	22.33	7.73	12.27	7.38	NA
USCensus90	3.75	0.89	1.08	3.03	NA
45×4026+2C	20.45	0.95	324.80	1.75	NA
internet-ads	149.42	63.33	53.58	31.27	NA
Average	48.99	18.23	97.93	10.86	NA

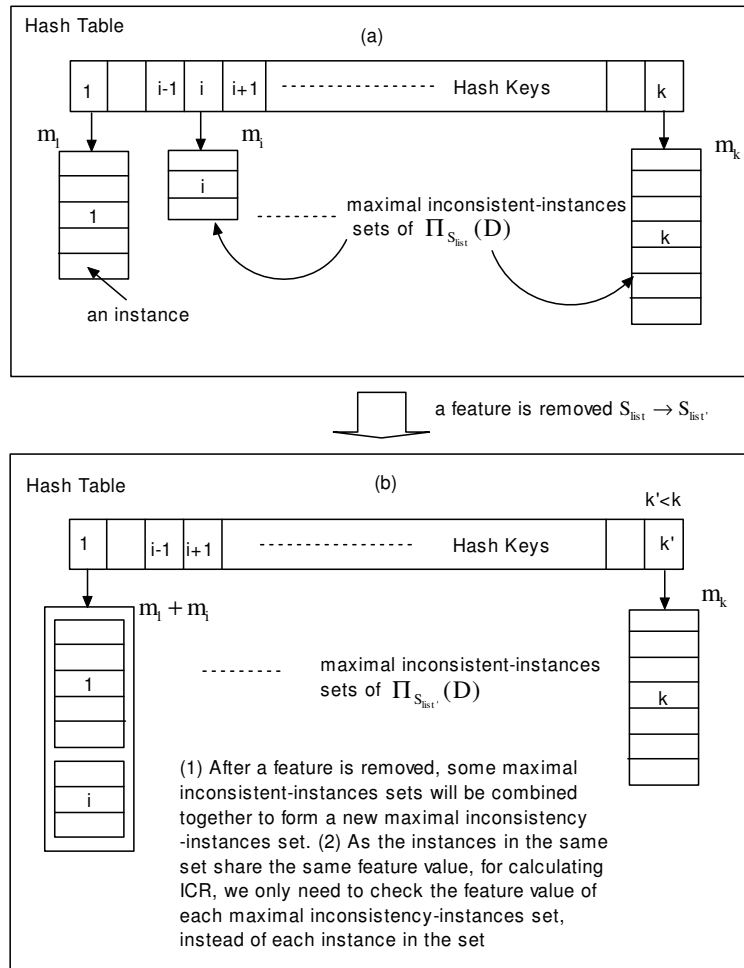
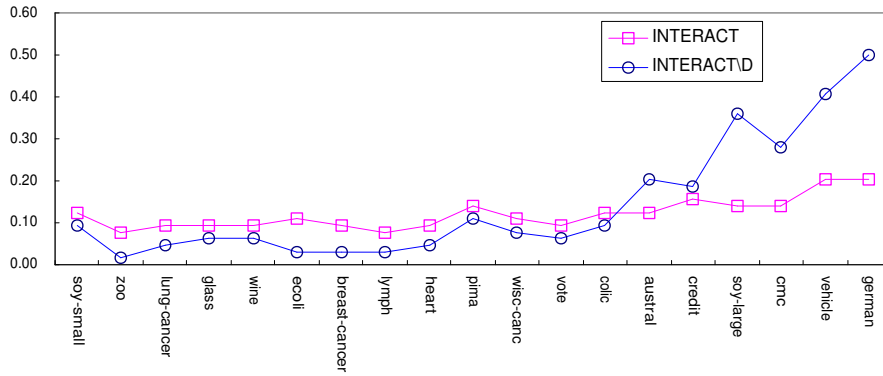
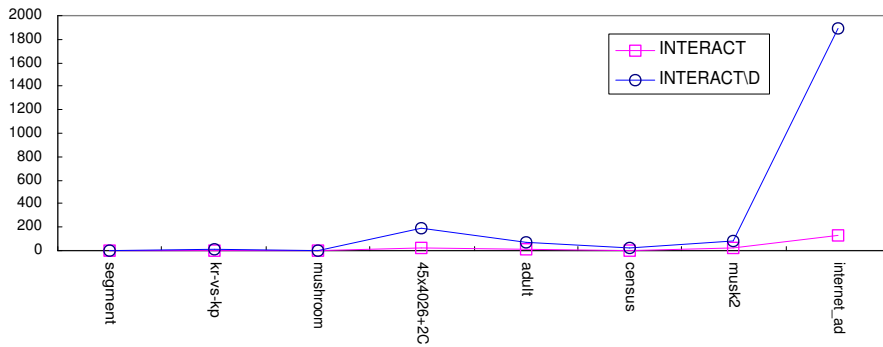


Figure 1: Efficient update of c -contribution

(a) Comparison on Runtime (Normal Scale, Datasets with Runtime <1s)



(b) Comparison on Runtime (Normal Scale, Datasets with Runtime >1s)



(c) Comparison on Runtime (Logarithmic Scale, Datasets with Runtime >1s)

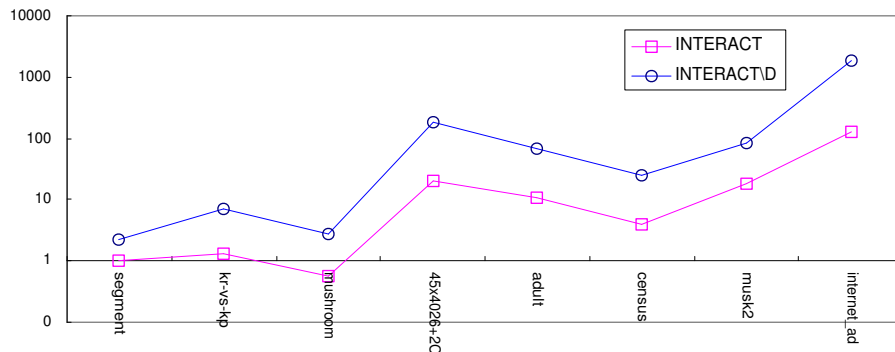


Figure 2: Run time (sec.) curves of INTERACT and INTERACT_D of the 27 data sets.

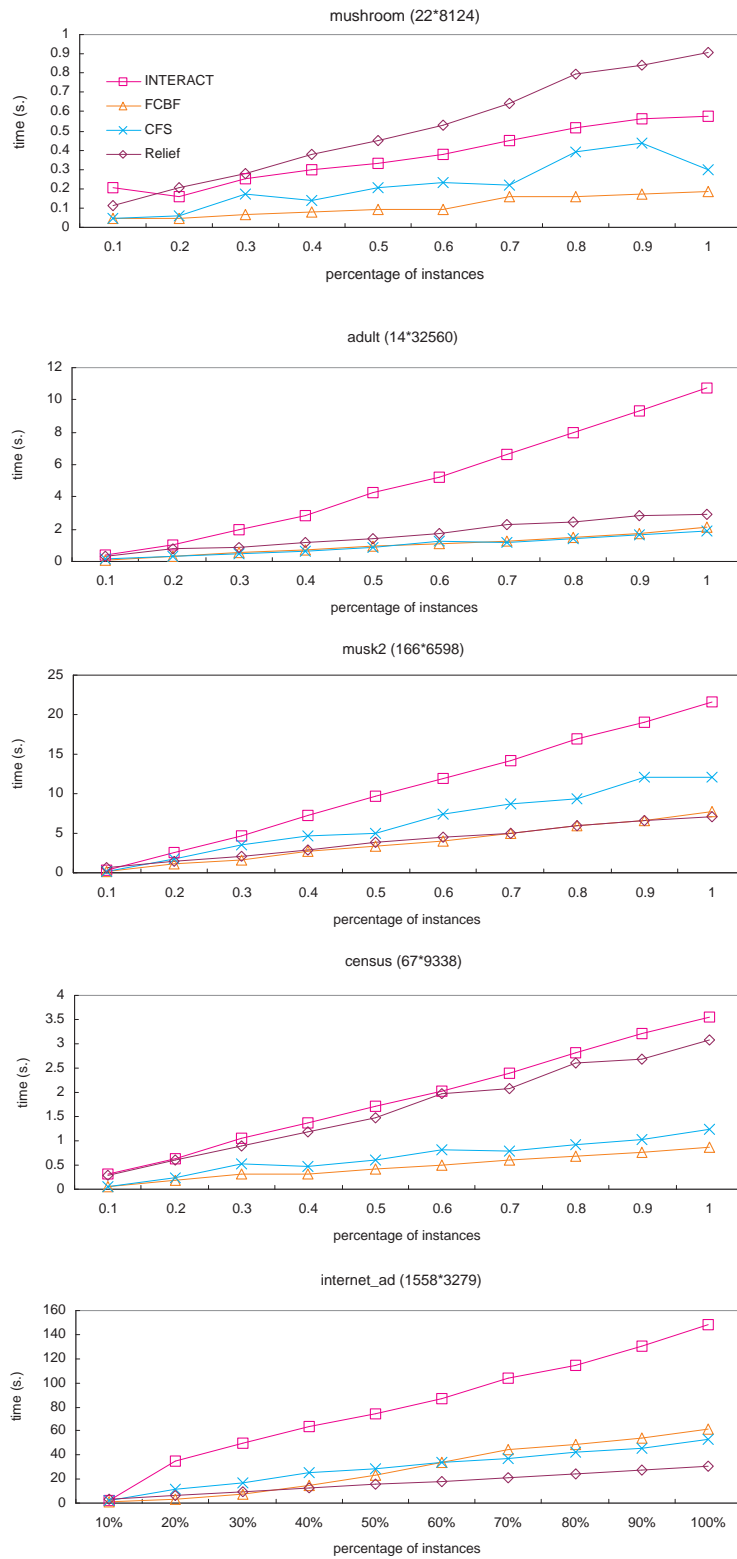


Figure 3: Run time (sec.) curves of each algorithm on five large data sets.