Mining Non-Redundant High Order Correlations in Binary Data

Xiang Zhang [1], Feng Pan [1], Wei Wang [1], and Andrew Nobel [2]
[1]Department of Computer Science, [2]Department of Statistics and Operations Research
University of North Carolina at Chapel Hill
[1]{xiang, panfeng,weiwang}@cs.unc.edu, [2]nobel@email.unc.edu

ABSTRACT

Many approaches have been proposed to find correlations in binary data. Usually, these methods focus on pair-wise correlations. In biology applications, it is important to find correlations that involve more than just two features. Moreover, a set of strongly correlated features should be non-redundant in the sense that the correlation is strong *only when all* the interacting features are considered together. Removing any feature will greatly reduce the correlation.

In this paper, we explore the problem of finding non-redundant high order correlations in binary data. The high order correlations are formalized using multi-information, a generalization of pair-wise mutual information. To reduce the redundancy, we require any subset of a strongly correlated feature subset to be weakly correlated. Such feature subsets are referred to as Non-redundant Interacting Feature Subsets (NIFS). Finding all NIFSs is computationally challenging, because in addition to enumerating feature combinations, we also need to check all their subsets for redundancy. We study several properties of NIFSs and show that these properties are useful in developing efficient algorithms. We further develop two sets of upper and lower bounds on the correlations, which can be incorporated in the algorithm to prune the search space. A simple and effective pruning strategy based on pair-wise mutual information is also developed to further prune the search space. The efficiency and effectiveness of our approach are demonstrated through extensive experiments on synthetic and real-life datasets.

1. INTRODUCTION

Finding correlations in high-dimensional binary data has attracted much research interest in recent years. Various approaches have been developed, including correlation pattern mining [26, 16, 13], feature selection [6, 27], finding correlated item pairs [35], and others. (See Section 2 for a more detailed discussion on related work.) Although often successful in different applications, these methods usually focus on pair-wise correlations between features. Some commonly used pair-wise correlation measurements are mutual information [3], all confidence [26], Pearson correlation [28] and so on. Methods such as neural networks [23] and multinomial mixture model [19] have been developed to capture the global correlations among all features.

PVLDB '08, August 23-28, 2008, Auckland, New Zealand

In this paper, we study the problem of finding higher order correlations in feature subspaces. Finding these high order correlations has important applications such as quantitative trait locus (QTL) finding [15] in genetics. In QTL finding, geneticists want to identify genomic regions that are associated with a phenotype (or trait) of interest by using single nucleotide polymorphism (SNP) data. Each SNP can be treated as a binary feature. It is well known that many diseases are complex traits, i.e., *multiple genes (SNPs) interacting with each other* to control the expression of the disease. Each disease SNP may only have weak correlation with the disease trait. However, when these disease SNPs are combined together, their correlation with the disease trait becomes very strong. Finding multiple SNPs showing strong correlation with phenotype variation is an active and growing research area in genetics [2, 32].

From the above application, we can see two aspects of the desired correlation patterns. First, the correlation involves more than two features. Second, the correlation is non-redundant, i.e., removing any feature will greatly reduce the correlation.

To make the scenario clearer, let's consider the following simple example. Suppose X, Y and Z are binary features, where X and Y are disease SNPs, and Z is the complex disease trait, which is controlled by X and Y together. Suppose X and Y are independent, and $Z = X \oplus Y$, where $\oplus$ denotes exclusive-or (XOR) operation. Clearly, $\{X, Y, Z\}$ together have strong correlation, since when combined together, X and Y uniquely determine Z. However, for each pair of the features, $\{X, Z\}$, $\{Y, Z\}$, and $\{X, Y\}$, there is no correlation. From this example, we can see that the high order correlation pattern cannot be identified by only examining the pair-wise correlations.

Besides the application in genetics, the importance of exploring high order correlations among features has also been demonstrated in [10, 36]. In [10], a statistic based on Kullback-Leibler divergence [3] is proposed to assess the significance of feature interactions. The multi-way correlations exist in many real-life datasets. In [36], the authors develop a feature selection method that takes the interactions among features into consideration, and demonstrate that the resulting features achieve higher classification accuracy than features selected without considering such interactions.

We now look at another example where strong correlation exists only when multiple features are considered together. The dataset in this example will also be used as the running dataset of this paper.

EXAMPLE 1.1. *Figure 1 shows a dataset consisting of 11 binary features. Let $H(X)$ denote the entropy [3] of feature X, and $\tilde{I}(Y;X) = \frac{H(Y) - H(Y|X)}{H(Y)}$ be the relative entropy reduction of Y based on X. Consider the three features X_1, X_2, and X_3 in the table. We have $\tilde{I}(X_3;X_1) = 21.97\%$ and $\tilde{I}(X_3;X_2) = 8.62\%$,*

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
s_1	0	0	0	0	0	1	0	0	0	0	1
s_2	0	0	0	1	0	0	0	0	0	0	1
s_3	1	1	0	0	0	0	1	1	1	1	0
s_4	1	1	0	1	0	1	1	0	1	0	1
s_5	0	1	0	1	0	0	0	1	0	1	1
s_6	0	0	0	1	0	0	1	0	1	0	1
s_7	1	0	1	1	1	0	0	1	0	1	0
s_8	1	0	1	0	0	1	1	1	0	0	1
s_9	0	0	0	1	0	1	1	1	1	1	0
s_{10}	0	1	0	1	0	0	1	0	0	1	1
s_{11}	0	1	0	1	0	0	0	1	0	1	0
s_{12}	0	0	0	1	0	1	0	1	1	0	1
s_{13}	0	1	1	0	1	0	1	0	0	1	0
s_{14}	1	0	1	0	1	0	0	0	1	1	0
s_{15}	1	1	0	1	0	1	0	0	0	0	1
s_{16}	1	0	1	0	1	0	0	1	0	1	0
s_{17}	0	0	0	1	0	1	0	0	0	0	0
s_{18}	1	1	0	1	0	1	1	0	1	0	1
s_{19}	1	0	1	0	1	0	1	0	0	1	0
s_{20}	0	0	0	1	0	0	0	1	0	1	0

Figure 1: An example dataset

i.e., the relative entropy reduction of X_3 given X_1 or X_2 alone is small. However, the relative entropy reduction of X_3 given both X_1 and X_2 is much higher, $\tilde{I}(X_3; X_1, X_2) = 81.59\%$. Therefore, X_1 and X_2 jointly reduce the uncertainty of X_3 more than they do separately. This strong correlation exists only when these three features are considered together.

Scope and contributions: In this paper, we study the problem of finding non-redundant high order correlations in binary data. We formalize these correlation patterns as Non-redundant Interacting Feature Subsets (NIFSs). In particular, we use an entropy based correlation measurement, multi-information [34], to measure the high order correlation among a set of features. An NIFS is a subset of features satisfying the following two criteria. First, the features in an NIFS together has high multi-information. Second, all subsets of an NIFS have low multi-information.

Since the NIFSs satisfy two criteria, the computational challenge of finding them is also two-folds. First, we need to enumerate feature combinations to find the feature subsets that have high correlation. Second, for each such feature subset, we need to check all its subsets to make sure there is no redundancy.

We study the properties of NIFSs and show that they can be used in developing efficient algorithms. In particular, we show that any superset of an NIFS cannot be an NIFS. This allows us to prune the search space whenever an NIFS is found. We develop two sets of upper and lower bounds on the correlation of an NIFS. One is based on Han's inequality [8], and the other is based on Hamming distances between the features. These bounds can be easily incorporated in the algorithm to improve the efficiency. Finally, we develop a pruning strategy based on mutual information, which enables the algorithm to further prune the search space.

We apply our approach to both synthetic and real-life datasets. The significance of identified patterns is carefully examined using various approaches.

2. RELATED WORK

There has been a significant amount of work in finding correlations in binary data.

Correlation pattern mining Correlation pattern mining [26, 16, 13] is an extensively studied area in data mining. The algorithms developed for correlation pattern mining typically measure the correlations between attributes by pairwise mutual information, measurements based on support, or other statistics, such as Pearson correlation. The problem of finding strongly correlated item pairs is studied in [35]. This method is designed for finding pairwise correlations,which is a special case of the problem studied in this paper. In [9, 14], the authors investigate the problem of finding itemsets having high or low entropy, which is different from finding non-redundant interacting features subsets.

Capturing global correlation Many methods have been proposed to capture the global correlations among all the features, such as principal component analysis (PCA) [11], neural networks [23], and multinomial mixture model [19]. These methods have been widely used and shown effective in various applications. However, these methods are designed to capture the correlations in the full feature space. Our work in this paper, on the other hand, focuses on finding the high order correlations hidden in feature subsets.

Feature selection Feature selection methods [18, 6, 27, 36] try to find a subset of features that are most relevant for a certain data mining task, such as classification. The selected feature subset usually contains the features that have low correlation with each other but have strong correlation with the target feature. Methods exploiting mutual information between pairs of binary features are studied in [6, 27]. In [27], the authors propose to use Max-Relevance and Min-Redundancy (mRMR) as the criteria for feature selection. The goal is to select non-redundant features that are most relevant to the dependent feature. However, the model of mRMR is still based on the mutual information between a pair of features and does not generalize to higher order correlations. In [36], a feature selection method utilizing interactions among features is proposed. This method performs a single pass backward elimination to find a best feature subset that predicts the class labels well. Our focus in this paper, however, is to find all non-redundant interacting feature subsets.

Statistical significance of feature interactions In [10], the authors propose a Kullback-Leibler (KL) divergence [3] based statistic to measure the interactions among a set of features. The basic idea is as follows. Given a subset of features, say F, use KL divergence as a statistic to measure the difference between (1) the observed joint probability distribution of F and (2) an estimated joint probability distribution derived from pair-wise approximations. The Kirkwood superposition approximation is suggested in the paper as the estimated distribution. For example, an approximation $\hat{p}_K(A, B, C)$ to the joint probability density function $p(A, B, C)$ is

$$\hat{p}_K(a, b, c) \approx \frac{p(a, b)p(a, c)p(b, c)}{p(a)p(b)p(c)} = p(a|b)p(b|c)p(c|a).$$

One then calculates the P-value of the resulting KL divergence using a chi-square test, or a bootstrap based percentile. If the P-value is small, it indicates that there are interactions among features in F. It has been shown that such interactions exist in many real datasets. In [10], besides the suggested statistic, no algorithm is presented to find the these patterns.

Dimension reduction Principal component analysis (PCA) and singular value decomposition (SVD) [11] are well known dimension reduction methods for real-value data. Some research has been done in designing dimension reduction methods specifically for binary data [7, 22, 24]. Similar to PCA and SVD, these methods focus on finding a set of new features that approximate the original data, but do not consider the problem of finding multiple sets of interacting features. In [33], a method based on fractal dimension is proposed to estimate the intrinsic dimensionality of binary data without actually finding the representative features.

Clustering Clustering methods designed for binary data [5, 17] partition the data points or features into groups based on pairwise similarities. These methods can identify groups of features with strong pairwise correlations, but are not adapted to higher order interactions.

3. PRELIMINARIES

Many statistical measures have been proposed in the data mining literature for measuring pair-wise correlations, such as Pearson correlation, Spearman correlation, χ^2 statistics etc [21]. Although these measurements are widely used in different applications, they are not specifically designed for capturing higher order correlations among a set of features.

The concept of multi-information was first discussed in detail in [34], though it had been described earlier in [20].

DEFINITION 1. *The* **multi-information** *of a set of features* $\{X_1, X_2, \cdots, X_n\}$ *is defined as*

$$C(X_1, X_2, \cdots, X_n) = \sum_{i=1}^{n} H(X_i) - H(X_1, X_2, \cdots, X_n),$$

where $H(X_i)$ *is the entropy of* X_i.

A multivariate correlation model based on multi-information allows exploration of bivariate as well as and higher order correlations. The multi-information measure is specifically designed for modelling the feature interactions and has rigorous theoretical backgrounds. It has been widely used in many domains, such as biology and physics [31]. Note that the pairwise mutual information is a special case of multi-information, with $n = 2$. Multi-information is always non-negative and equal to zero only when $X_1, \ldots, X_n$ are independent. See [34] for more details.

4. PROBLEM FORMALIZATION

In this section, we formalize the concept of a non-redundant interacting feature subset (NIFS) using multi-information. An NIFS is a subset of features that have high multi-information when and only when all features in the subset are considered together. After defining NIFSs, we study several properties that can be utilized to design efficient algorithms.

4.1 Definitions

We first define strongly and weakly correlated feature subsets. The definition of NIFS is based on these two concepts. A set of features is strongly correlated if the multi-information is above some user specified threshold. It is weakly correlated if its multi-information is lower than a user-specified threshold.

DEFINITION 2. *A set of features* $\{X_1, X_2, \cdots, X_n\}$ *is* strongly correlated *if* $C(X_1, X_2, \cdots, X_n) \geq \beta$, *where* $\beta > 0$ *is a user-defined threshold. In this case,* $\{X_1, X_2, \cdots, X_n\}$ *is called a* Strong-correlated Feature Subset (SFS).

DEFINITION 3. *A set of features* $\{X_1, X_2, \cdots, X_n\}$ *is* weakly correlated *if* $C(X_1, X_2, \cdots, X_n) \leq \alpha$. *where* $0 < \alpha < \beta$ *is a user-defined parameter. In this case,* $\{X_1, X_2, \cdots, X_n\}$ *is called a* Weak-correlated Feature Subset (WFS).

Larger β corresponds to higher correlation and smaller α corresponds to weaker correlation.

EXAMPLE 4.1. *Consider the dataset in Figure 1. Let* $\alpha = 0.25$ *and* $\beta = 0.8$. *We have* $\{X_1, X_2, X_3\}$, $\{X_1, X_2, X_3, X_6\}$ *and* $\{X_7, X_8, X_9, X_{10}\}$ *are SFSs, since their multi-information is greater than* β, *with* $C(X_1, X_2, X_3) = 0.82$, $C(X_1, X_2, X_3, X_6) = 0.97$, *and* $C(X_7, X_8, X_9, X_{10}) = 1.15$. *On the other hand,* $\{X_1, X_2\}$ *and* $\{X_7, X_8, X_9\}$ *are WFSs, since their multi-information is smaller than* α, *with* $C(X_1, X_2) = 0.03$ *and* $C(X_7, X_8, X_9) = 0.15$.

In the example above, although $\{X_1, X_2, X_3, X_6\}$ is an SFS, its subset $\{X_1, X_2, X_3\}$ is also an SFS. Therefore, the collection $\{X_1, X_2, X_3, X_6\}$ is not parsimonious, since one of its subsets has already shown strong correlation. In order to remove redundancy from interacting feature subsets, we require that any subset of an NIFS is weakly correlated.

DEFINITION 4. *A subset of features* $\{X_1, X_2, \cdots, X_n\}$ *is* **Non-redundant Interacting Feature Subset (NIFS)** *if the following two criteria are satisfied:*
(1) $\{X_1, X_2, \cdots, X_n\}$ *is an SFS; and*
(2) every proper subset $X' \subset \{X_1, X_2, \cdots, X_n\}$ *is a WFS.*

EXAMPLE 4.2. *Consider the dataset shown in Figure 1.* $\{X_1, X_2, X_3\}$ *is an NIFS, since* $\{X_1, X_2, X_3\}$ *is an SFS, with* $C(X_1, X_2, X_3) = 0.82$, *and all its subsets are WFSs:* $C(X_1, X_2) = 0.03$, $C(X_1, X_3) = 0.22$,*and* $C(X_2, X_3) = 0.22$. *Similarly,* $C(X_7, X_8, X_9, X_{10})$ *is an NIFS, since it is an SFS, and all its subsets are WFSs.*

Overall goal: Given a binary data set and parameters $0 < \alpha < \beta$, find all NIFSs.

To find the NIFSs in a dataset, we generate candidate feature subsets by enumerating combinations of the features. In order to verify if a candidate feature subset is an NIFS, in addition to computing its own multi-information, we also need to compute the multi-information for its subsets to make sure that they are WFSs. In the next section, we establish some properties of NIFSs that can greatly reduce the computational complexity.

4.2 Properties related to NIFSs

In this subsection, we exploit some general properties of NIFSs that can be used for the designing of efficient algorithms.

PROPERTY 4.3. *(*Downward closure property of WFSs*) If feature subset* $\{X_1, X_2, \cdots, X_n\}$ *is a WFS, then all its subsets are WFSs.*

PROOF. It suffices to prove that if $\{X_1, X_2, \cdots, X_n\}$ is a WFS, then $\{X_1, X_2, \cdots, X_{n-1}\}$ is also a WFS. Note that

$C(X_1, X_2, \cdots, X_n) - C(X_1, X_2, \cdots, X_{n-1})$
$= H(X_n) - H(X_1, \cdots, X_n) + H(X_1, \cdots, X_{n-1})$
$= -H(X_1, \cdots, X_{n-1}|X_n) + H(X_1, \cdots, X_{n-1})$
$= I(X_1, X_2, \cdots, X_{n-1}; X_n) \geq 0.$

Thus, if $C(X_1, X_2, \cdots, X_n) \leq \alpha$, then have $C(X_1, X_2, \cdots, X_{n-1}) \leq \alpha$.

Therefore, $\{X_1, X_2, \cdots, X_{n-1}\}$ is also a WFS. □

The significance of Property 4.3 is following. For a candidate feature subset, in order to justify the second criterion of Definition 4, we need to check all its subsets. Given a feature subset $X = \{X_1, X_2, \cdots, X_n\}$, there are 2^n distinct subsets. Property 4.3 tells us that we only need to compute the multi-information of the subsets X' of X with $(n-1)$ features. This is because if X' is a WFS, then all its subsets are WFSs. This greatly reduces the complexity of the problem. Among the 2^n subsets of X, there are only n subsets of size $(n-1)$.

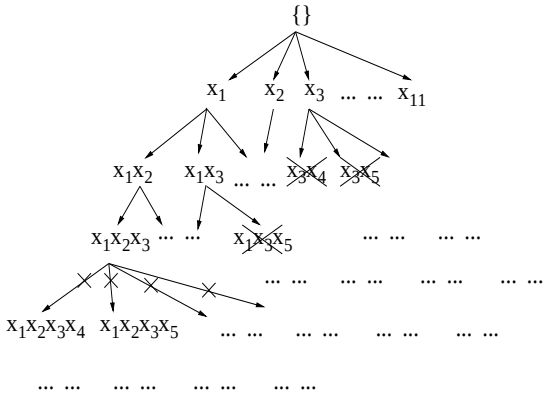


Figure 2: Enumerating candidate NIFSs

NIFSs do not satisfy upward or downward closure property. However, it has an interesting property that can help to prune the search space. That is, if a feature subset is an NIFS, then all its supersets are not NIFSs.

PROPERTY 4.4. *Let $X = \{X_1, X_2, \cdots, X_n\}$ be an NIFS. Any $Y \supset X$ is not an NIFS.*

PROOF. Since X is an NIFS, we have $C(X) \geq \beta > \alpha$. Thus, for any superset Y of X, $\exists X \subset Y$, such that X is not a WFS. Thus the second criterion of Definition 4 is not satisfied. Therefore Y is not an NIFS. □

Property 4.4 offers the possibility of developing efficient algorithms for finding NIFSs. If the algorithm finds an NIFS, then its subtree in the search space can be safely pruned without any further examination.

EXAMPLE 4.5. *Consider the dataset shown in Figure 1. Figure 2 shows the search space for finding NIFSs based on the generic set-enumeration tree search framework [30]. Once an NIFS is found by the algorithm, say $\{X_1, X_2, X_3\}$, then its subtree can be pruned as shown in the figure.*

For each candidate feature subset, we need to verify both criteria in Definition 4. In Section 5, using Han's inequality [8], we develop upper and lower bounds that can be used to estimate the multi-information of a candidate before actually calculating it. In Section 6, we further develop upper and lower bounds of multi-information for a candidate feature subset based on the multi-information of its parent and sibling nodes in the search space. In Section 7, we present a pruning strategy based on the mutual information between pairwise features. The overall algorithm is described in Section 8.

5. BOUNDS BASED ON PAIRWISE CORRELATIONS

In this section, we investigate upper and lower bounds on the multi-information of a candidate feature subset. The bounds involve the pair-wise correlations (mutual informations) of the features in the candidate, and are derived from existing inequalities in the information theory literature. These bounds are used in the algorithm to provide a range of values of the multi-information of a candidate without computing the multi-information directly. We refer to the bounds developed in this section as ***pairwise correlation propositions***.

DEFINITION 5. *Let $\{X_1, X_2, \cdots, X_n\}$ be a set of features, and for every $S \subseteq \{1, 2, \cdots, n\}$, denote by $X(S)$ the subset $\{X_i : i \in S\}$. Let $h_k^{(n)} = \frac{1}{\binom{n}{k}} \sum_{S:|S|=k} \frac{H(X(S))}{k}$. Thus $h_k^{(n)}$ is the average entropy in bits per symbol of a randomly drawn k-element subset of $\{X_1, X_2, \cdots, X_n\}$.*

The following result of Han [8] says that the average entropy decreases monotonically as the size of the subset increases.

THEOREM 5.1. $h_1^{(n)} \geq h_2^{(n)} \geq \cdots \geq h_n^{(n)}$.

Applying Han's inequity, we can get the following two propositions. Propositions 5.2 and 5.3 give lower and upper bounds on the multi-information of a candidate feature set in terms of the pairwise multi-information (i.e., mutual information) of its members. The proofs are omitted due to space limitation.

PROPOSITION 5.2.

$$C(X_1, X_2, \cdots, X_n) \geq \frac{1}{n-1} \sum_{i<j} C(X_i X_j).$$

PROPOSITION 5.3.

$$C(X_1, \cdots, X_n) \leq \sum_{i=1}^{n} H(X_i) - \max_{i \neq j} \{H(X_i, X_j)\}.$$

Propositions 5.2 and 5.3 are refereed to as pairwise correlation propositions, since these bounds depend on the pairwise correlations between the features in the candidate.

6. BOUNDS BASED ON HAMMING DISTANCES

In this section we investigate the effect of adding or replacing a feature on the multi-information of a candidate feature subset. In each case, we obtain upper and lower bounds on the change in the multi-information in terms of Hamming distance between features. As in the previous section, the bounds obtained in this section are used to provide a pruning scheme for the algorithm. We call the result of Section 6.1 the ***adding proposition***, and the result of Section 6.2 the ***replacing proposition*** respectively.

Let $X = \{X_1, X_2, \cdots, X_n\}$ be a node (candidate feature subset) in the search space of NIFSs. The multi-information of X can be used to estimate the multi-information of two kinds of candidate feature subsets. The first are $X' = \{X_1, X_2, \cdots, X_n, X_{n+1}\}$, i.e., the child nodes of X in the search space. These are candidates that include exactly one more feature than X. The second are $X'' = \{X_1, X_2, \cdots, X_{n-1}, X_{n+1}\}$, i.e., the sibling nodes of X in the search space. These are candidates that replace one feature in X by a new feature. Below we show that the multi-information of X' is bounded by a function of the Hamming distance between the newly added feature and features in X. Likewise, the multi-information of X'' is bounded by a function of the Hamming distance between the new feature and the feature being replaced in X.

Let K be the number of instances (data points) in the dataset. In what follows, define $f(k) = \frac{k}{K} \log \frac{K}{k}$, with $0 \leq k \leq K$ and define $f(0)$ to be 0. Propositions 6.1 and 6.2 are the basis for the bounds developed in this section. Their proofs are omitted here.

PROPOSITION 6.1. *For $1 \leq c \leq k \leq K$, we have*

$$-f(K-c) \leq f(k) - f(k-c) \leq f(c).$$

PROPOSITION 6.2. *For $1 \leq k \leq K$, we have*

$$f(k) + f(K-k) \leq k(f(1) + f(K-1)).$$

6.1 Adding a new feature

Suppose that $\{X_1, X_2, \cdots, X_n\}$ is the current candidate feature subset, and $\Delta C = C(X_1, \cdots, X_n, X_{n+1}) - C(X_1, \cdots, X_n)$. We develop bounds for ΔC, beginning with Proposition 6.3 and it's generalization in Proposition 6.4.

PROPOSITION 6.3. *Let X_1 and X_2 be two features in the dataset. If the Hamming distance between X_1 and X_2 is d, then*

$$0 \leq H(X_1, X_2) - H(X_1) \leq d(f(1) + f(K-1)).$$

PROOF. Let

$A' = -p_{x_1}(0) \log p_{x_1}(0)$,

$B' = -p_{x_1}(1) \log p_{x_1}(1)$,

$A = -p_{x_1x_2}(0,0) \log p_{x_1x_2}(0,0)$,

$B = -p_{x_1x_2}(1,1) \log p_{x_1x_2}(1,1)$,

$C = -p_{x_1x_2}(1,0) \log p_{x_1x_2}(1,0)$,

$D = -p_{x_1x_2}(0,1) \log p_{x_1x_2}(0,1)$.

Then, $H(X_1) = A' + B'$, and $H(X_1, X_2) = A + B + C + D$. It is easy to see that $p_{x_1}(0) = p_{x_1x_2}(0,0) + p_{x_1x_2}(0,1)$, and $p_{x_1}(1) = p_{x_1x_2}(1,0) + p_{x_1x_2}(1,1)$.

Suppose that $p_{x_1}(0) = t/K$, and $p_{x_1x_2}(0,1) = s/K$. Then, $p_{x_1x_2}(0,0) = (t-s)/K$ and we have

$A + D - A' = f(s) - (f(t) - f(t-s))$.

According to Proposition 6.1,

$0 \leq A + D - A' \leq f(s) + f(K-s)$.

Similarly, suppose $p_{x_1}(1) = u/K$, and $p_{x_1x_2}(1,0) = v/K$. Then $p_{x_1x_2}(1,1) = (u-v)/K$ and we have

$0 \leq B + C - B' \leq f(v) + f(K-v)$.

Moreover, Proposition 6.2 implies that

$f(k) + f(K-k) \leq k(f(1) + f(K-1))$

and therefore,

$H(X_1, X_2) - H(X_1) \leq (s+v)(f(1) + f(K-1))$.

Since there are d different positions between X_1 and X_2, we have $s + v = d$ and the proof is complete. □

The following generalization is easy to derive and its proof is therefore omitted.

PROPOSITION 6.4. *Let*

$$\Delta H = H(X_1, \cdots, X_n, X_{n+1}) - H(X_1, \cdots, X_n).$$

If the minimum Hamming distance between X_{n+1} and X_i ($1 \leq i \leq n$) is d, then

$$0 \leq \Delta H \leq d(f(1) + f(K-1)).$$

From Proposition 6.4, it is easy to derive upper and lower bounds on ΔC.

PROPOSITION 6.5. (Adding Proposition) *Let*

$$\Delta C = C(X_1, \cdots, X_n, X_{n+1}) - C(X_1, \cdots, X_n).$$

If the minimum Hamming distance between X_{n+1} and X_i ($1 \leq i \leq n$) is d, then

$$H(X_{n+1}) - d(f(1) + f(K-1)) \leq \Delta C \leq H(X_{n+1}).$$

The bounds provided in Proposition 6.5 can be used to estimate the multi-information of the child nodes of a candidate feature subset based on the Hamming distances between the features in the candidate and the newly added features.

6.2 Replacing a feature

Suppose that the current candidate feature subset is $\{X_1, X_2, \cdots, X_n\}$. In this section, we develop the bounds of multi-information for its sibling nodes in the search space, i.e., for candidate $\{X_1, X_2, \cdots, X_{n-1}, X_{n+1}\}$.

In Propositions 6.6 and 6.7, we establish bounds on the change in joint entropy when replacing a feature in the candidate set by a new one. Bounds on multi-information are established in Proposition 6.8. In this subsection, we use ΔH denote the entropy difference, i.e.,

$$\Delta H = H(X_1, \cdots, X_n) - H(X_1, \cdots, X_{n-1}, X_{n+1}).$$

PROPOSITION 6.6. *If the Hamming distance between X_{n+1} and X_n is 1, then*

$$|\Delta H| \leq f(1) + f(K-1).$$

The proof of Proposition 6.6 is similar to that of Proposition 6.3 and thus omitted. Proposition 6.6 can easily be generalized to yield the following proposition, whose proof is omitted.

PROPOSITION 6.7. *If the Hamming distance between X_{n+1} and X_n is d, then*

$$|\Delta H| \leq d(f(1) + f(K-1)).$$

From Proposition 6.7, it is easy to obtain bounds for ΔC.

PROPOSITION 6.8. (Replacing Proposition) *Let*

$$H_\delta = H(X_{n+1}) - H(X_n),$$

$$\Delta C = C(X_1, \cdots, X_n) - C(X_1, \cdots, X_{n-1}, X_{n+1}).$$

If the Hamming distance between X_{n+1} and X_n is d, then

$$H_\delta - d(f(1) + f(K-1)) \leq \Delta C \leq H_\delta + d(f(1) + f(K-1)).$$

The bounds in Proposition 6.8 can be used to estimate the multi-information of sibling nodes of a candidate feature subset using the Hamming distances between the new features and the features being replaced.

As a brief summary of this section, Propositions 6.5 and 6.8 establish theoretic bounds on the multi-information for the child and sibling nodes of a candidate feature subset based on the Hamming distance between features. These two propositions are referred to as adding proposition and replacing proposition respectively.

7. PRUNING CANDIDATES BY MUTUAL INFORMATION

In Section 5, it is shown that the mutual information between feature pairs can be used to bound the multi-information of candidate feature subsets. In this section, we show that mutual information can also be used as a pruning strategy in the process of enumerating of candidate feature subsets.

The basic idea is simple. Suppose that the mutual information between two features $\{X_i, X_j\}$ is greater than α, i.e., $\{X_i, X_j\}$ is not a WFS. Then due to the downward closure property (Property 4.3) of WFSs, any superset of $\{X_i, X_j\}$ cannot be an NIFS, since it has a subset $\{X_i, X_j\}$ that is not a WFS. Therefore, all supersets of $\{X_i, X_j\}$ can be safely pruned.

EXAMPLE 7.1. *Consider the example dataset shown in Figure 1. To be consistent with previous examples, let $\alpha = 0.25$ and*

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}
$\emptyset$	$\emptyset$	X_4	X_3	X_3	X_{10}	$\emptyset$	$\emptyset$	$\emptyset$	X_5	X_5
		X_5		X_{10}					X_6	X_{10}
				X_{11}					X_{11}	

Table 1: Feature pairs with mutual information larger than α

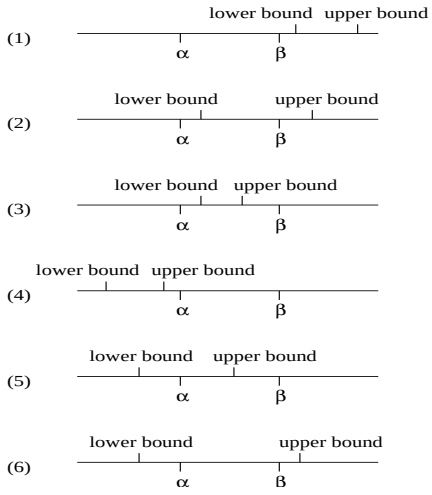


Figure 3: The possible relative positions of upper bound, lower bound, α, and β

$\beta = 0.8$. *Table 1 shows the feature pairs whose mutual information is larger than α. For each feature X_i in the dataset, the features having strong mutual information with X_i are listed in the corresponding columns. For example, X_4 and X_5 both have strong mutual information with X_3, i.e., $\{X_3, X_4\}$ and $\{X_3, X_5\}$ are not WFSs. Then in the search space, any node containing $\{X_3, X_4\}$ or $\{X_3, X_5\}$ can be pruned. This also applies to other feature pairs in the table. Some candidates that can be pruned using this strategy are shown in the Figure 2.*

8. THE ALGORITHM

In this section, we present the overall algorithm. In Section 5 and Section 6, we have established upper and lower bounds on the multi-information of a feature set based on mutual informations and Hamming distances. Before calculating the multi-information of a node in the search space, we first check its upper and lower bounds. The pairwise correlation propositions (Propositions 5.2 and 5.3) can be applied whenever the algorithm examines a new node. The adding proposition (Proposition 6.5) can be applied to the child nodes of a candidate feature subset and the replacing proposition (Proposition 6.8) can be applied to the siblings.

There are six possible relative positions of the upper and lower bounds with respect to α and β, shown in Figure 3. We now examine these possibilities and how the algorithm uses the upper and lower bounds to effectively explore the search space.

Suppose that the current candidate feature subset is $V = (X_a, X_{a+1}, \cdots, X_b)$. Denote the upper and lower bounds of the multi-information of V by $ub(V)$ and $lb(V)$ respectively.

(a) If the lower bound of V, $lb(V) > \alpha$, then the subtree of V can be pruned since any node U in the subtree containing V as its subset is not a WFS. This situation corresponds to cases (1) - (3) in Figure 3. If $lb(V) \geq \beta$ (case (1)), then $C(V) \geq \beta$ and we can check whether the second criterion of Definition 4 is satisfied, i.e., whether all subsets of V of size $(b-a-1)$ are WFSs. If the second criterion is satisfied then V is reported as an NIFS. In case (2) we need to calculate $C(V)$ and check criterion 2. For case (3), since the upper bound $ub(V) < \beta$, we have $C(V) < \beta$ and therefore V is not an NIFS.

(b) If the upper bound $ub(V) \leq \alpha$ (case (4)), then there is no need

Algorithm 1: Finding NIFSs

Input: binary dataset D, thresholds α and β
Output: all NIFSs in D.

```
1  calculate the mutual information and Hamming distances
   between each pair of features in D;
2  for each node V at the first level of the search space do
3  |  Explore(V);
4  end
```

Procedure `Explore`

Input: current candidate feature subset V

```
1  if a pair of features in V have high mutual information then
2  |  return; //pruning strategy in Section 7
3  end
4  update ub(V) and lb(V) by replacing proposition;
5  update ub(V) and lb(V) by pairwise correlation proposition;
6  if lb(V) > α then
7  |  if lb(V) ≥ β then
8  |  |  if criterion 2 of Definition 4 is satisfied then
9  |  |  |  output V;
10 |  |  end
11 |  else
12 |  |  if ub(V) ≥ β then
13 |  |  |  calculate C(V);
14 |  |  |  if C(V) ≥ β then
15 |  |  |  |  if criterion 2 is satisfied then
16 |  |  |  |  |  output V;
17 |  |  |  |  end
18 |  |  |  end
19 |  |  end
20 |  end
21 |  return;
22 else
23 |  if ub(V) ≤ α then
24 |  |  for each child node U of V do
25 |  |  |  update ub(U) and lb(U), by the adding
   |  |  |  proposition;
26 |  |  |  Explore(U);
27 |  |  end
28 |  else
29 |  |  calculate C(V);
30 |  |  if C(V) ≥ β then
31 |  |  |  goto line 15;
32 |  |  else
33 |  |  |  if C(V) ≤ α then
34 |  |  |  |  goto line 24;
35 |  |  |  end
36 |  |  end
37 |  end
38 end
```

to calculate $C(V)$ and we can directly proceed to its subtree. The pairwise correlation proposition and adding proposition can be applied to get upper and lower bounds on the multi-information for each direct child node of V.

(c) If $lb(V) \leq \alpha$ and $ub(V) \geq \alpha$ then we are in cases (5) and (6) and must calculate $C(V)$. If $C(V) \leq \alpha$, then we can proceed to its subtree and apply the pairwise proposition and the adding proposition to get the bounds for the child nodes of V. If $\alpha < C(V) < \beta$, then its subtree is pruned. If $C(V) \geq \beta$, and criterion 2 is satisfied, then V is output as an NIFS. The subtree is then pruned.

(d) The algorithm is performed in depth-first recursion. Whenever the algorithm finishes examining the current node V and its subtree, it proceeds to one of V's siblings, denoted by V'. The replacing proposition can be applied to get upper and lower bounds on the multi-information of V'.

The overall algorithm for finding NIFS given in Algorithm 1 and Procedure Explore. In Procedure Explore, through Line 1 to 3, the algorithm applies the pruning strategy described in Section 7 before performing any multi-information calculation. The remaining part of the procedure utilizes the bounds developed in Section 5 and Section 6 to prune the search space. Note that for the first node being searched, its upper bound is some random initial value greater than β and its lower bound is some random initial value smaller than α.

Note that in the worse case scenario, the algorithm has to enumerate all feature combinations, that is, the size of the search space is exponential with respect to the number of features. In Section 9, the experimental results show that our algorithm scales quadraticly with respect to the number of features, which demonstrate the effectiveness of the pruning methods discussed before. Moreover, the algorithm scales linearly to the number of observations (data points) in the datasets, since the multi-information of a feature subset can be calculated by scanning all data points once.

9. EXPERIMENTS

We use both synthetic and real-life datasets to evaluate our approach for finding NIFSs. The algorithm is implemented using Matlab 7.0.4. The experiments are performed on a 2.4 GHz PC with 1G of memory running the Windows XP operating system.

9.1 Efficiency evaluation

To evaluate the efficiency of the algorithm, we use a binary SNP data set derived from 37 strains of BXD mice. The data is available from the following website: http://www.broad.mit.edu. A SNP is a DNA sequence variation occurring when a single nucleotide in the genome differs between members of a species. SNP data associated with inbred mice are usually binary. We apply our algorithm to 100 randomly chosen SNPs from the dataset.

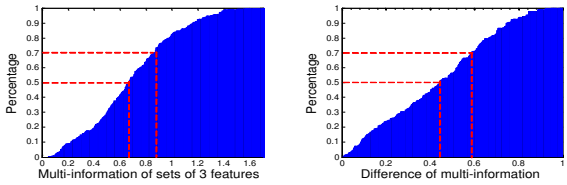
9.1.1 Runtime analysis

The default setting for efficiency evaluation is as follows. The number of features (SNPs) is 90, the number of rows is 33, $\alpha = 0.2$, and $\beta = 0.8$. When varying one of the parameters, the other ones take the default values.

Figure 4(a) shows that the runtime of our algorithm is approximately quadratic to the number of features in the dataset. This demonstrates the effectiveness of the pruning methods, as the potential search space is exponential to the number of features. Figure 4(b) shows that the runtime is linear to the number of data points (rows). The reason is that the overhead of computing multi-information for a candidate feature subset is roughly linear to the number of data points. Figure 4(c) and Figure 4(d) show the runtime when varying α and β. We observe that the effect of both parameters on the runtime is close to linear.

Figure 5(a) and 5(b) examine the effectiveness of the pruning strategies. The effects of pruning using the upper and lower bounds discussed in Section 5 and Section 6 are shown in Figure 5(a). (The pruning method discussed in Section 7 is not considered in this figure.) We observe that the bounds based on Hamming distances are more effective than the bounds developed using Han's inequality. This is because the bounds based on Han's inequality only consider the entropy of individual features and feature pairs. One the other hand, the bounds based on Hamming distances take the distance relationship between the features into consideration, and are based on the joint entropy of a set of features. Hence, they provide better pruning effects. Figure 5(b) shows the results of applying the pruning strategy based on mutual information presented in Section 7. Clearly, this strategy provides further effective pruning of the search space.

9.1.2 Tuning the parameters



(a) size 3 (b) Difference

Figure 6: Cumulative distributions of multi-information

Here we present a simple heuristic that can be used to select the parameter values of the algorithm. Based on the experiments, most of NIFSs discovered by our algorithm are of size 3. Accordingly, we explored the distribution of the multi-information of feature sets of size 3, and the distribution of the (minimum) difference between the multi information of a feature set of size 3 and its size 2 subsets. In each case, the distributions are estimated by randomly selecting 3 features from the available data 10,000 times. The resulting cumulative distribution functions (CDFs) are shown in Figures 6(a) and 6(b). Based on these CDFs, we suggest setting the value of β so that $30\% - 50\%$ sampled patterns have multi-information greater than β. The value α can be set so that the $30\% - 50\%$ of the sampled patterns have multi-information difference greater than $(\beta - \alpha)$.

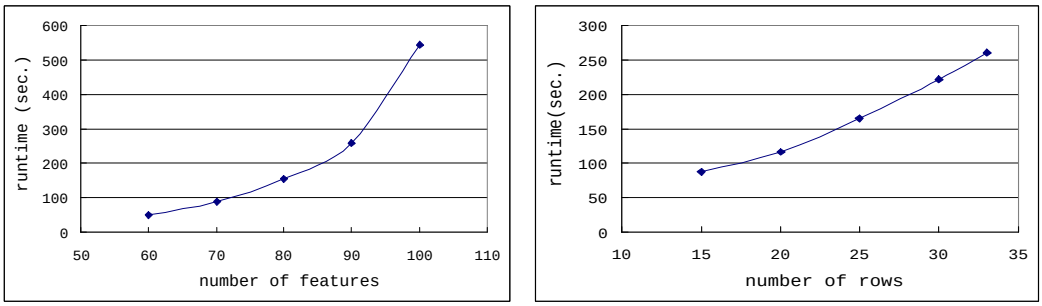
The same procedure can be used with feature sets larger than 3. Note that whenever the users are interested in NIFSs with stronger correlations, the parameters should take higher values.

9.2 Effectiveness evaluation

We use both synthetic and real-life datasets to examine the effectiveness of our method.

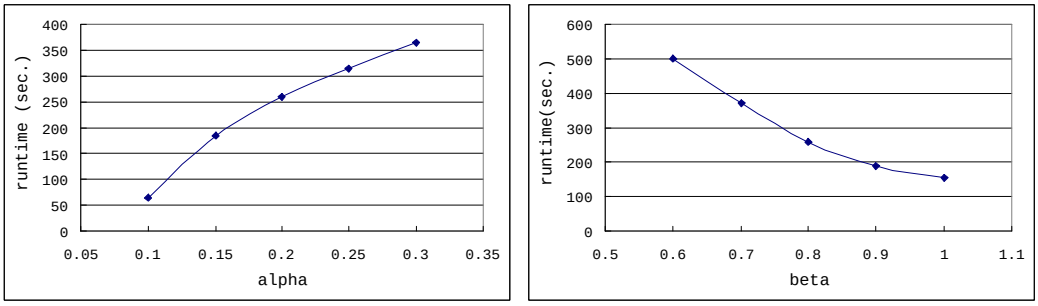
9.2.1 Finding embedded patterns

We generate a synthetic dataset of 150 points and 15 features in the following way. The dataset is first populated with randomly generated 0's and 1's (fair coin flips) for each one of the 15 features. Then we embed three patterns into the dataset. The embedded patterns are $X_{10} = X_5 \oplus X_{15}$, $X_2 = (X_4 \oplus X_8) \oplus X_{13}$,



(a) Varying number of features

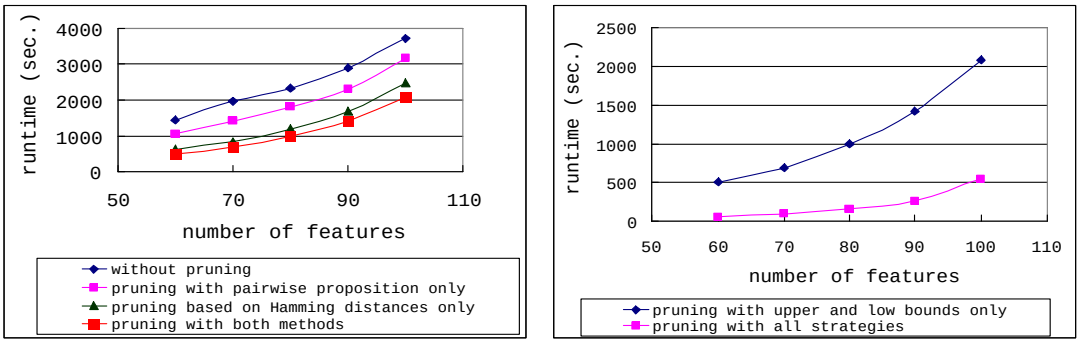
(b) Varying number of rows



(c) Varying α

(d) Varying β

Figure 4: Efficiency evaluation



(a) Pruning with bounds

(b) Pruning with MI

Figure 5: Pruning effects

and $X_3 = (X_6 \oplus X_7) \oplus (X_9 \oplus 11)$. Additional noise is added by flipping each element in the dataset with probability 0.02. We apply the algorithm on this dataset. All the three and only the three embedded patterns are identified with $\alpha = 0.2$ and $\beta = 0.8$.

For the purpose of comparison, we further examine two related approaches for finding correlation patterns. One approach is correlation pattern mining [26, 16, 13]. Another one is a feature selection method called Max-Relevance and Min-Redundancy (mRMR) [27].

9.2.1.1 *Comparison with correlation pattern mining.*

We study two commonly used interesting measurements in correlation pattern mining, i.e., all confidence and coherence [26, 16, 13]. We find that it is very hard to set appropriate thresholds for these two measurements in order to find the embedded patterns. The embedded patterns will be missed if the thresholds are too high. On the other hand, too many irrelevant patterns will be identified if the thresholds are too low. We take a closer look of these two criteria by examining their distributions. We take the pattern $X_3 = (X_6 \oplus X_7) \oplus (X_9 \oplus 11)$ for example. We enumerate all size 5 patterns and plot their all confidence and coherence distributions in Figure 7(a) and 7(b). The red line in these two figures indicate the all confidence value and coherence value of the pattern $X_3 = (X_6 \oplus X_7) \oplus (X_9 \oplus 11)$. As shown in the figures, both values of the embedded pattern are on the left tails of their distributions. Therefore, if we set the thresholds low enough to find the embedded pattern, a large number of irrelevant patterns will also be

Dependent feature	Features selected by mRMR
X_{10}	$\{X_{12}, X_{11}\}$
X_2	$\{X_{15}, X_{14}, X_9\}$
X_3	$\{X_5, X_{12}, X_7, X_6\}$

Table 2: Features selected by mRMR

identified. This indicates that all confidence and coherence are not suitable measurements for finding high order correlations.

9.2.1.2 Comparison with mRMR.

mRMR [27] is a feature selection method, which seeks a feature subset that maximizes the relevance to the dependent feature (class label). The relevance is defined as the sum of the mutual information between each selected feature and the class label. To reduce the redundancy, mRMR also tries to minimize the mutual information among the selected features. Similar to our approach, mRMR consider finding minimal redundant feature subset. However, this approach only consider pair-wise correlations.

To see if mRMR can find the three embedded patterns, we perform the following experiments. We apply mRMR three times. In each time, we take one feature of $\{X_{10}, X_2, X_3\}$ out as the dependent feature. Table 2 shows the features selected by mRMR for each dependent feature. Clearly, the selected features do not correspond to the embedded patterns. The reason is that the two criteria used by mRMR, relevance and and redundancy, are based on mutual information, which is not effective in detecting the embedded high order correlation patterns.

9.2.2 Colon-cancer susceptibility study

We apply our algorithm to real-life SNP data to find multiple SNPs that show strong correlations with colon-cancer susceptibility, for which research has shown that combining two SNPs from two candidate regions can result in strong correlation [4]. The two candidate regions we test on include 224 SNPs located from 185MB to 189MB on chromosome 1, and 188 SNPs from 119MB to 124MB on chromosome 4 [29]. The values of cancer susceptibility range from 0 to 100% indicating the probability of getting tumor, which are discretized into 5 bins. In total, there are 110 SNP pairs found showing strong associations with the phenotype when and only when the two SNPs are considered together.

9.2.2.1 Statistical significance.

We use the approach proposed in [10] to assess the statistical significance of the patterns identified by the algorithm. The statistical significance of an identified pattern F, is assessed using the KL divergence $D(P(F)||P(F))$, where $P(F)$ is the observed joint probability distribution of F, and $P(F)$ is the approximate jointed distribution of F which is derived from distributions of feature pairs in F. An example of the approximation can be found in Section 2. We bootstrap the dataset 1000 times. Each bootstrap sample is created by randomly and independently picking instances from the original dataset with replacement. We then get $P'(F)$ which is the observed joint probability distribution of F using the sample dataset. The P-value of F is $Pr(D(P'(F)||P(F)) >= D(P(F)||P(F)))$. See [10] for the theoretical background for this method. The distribution of the KL divergence between the probability mass function (pmf) of the features in an NIFS and its approximate pmf derived from a pairwise Kirkwood superposition approximation is assessed by bootstrapping samples from the dataset 1,000 times. The size of each bootstrap sample is the same as the number of original instances in the dataset.

Figure 8(a) shows the distribution of the P-values. As is clear from the figure, most of the patterns identified have very low P-values, and all of them have P-value less than 0.05. This implies the existence of interactions in the discovered patterns.

9.2.2.2 Biological evidence.

The locations of the identified pairwise SNPs are plotted in Figure 8(b). The x-axis represents chromosome 1 and the y-axis represents chromosome 4. Among the SNP pairs identified, two genes have been previously reported as colon-cancer susceptibility candidate genes. One is Dusp10, which is located on chromosome 1 from bp185,735,717 to bp185,776,892. The other one is Nfyc, which is located on chromosome 4 from bp120,262,892 to bp120,323,342. The locations of the two genes are plotted in the figure using red dotted lines. The Dusp10 protein is believed to play an active role in MAPK phosphatase activity [25]. The MAPK pathway plays an important role in colon cancers, indicating Dusp10 is a candidate gene for colon-cancer susceptibility. Moreover, [12] suggests that Nfyc is involved with transcription factor and DNA binding activity and is involved in the positive regulation of transcription through a direct assay. The remaining SNP patterns have significance levels similar to the reported genes, and may be worthy of further examination.

9.2.3 NIFSs as features in CART

Here we give an example showing how NIFSs can help with feature selection and classification. Note that feature selection method usually finds one representative feature subset. On the other hand, the goal of our method is to find all NIFSs in the dataset. Feature selection and classification themselves are a wide research areas and we do not claim that our method serves as a replacement of any of them. Our intension here is not to suggest another feature selection or classification method, but rather, to show that when the interacting features considered together, we can improve the accuracy of traditional classification and regression tree (CART) [1] methods.

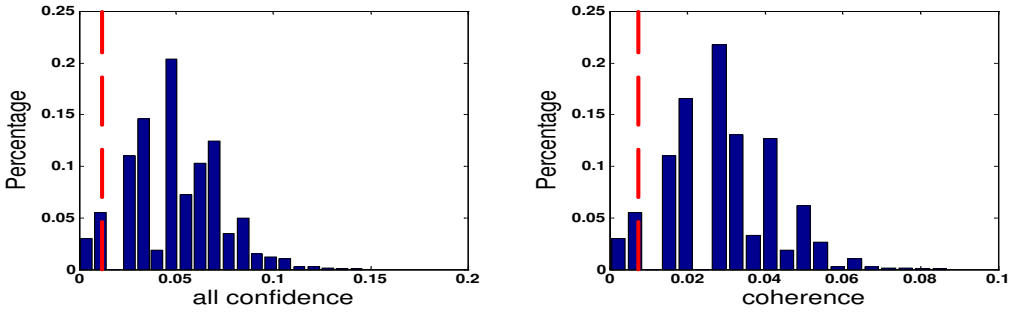
We apply our algorithm on the "zoo" dataset, which is available at the UC Irvine Machine Learning Repository. The dataset contains 101 animals, each of which has 15 binary features and a categorical class label. Among the NIFSs identified by the algorithm, we choose the top-5 NIFSs with the highest multi-information. For each one of the 5 NIFSs, we combined the features in it as a new feature, and used these 5 new features as input features of CART. For example, combining two binary features would generate a new feature having 4 values.

	All features	Selected features	Combined features
Accuracy	40.59%	73.27%	85.15%

Table 3: Accuracy of CART

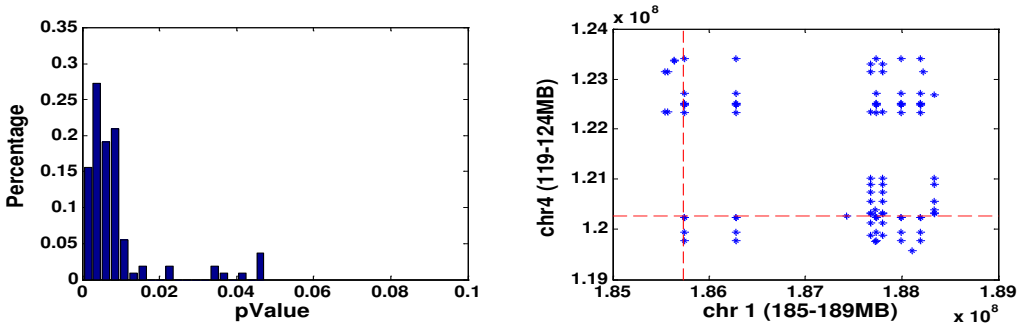
Table 3 shows the accuracy of CART measured by the percentage of correctly classified instances using 10-fold cross validation. The first column of the table shows the result using the full set of features. In the second column is the accuracy when using the original features in the 5 NIFSs without combining them. The last column shows the result when the features in each NIFS are combined as a new feature. Note that the features used for the last two columns are exactly the same. The only difference is whether these features are considered together or individually.

As we see from the results, combining features achieves the high-



(a) All confidence (b) Coherence

Figure 7: Distributions of all confidence and coherence



(a) P-values (b) SNP-pair locations

Figure 8: results on SNP-phenotype dataset

est accuracy. In the decision tree building process, CART chooses a single feature in each interaction. The results demonstrates that choosing multiple interacting features in each iteration of the tree building process can improve the accuracy of CART.

10. CONCLUSION AND FUTURE WORK

In this paper, we study the problem of finding non-redundant interacting feature subsets (NIFSs) in high dimensional binary data. We address this problem using an entropy-based correlation measurement, namely the multi-information. We study the properties of NIFSs, which enable the development of efficient algorithms. We obtain useful bounds on the multi-information using existing inequalities from information theory and additional inequalities based on the Hamming distance between two features. We also develop a pruning strategy based on mutual information which effectively prunes the search space. We evaluate the efficiency of the proposed algorithm and assess the significance of the discovered patterns using both synthetic and real-life datasets.

In real life applications, the number of features in the datasets can be very large. For example, in the study of association between genetic variations and phenotypes, the number of SNPs can be up to millions. The large number of features imposes great computational challenge because of the enormous search space of the feature combinations. Efficient algorithm for analyzing these very high dimensional datasets are in demand. How to make the algorithm scalable for such large number of features is worth further investigation.

11. ACKNOWLEDGMENTS

We thank Yuying Xie in the Department of Genetics at University of North Carolina at Chapel Hill for validating the biological evidence of the findings in the colon-cancer susceptibility study in Section 9.2.2.

This research was partially supported by EPA grant STAR-RD832720, NSF grant IIS-0448392, and a Micosoft New Faculty Fellowship.

12. REFERENCES

[1] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Monterey, Calif., U.S.A.: Wadsworth, Inc., 1984.

[2] O. Carlborg and C. S. Haley. Epistasis: too often neglected in complex trait studies? *Nature Reviews Genetics*, 5:618–625, 2004.

[3] T. M. Cover and J. A. Thomas. *The Elements of Information Theory*. Wiley & Sons, New York, 1991.

[4] P. Demant. Cancer susceptibility in the mouse: Genetics, biology and implications for human cancer. *Nature Review Genetics*, 4:721–734, 2003.

[5] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *KDD*, 2003.

[6] F. Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5:1531 – 1555, 2004.

[7] F. Geerts, B. Goethals, and T. Mielikainen. Tiling databases. *Lecture Notes in Computer Science*, 3245:278–289, 2004.

[8] T. S. Han. Nonnegative entropy measures of multivariate symmetric correlations. *Inform. Contr.*, 36:133–156, 1978.

[9] H. Heikinheimo and et al. Finding low-entropy sets and trees from binary data. In *KDD*, 2007.

[10] A. Jakulin and I. Bratko. Testing the significance of attribute interactions. In *ICML*, 2004.

[11] I. Jolliffe. *Principal component analysis*. New York: Springer, 1986.

[12] Y. Kabe, J. Yamada, H. Uga, Y. Yamaguchi, T. Wada, and H. Handa. Nf-y is essential for the recruitment of rna polymerase ii and inducible transcription of several ccaat box-containing genes. *Mol. Cell. Biol.*, 25(1):512–522, 2005.

[13] Y. Ke, J. Cheng, and W. Ng. Mining quantitative correlated patterns using an information-theoretic approach. In *KDD*, 2006.

[14] A. Knobbe and E. Ho. Maximally informative k-itemsets and their efficient discovery. In *KDD*, 2006.

[15] R. Korstanje and B. Paigen. From qtl to gene: the harvest begins. *Nat. Genet.*, 31:235–236, 2002.

[16] Y. Lee, W. Y. Kim, Y. Cai, and J. Han. Comine: Efficient mining of correlated patterns. In *ICDM*, 2003.

[17] T. Li. A general model for clustering binary data. In *KDD*, 2005.

[18] H. Liu and H. Motoda. *Feature selection for knowledge discovery and data mining*. Boston: Kluwer Academic Publishers, 1998.

[19] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on*.

[20] W. J. McGill. Multivariate information transmission. *IEEE Trans. Information Theory*, 4(4):93–111, 1954.

[21] W. Mendenhall, R. L. Scheaffer, and D. D. Wackerly. *Mathematical statistics with applications*. Duxbury Press, Boston, Mass., 1981.

[22] P. Miettinen, T. Mielikainen, A. Gionis, G. Das, and H. Mannila. The discrete basis problem. In *PKDD*, 2006.

[23] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[24] S. Monson, R. Rees, and N. Pullman. A survey of clique and biclique coverings and factorization of (0,1)-matrices. *Bull. Institute Combinatorics Appl.*, 14:1786, 1995.

[25] Y. Okazaki and et. al. Analysis of the mouse transcriptome based on functional annotation of 60,770 full-length cdnas. *Nature*, 420:563–573, 2002.

[26] E. Omiecinski. Alternative interest measures for mining associations. *IEEE Trans. Knowledge and Data Engineering*, 15:57–69, 2003.

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

[28] H. T. Reynolds. *The analysis of cross-classifications*. The Free Press, New York, 1977.

[29] C. Ruivenkamp and et. al. Ptprj is a candidate for the mouse colon-cancer susceptibility locus scc1 and is frequently deleted in human cancers. *Nature Genetics*, 31:295–300, 2002.

[30] R. Rymon. Search through systematic set enumeration. *Int'l Conf. on Principles of Knowledge Represenation and Reasoning*, 1992.

[31] E. Schneidman, M. Berry, R. Segev, and W. Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440:1007–1012, 2005.

[32] H. Tao, D. R. Cox, and K. A. Frazer. Allele-specific krt1 expression is a complex trait. *PLoS Genet.*, 2(6):e93, 2006.

[33] N. Tatti, T. Mielikainen, A. Gionis, and H. Mannila. What is the dimension of your binary data? In *ICDM*, 2006.

[34] S. Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of Research and Development*, 4:66–82, 1960.

[35] H. Xiong, S. Shekhar, P.-N. Tan, and V. Kumar. Exploiting a support-based upper bound of pearson's correlation coefficient for efficiently identifying strongly correlated pairs. In *KDD*, 2004.

[36] Z. Zhao and H. Liu. Searching for interacting features. In *IJCAI*, 2007.