

Mining Complex, Maximal and Complete Sub-graphs and Sets of Correlated Variables with Applications to Feature Subset Selection

Florian Verhein
School of Information Technologies,
University of Sydney, Australia
(fverhein@it.usyd.edu.au)

January 26, 2008

Abstract

Finding interactions between variables is a fundamental concept in Data Mining. In this work, correlations between variables are considered using Pearson's product moment correlation coefficient. Of interest are *complex*, *complete*, and *maximal* sub-graphs which describe the correlation structure between variables. This paper considers both positive and negative correlations – *complex* interactions. It is proved that under a constraint on the minimum level of correlation desired, there are useful guarantees on the structure of the correlations. In particular, the sign of the correlation between variables can be mapped to the variables themselves (i.e. to the vertices). This means that the complete complex sub-graphs can be represented as a complex set, where each element – a variable with a positive or a negative sign – is highly positively correlated with every other. This makes the interaction much easier to understand. It is also exploited to develop an algorithm that runs in the same time as if complex interactions were not considered, resulting in significantly improved scalability. Mining maximal sets of variables characterized by the *lack* of correlations is also briefly considered.

The approach is useful for examining complex correlation structures, as well as mining a representative subset of the entire data set. The latter idea is extended to the problem of *feature subset selection* in a way that gives guarantees on the minimum correlation required for features to be considered interchangeable (redundant), while guaranteeing that the selected features are not correlated with each other. Experiments show the approach performs well.

1 Introduction

Finding interactions between variables is a fundamental concept in Data Mining. This paper considers a type of correlation structure between variables as the desired

interaction. In the graph view, each variable is a vertex, and an edge exists between vertices if the magnitude of the correlation between the corresponding variables exceeds a threshold. Graphs defined by a lack of correlation are also briefly considered. The sign of the correlation (positive or negative) is also taken into account and the edge labeled accordingly.

What properties should sub-graphs of interest have? In this work, *completely connected* sub-graphs (cliques) are of interest. The reasoning behind this is quite straightforward: the resulting sub-graph has the guarantee that each variable is highly correlated with each other variable in the sub-graph, therefore describing a very strong symmetric relationship between all variables. An application is that one variable could be used in place of the entire sub-graph. In this application, being completely connected is useful as the user may define a level of correlation over which the variables are considered to be equivalent – or more precisely; of insufficient difference to warrant inclusion of more than one of them.

An important consideration is the inclusion of positive and negative correlations. If only positive correlations or high magnitude correlations are considered, much of the structure will be missed. That is, potentially important negative correlations will not be found. For example, A may be highly correlated with D , but both of these may also be *negatively correlated* with B and C . The goal is to mine complete and *complex* sub-graphs – that is, allowing positive and negative relationships – that describe such a structure. Furthermore, the goal is to be able to represent these as *complex sets* of variables – sets of variables that may include negated variables – and that are all highly positively correlated with each other. For instance, the complex set $\{A, -B, -C, D\}$ indicates that A , $-B$ (negative B), $-C$ and D are highly *positively* correlated. Without consideration for complex relationships, either a) two

separate sets $\{A, B\}$ and $\{C, D\}$ would be mined instead or b) the set $\{A, B, C, D\}$ would be mined – in both cases failing to show the complete structure of the correlations.

In part, this work shows that under a practical constraint on the correlation coefficient, mining sub-graphs with positive and negative edges can be reduced to mining complex sets of variables, as a majority of the edge combinations are impossible. Furthermore, the positive and negative labeling of variables in the sets can be achieved completely for free. More specifically, suppose there is a complete sub-graph on the variables $V' \subseteq V$, where V is the set of all variables in the data set. There are $|V'|^2/2$ edges in the complete sub-graph and therefore $2^{|V'|^2/2}$ possible labellings of edges as either positive or negative. In other words, there are $2^{|V'|^2/2}$ different *complex* correlation structures for the *single* complete set of variables V' . However, the results in this paper show that under the constraint, only $2^{|V'|}$ of these are possible. This has a number of consequences. First, this is precisely the number of labellings of *vertices*, which means that instead of mining and reporting entire sub-graphs (i.e. including edge labels) and incurring the correspondingly higher complexity, only *complex sets* of variables must be mined and reported. That is, the same information represented by a *complex complete sub-graph* can be represented by a *complex set*. Furthermore, of the $2^{|V'|}$ possibilities, half are simply the negation of all variables in another combination, leaving $2^{|V'|-1}$ configurations that are of practical interest. Finally, the way in which the algorithm works provides the labellings completely for free: searching through all the possible subsets of all vertices V takes $O(2^{|V|})$ time, but the algorithm presented here *also labels the variables within this time*. For comparison, note that if any combination of positive and negative variables are possible, and excluding combinations having both positive and negated versions of the same variable, there would be $O(2^{2|V|})$ different complex sets – the complexity of the algorithm *squared*. Furthermore, if the edge labellings could not be mapped to vertices, then it would not be possible to report sets and there would be $O(2^{|V|^2/2})$ different sub-graphs.

Simply mining these complex sets creates the problem of redundancy, as each set of size k will contain $2^k - 1$ subsets. Consequently, this work focuses on *maximal* sets (maximal complete sub-graphs).

What is the **motivation** for all this? Each maximal complex set of variables indicates that all the variables in that set are highly positively correlated with every other variable. Recall that variables may be negated in a complex set. Furthermore, no other variable (or its negation) can be added to the set without breaking

this property. For a start, such correlation structures are interesting in their own right and can indicate near duplicate variables or flag previously unknown interactions. By comparison, analysing or graphing a correlation matrix usually hides interactions that involve more than two variables at a time. Each maximal complex set can also be thought of as capturing an underlying feature, or ‘factor’ in the process captured by the data set. Of course, there are other approaches for doing this, namely Principle Component Analysis (PCA) [3], Singular Value Decomposition (SVD) – which is related to PCA – and Factor Analysis [3] – which uses PCA. Each principle component is considered to capture a source of variability in the data – that is, a factor. While it is possible to examine the coefficients of a principle component in order to determine what variables are associated with it, it is a technique that does not provide the type of guarantees on the correlation structure that the approach in this paper does. It also becomes difficult to do when many variables are involved.

The advantages of the proposed technique are that it gives guarantees on the correlations in a set, it maintains the actual variables, and the resulting patterns are easy to interpret.

Another advantage is that it can provide suggestions for selecting a representative set of features. It is therefore applied to the problem of **feature subset selection** [8] using a three stage *filter* [8] approach: First, maximal sets of variables (features) are mined. The variables in such a set are considered interchangeable, as they are highly correlated with each other. Then, a representative variable for each maximal set is found, taking account the overlap between such sets. This is intended to remove from consideration any redundant, duplicate, or otherwise unnecessary variables while capturing the primary factors in the underlying process. Finally, a subset of the representative variables is chosen so that none of them are correlated with each other.

The approach allows the user to define the minimal correlation required for features to be considered interchangeable, and provides a guarantee that the features selected will not be correlated. Another advantage of this approach is that a *subset* of the original features are used as selected features. This means models such as trees and rules built on these remain highly interpretable. This contrasts approaches such as PCA or SVD which produce features that are linear combinations of *all* original features. These make the resulting models very difficult to interpret as the original features are lost. Furthermore, they do not reduce the number of attributes that need to be *collected* in future: The principle components are only orthogonal if the linear combination is not truncated. This means that while

the algorithm uses fewer features, the features used are still a function of *all* the original features.

The **contributions** of this paper are as follows:

- *Complete, complex and maximal sub-graphs (sets)* of correlated variables are considered useful patterns for describing complicated correlation structures in an easily understood manner.
- It is proved that under a constraint on the minimum correlation desired, there is a specific structure on the correlations between variables that allows edge relationships to be mapped to the vertices, and thus allows *complex sets* to capture the same information as *complete complex sub-graphs*.
- An algorithm is developed that exploits the above theory in order to mine all complex maximal sets of variables. This is a data mining technique, where the patterns mined highlight interesting and complex interactions between variables that would otherwise be hidden. Experiments show the algorithm is very efficient at mining such sets, due also in part to the extensive pruning it employs.
- The approach is further developed for mining a representative subset of the variables. In particular, for the feature subset selection problem. As a result, an unsupervised feature subset selection method is proposed. Experiments on the UCI cardiac arrhythmia data set show that it performs well.

The remainder of the paper is organised as follows: Section 2 presents the theory, Section 3 describes the data mining algorithm, Section 4 describes the feature subset selection algorithm, Section 5 provides experimental results and Section 7 concludes this paper.

2 Complete, Complex Variable Sub-graphs, Sets and Correlation

Recall that the graph on the variables was defined as follows: each variable is a vertex, and there is an edge between vertices if the corresponding variables are correlated. Specifically, given a threshold t , an edge exists between two variables A and B if $|\rho_{A,B}| \geq t$. The weight of the edge is $\rho_{A,B}$ and of specific interest is whether $\rho_{A,B}$ is positive or negative – which is called the *label* of the edge. Later, the problem of mining uncorrelated sets is considered, where $|\rho_{A,B}| \leq t$.

Pearson’s correlation coefficient between two random variables A and B is

$$\rho_{A,B} = \frac{\text{cov}(A,B)}{\sigma_A \sigma_B} = \frac{E((A - \mu_A)(B - \mu_B))}{\sigma_A \sigma_B}$$

If the data is centered, that is, $E(A) = E(B) = 0$, then

$$\rho_{A,B} = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} = \text{corr}(\vec{a}, \vec{b})$$

where \vec{a} and \vec{b} are the vectors of samples for the variables A and B . In this work, $\text{corr}(\vec{a}, \vec{b})$ is used, and the data is assumed to be centered¹. The use of the dot product also means that the kernel trick is applicable – potentially allowing non-linear correlations to be used.

Recall that the goal is to mine *complete, complex* and *maximal* sub-graphs of variables, and to be able to represent these as *complex* and *maximal sets*. Recall that a sub-graph is *complete* if it is completely connected. A set will only ever be used to describe a *complete* subgraph. Recall that the term *complex* is used to describe the inclusion of negative and positive relationships – that is, positive and negative labelings of edges or variables. Recall that a complete sub-graph is called *maximal* if no other complete sub-graph subsumes it. Equivalently, a set is *maximal* if no super-set exists.

Section 2.1 considers the problem of mining maximal and complex sets of highly correlated variables – which is the focus of this paper. Section 2.2 briefly considers the problem of mining *uncorrelated* variables.

2.1 Highly Correlated, Complex Variable Sets

In this Section, the theory required to mine highly correlated, complex variable sets is developed.

LEMMA 2.1. $\text{corr}(\vec{a}, \vec{b}) > t \wedge \text{corr}(\vec{b}, \vec{c}) > t \wedge |\text{corr}(\vec{a}, \vec{c})| > t \implies \text{corr}(\vec{a}, \vec{c}) > t$ if and only if $t \geq 0.5$. In other words, if (\vec{a}, \vec{b}) are highly positively correlated and (\vec{b}, \vec{c}) are highly positively correlated and (\vec{a}, \vec{c}) are highly positively or negatively correlated, then (\vec{a}, \vec{c}) are in fact highly positively correlated. In this case, “highly” means with a correlation coefficient above 0.5.

Proof. Without loss of generality, assume $\{\vec{a}, \vec{b}, \vec{c}\}$ are all unit vectors (this does not change the correlation: $\frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} = (\frac{\vec{a}}{\|\vec{a}\|} \cdot \frac{\vec{b}}{\|\vec{b}\|}) / (\|\frac{\vec{a}}{\|\vec{a}\|}\| \|\frac{\vec{b}}{\|\vec{b}\|}\|)$). Then $\text{corr}(\vec{a}, \vec{b}) = \vec{a} \cdot \vec{b}$ – the dot product. The following identity is used: $\sum_i (a_i + c_i - b_i)^2 = \sum_i [(a_i^2 + b_i^2 + c_i^2) + 2(a_i c_i - b_i c_i - a_i b_i)] = 3 + 2(\vec{a} \cdot \vec{c} - \vec{b} \cdot \vec{c} - \vec{a} \cdot \vec{b})$. The last equality follows as the vectors are unit vectors (i.e. $\|\vec{a}\| = 1 \implies \sum_i a_i^2 = 1$). Using the thresholds $\vec{a} \cdot \vec{b} > t$ and $\vec{b} \cdot \vec{c} > t$ and the fact that $\sum_i (a_i + c_i - b_i)^2 \geq 0$ we have: $0 \leq 3 + 2(\vec{a} \cdot \vec{c} - \vec{b} \cdot \vec{c} - \vec{a} \cdot \vec{b}) < 3 + 2\vec{a} \cdot \vec{c} - 4t$.

To avoid a contradiction we must therefore have $\vec{a} \cdot \vec{c} \geq 2t - 1.5$. If $\vec{a} \cdot \vec{c} < -t$ then $-t > 2t - 1.5 \iff$

¹Centering the data is not necessary, and this is sometimes preferred in practice, but in that case it does not equal $\rho_{A,B}$.

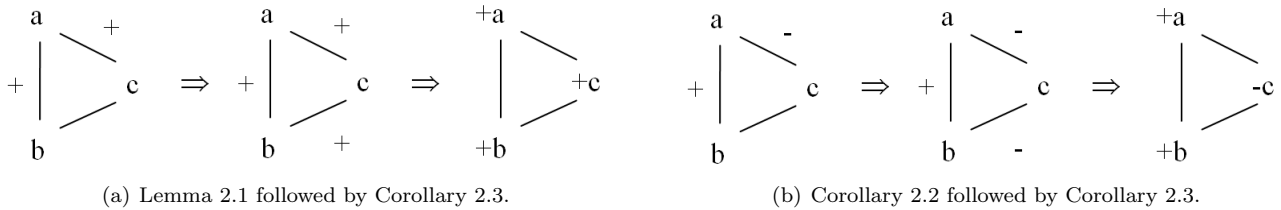


Figure 1: Simple Example of the Lemma and Corollaries for sub-graphs of size 3. Recall that an edge exists between two variables a, b if $|corr(a, b)| \geq t$. It is assumed $t \geq 0.5$ so the Lemma and Corollaries apply. In the first step (implication) in (a), Lemma 2.1 is applied. In the first step in (b), Corollary 2.2 is applied. The second step of both (a) and (b) is the application of Corollary 2.3, choosing a as the arbitrary $+$ variable. Hence, the relationships can be represented as the complex sets $\{a, b, c\}$ for (a) and $\{a, b, -c\}$ for (b).

$t < 0.5$. Therefore, when $t \geq 0.5$, $a \cdot c < -t$ provides a contradiction and therefore we must have $\vec{a} \cdot \vec{c} > t$.

In the reverse direction, we have $\vec{a} \cdot \vec{c} > t$ (as the implication is true). Suppose for the purpose of a contradiction that $t < 0.5$. Then we can see from $\vec{a} \cdot \vec{c} \geq 2t - 1.5$ that it is possible to have $\vec{a} \cdot \vec{c} < -t$ – providing the contradiction (for example substitute any value $t < 0.5$). \square

A Corollary follows immediately:

COROLLARY 2.2. $corr(\vec{a}, \vec{b}) > t \wedge corr(\vec{b}, \vec{c}) < -t \wedge |corr(\vec{a}, \vec{c})| > t \implies corr(\vec{a}, \vec{c}) < -t$ if and only if $t \geq 0.5$

Proof. Replace \vec{c} with $-\vec{c}$ in Lemma 2.1. \square

These are illustrated graphically in the left hand implications of Figures 1 (a) and (b).

These results mean that given a complete complex sub-graph of size three, the sign of the third edge can be obtained from the sign of the other two, simply by multiplying them together. Since this works for any triple in a complete sub-graph, this can be extended to the entire sub-graph. Furthermore, it allows the signs of the edges to be mapped to the variables themselves. The following Corollary describes this:

COROLLARY 2.3. *If $t \geq 0.5$, then relationships between variables in a complete sub-graph can be assigned to the variables themselves (without loss of information) using the following procedure: Select an arbitrary variable a and label it $+$. Then, for each other variable b in the subgraph, label it according to the sign of its correlation to a . All relationships between two variables can be inferred (reconstructed) from their labeled sign: if they have the same (different) sign, they have a positive (negative) correlation.*

Proof. In the procedure, every variable $b \in V : b \neq a$ will clearly be assigned only one sign. It suffices to

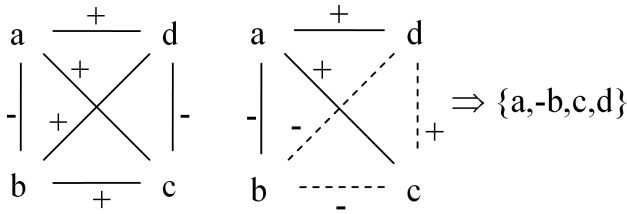
show that after this has been done, the reconstruction of edge signs works. Consider two variables $b \neq a$ and $c \neq a$. By the construction, the sign of their correlation with a is known. The sign of $corr(b, c)$ can therefore be determined by Lemma 2.1. By considering all such pairs (b, c) , every edge’s sign can be constructed. \square

Actually, there are exactly two ways of labeling every complete complex sub-graph, both of which express exactly the same edge relationships. In Corollary 2.3, a may be arbitrarily labeled $-$ (instead of $+$), which simply flips all the other signs also. Of course this would be redundant, hence only one is used. In the algorithm, an arbitrary order is imposed on variables and the greatest variable in a sub-graph is arbitrarily chosen to be $+$. A simple example is shown in Figure 1.

Corollary 2.3 means that the sign of the edges between variables in the graph can be assigned to the variables themselves. This has two important consequences:

- *Complex sets* completely describe the relationships. This means that with the assigned signs, every variable in a complex set is highly positively correlated with each other variable in the set. This makes the structure very easy for the user to understand as a set is a much simpler construct than a graph.
- The search space of the mining algorithm is significantly decreased, as the problem is reduced to mining sets of variables, rather than sub-graphs. The relevant part of the algorithm presented actually takes $O(2^{|V|})$ time, precisely the same as only enumerating sets – that is, not considering complex interactions.

Observe that the inclusion of negative correlations only makes sense if the above Lemma holds – otherwise we cannot assign the direction of the correlation between variables to the variables themselves. For example, when $t < 0.5$ it is not possible to report a set of variables



(a) A configuration like this is possible only when $t < 0.5$. It cannot be mapped to a complex set.

(b) When $t \geq 0.5$, any complete complex sub-graph can be mapped to a complex set. The signs on the dotted edges can be inferred from the others.

Figure 2: Example of Corollary 2.3

such as $\{a, -b, c, d\}$ with the interpretation that these four variables are highly positively correlated with each other. When $t < 0.5$ it is possible that $\text{corr}(a, b) < -t$, $\text{corr}(a, c) > t$ but $\text{corr}(b, c) > t$. In this case there is no labeling of variables that can produce a set so that each element is positively correlated with the others. Therefore, complex sets are not meaningful when $t < 0.5$. The reader may like to try this on the example in Figure 2(a). Following the procedure of Corollary 2.3 does not work as the edges cannot be reconstructed, so it is impossible to map the complex sub-graph it to a complex set. The theory shows this situation is only possible when $t < 0.5$. On the other hand, the example in Figure 2(b) does work and demonstrates the procedure.

Therefore, if $t < 0.5$ it makes little sense to consider complex interactions as they cannot be mapped to sets and are therefore of limited use as they are too complicated to understand in general. In such cases only the existence of *any* correlation is of interest, in order to report variables as sets. Although in that case it is not possible to state that they are all highly positively correlated with each other. It can only be stated that the *magnitude* of the correlation is high.

In Section 3, a method of enumerating the possible sets will be presented that, in conjunction with Corollary 2.3, means that all complex and complete variable sets can be mined and labeled in $O(2^{|V|})$ time – the same complexity as without considering the sign of the correlations. For comparison, note that a naive approach would be to enumerate possible sets, and for each, apply Corollary 2.3. This would require $O(|V| \cdot 2^{|V|})$ time due to the $O(|V|)$ operations used for labeling the extra $O(|V|)$ edges added whenever another variable is added in the search.

2.2 Uncorrelated Variable Sets

An interesting but simpler problem is to find maximal

sets of variables that are pairwise uncorrelated, in the sense that the absolute correlation is below a threshold. That is, an edge exists between two variables A and B if $|\text{corr}(A, B)| \leq t$, where t is a (usually small) threshold. This mines sets of uncorrelated variables. Of course, complex relationships don't make sense for these.

3 Mining Complex Maximal Sets: Algorithm

In order to make the algorithm easy to understand, a *Prefix-Tree* will be used to help describe it and prove properties. Without loss of generality, assume the variables are integers $V = \{1, 2, \dots, n\}$. A complete *Prefix-Tree* can be constructed as follows: First, an arbitrary but fixed order is chosen on the variables – in this paper I will use ascending order. Each node has a label corresponding to a variable $v \in V$. The root node is special, and is labeled with ∞ .

The tree is constructed so that each node can only have a parent with a label *greater than* its own label. In the algorithm, a node has a reference to its parent (but not to its children). Each node in the *Prefix-Tree* represents a distinct subset of the variables, represented as a *sequence* in decreasing order, which may be constructed by traversing toward the root. The root corresponds to the empty set (in the algorithm, ∞ is treated as a constant). Whenever a set is mentioned henceforth, it is assumed to be represented as a sequence in decreasing order. A complete *Prefix-Tree*, an example of which is shown in Figure 3, clearly has $2^{|V|}$ nodes. A complete *Prefix-Tree* is the worst case search space for mining the sets of variables, as the algorithm is an enumeration approach.

The algorithm effectively works by performing a depth first traversal of the search space, expanding sibling nodes in *increasing* order – which is important

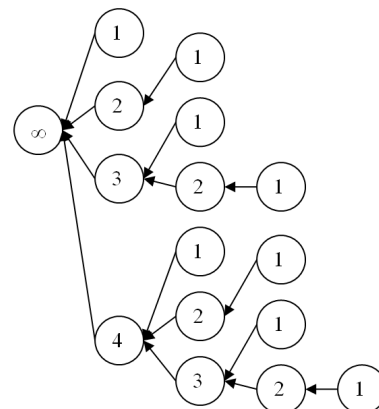


Figure 3: A complete *Prefix-Tree* for $V = \{1, 2, 3, 4\}$. Each node represents a *complete* set of variables.

as described later – and pruning the search as soon as possible.

Specifically, the following properties are exploited. Here, a set is called complete if the corresponding sub-graph is complete. Elsewhere in the paper this is implicit.

1. Whenever a new variable v_2 is considered to be added to a complete set C , and v_2 is not highly correlated with *each* variable in C , then neither $C \cup v_2$ or any super-set of $C \cup v_2$ can be complete. That is, the corresponding sub-graphs will also be missing at least one edge. *One* consequence of this is the following: Since by construction $v_2 < v_1 \forall v_1 \in C$, the entire sub-tree rooted at the node corresponding to $C \cup v_2$ may be pruned. The case $C = \emptyset$ holds trivially by defining it as complete.
2. When checking whether a new variable v_2 can be added to a complete set $C \cup v_1$, the algorithm only needs to consider those v_2 for which $C \cup v_2$ is complete, by property 1. That is, if $C \cup v_2$ is not complete, then neither can its super-set $C \cup v_1 \cup v_2$ be. Now, if $C \cup v_1$ and $C \cup v_2$ are complete, then $C \cup v_1 \cup v_2$ is complete if and only if $v_1 \cup v_2$ is complete (that is, if and only if v_1 and v_2 are highly positively or negatively correlated). The reason for this is straightforward: the only edge that can be missing in the sub-graph defined by $C \cup v_1 \cup v_2$ is (v_1, v_2) , as the existence of all the other edges has already been established. Translated to the *Prefix-Tree* and the algorithm, this means that only *siblings* need to be considered – note that $C \cup v_1$ and $C \cup v_2$ will become siblings in the *Prefix-Tree*, with common prefix C . The algorithm is said to progress by *joining siblings*.
3. The above two properties also work in combination. If $C \cup v_2$ is not complete, then neither can $C \cup v_1 \cup v_2$ be. By never creating the node for $C \cup v_2$ (recall this part of the search space is pruned), $C \cup v_1$ will have one less sibling that must be considered.

In Algorithm 1, Properties 2 and 3 are achieved using the *newSiblings* list, which is used as the *siblings* list for expanding new child nodes in the depth first search. Property 1 is achieved by not adding the corresponding node or expanding the search (no recursive call). Note that the for loop in Algorithm 1 traverses the siblings in increasing order.

It can be of use to report the minimum correlation between any pair or variables in a set. This is useful, as it provides a bound that is generally higher than t . This can be achieved by storing the minimum at the corresponding node, and computing the new minimum

for a new node as the minimum over the siblings and the additional link.

Note that the algorithm works by growing sets, and using heavy pruning. This approach is appropriate when the graph of correlations is sparse – precisely what happens when high correlations are desired.

3.1 Complex Sets

Finally, the only thing left in the search part of the algorithm is to label the variables. Accordingly, each *Prefix-Tree* node also has a sign associated with it – either $+$ or $-$. The sign corresponds to the relationship that the node’s variable has to the *first* node in the sequence – the node whose parent is the root.

Without loss of generality², the children of the root are labeled $+$. The sign of a new node is calculated as follows. When joining the siblings corresponding to $C \cup v_1$ and $C \cup v_2$, the sign of $C \cup v_1 \cup v_2$ is the sign of $C \cup v_1$ multiplied by the sign of the correlation between v_1 and v_2 . This is a direct consequence of Lemma 2.1 and Corollary 2.2 applied to the variables v_1 , v_2 and x , where x is the *first* node in the sequence (the first element of C). Note that this is the application of the procedure in Corollary 2.3. Furthermore, by that Corollary, the signs of the relationships between any of the variables can be derived from the sign of the node (variable). When the sets are output, the sign also becomes the sign of the variable.

The case of complex relationships when $t < 0.5$ is not covered. As discussed earlier as a consequence of Lemma 2.1, it makes little sense as the positive and negative correlations cannot be mapped back to the variables. The algorithm can still be used, but the labels do not capture all the relationships and should therefore simply be ignored.

3.2 Maximal Complex Sets

The algorithm must also calculate the *maximal* complex sets. It does this by maintaining the current maximal sets, and as new sets are added, deleting any subsets. Labels can be ignored during this process. The following Lemma makes this much easier.

LEMMA 3.1. *Subsets of a set represented by a node currently being examined can only occur in a part of the tree that has already been examined by the algorithm.*

Proof. By construction, the algorithm progresses through the search space by joining existing sets together (“joining siblings”), creating sets that are one variable larger than the two original sets. Suppose for

²There are two equivalent labellings for variables in complex sets – just flip the sign of each variable.

the purpose of contradiction that a set S exists that is a super-set of a set mined later. Proceed by showing that each subset of size $|S| - 1$ has already been mined, so that the result follows by induction on S (the base case is trivial). The immediate subsets of S can be obtained by removing one element (variable) at a time. Suppose $x \in S$ is removed, so that $S = S_p \cup x \cup S_s$ where S_p and S_s are the prefix and suffix (either potentially empty) respectively. Since the expansion of the search is done in depth first fashion and with increasing order amongst the siblings, $S_p \cup S_s$ must be expanded first, since by definition the sequences in the *Prefix-Tree* appear in *decreasing* order. Since this is true for all $x \in S$, the result follows by induction and contradiction. \square

Note that this is why the order of expansion of siblings is important. More specifically, maintaining a *consistent* (but possibly arbitrary) order is important.

The algorithm only updates the *maximalSets* list with sets (nodes) that are known to be maximal *so far* and *in the near future* in the search. The first constraint is trivially met by Lemma 3.1. The second constraint is met by adding those sets (nodes) that have no children when that path is complete, as such a set may only be a subset of a node on a *different path* of the search, which occurs *later* (that is, only after the current path is completed). Because of Lemma 3.1, new maximal sets can only replace existing ones, and therefore only sets that have been mined earlier must be checked for being subsets of a new one.

Finally, note that since the *Prefix-Tree* shares as many nodes as possible, the space of the collection of maximal sets is minimized since prefixes of the stored maximal sets are shared.

Considering all of the above, the resulting algorithm can be written surprisingly simply – especially in recursive form as shown in Algorithm 1.

3.3 Mining Uncorrelated Sets

In order to mine sets where each variable is uncorrelated with every other, Algorithm 1 is modified as follows. “ $|corr(v_1, v_2)| \geq t$ ” in *mine*(, ,) is replaced with “ $|corr(v_1, v_2)| \leq t$ ”, and “return 1” in *corr*(,) is replaced with “return 0”. The *sign* of the variables should also be ignored, as they cannot represent all relationships.

However, it should be pointed out that data sets generally have *many* uncorrelated variables, so using the enumeration approach of Algorithm 1 is not the most practical method. Algorithm 1 is designed for mining sets defined by high correlations, as this allows it to take maximum advantage of the pruning abilities described earlier.

Algorithm 1 Simplified algorithm for mining complete and maximal sets when $t \geq 0.5$. The algorithm assumes a garbage collector, or an alternative approach to delete nodes in the *Prefix-Tree* that are no longer required.

Input:

double *corr*[][] //precomputed correlation matrix
double *t* //correlation threshold, $t \in [0, 1]$

Output:

maximalSets //complete, maximal sets
 //as *PrefixTree* nodes

Data Type:

Node(*Node* *parent*, *int* *v*, *int* *sign*)
 //nodes in the *PrefixTree*

List(*int*) *V* = [1, 2, ..., *corr*[0].length] //variables

List(*PrefixNode*) *maximalSets* = \emptyset

mine(*V*, \emptyset , *Node*(*null*, ∞ , 1)) //

mine(*List*(*int*) *siblings*, *List*(*int*) *newsiblings*, *Node* *n*)

int *v*₁ = *n.v*

boolean *hasChild* = *false*

for each (*int* *v*₂ in *siblings*)

if ($|corr(v_1, v_2)| \geq t$)

Node *nn* = *Node*(*n*, *v*₂, *n.sign* * *corr*(*v*₁, *v*₂))

mine(*newsiblings*, \emptyset , *nn*) //recursive, DFS

newsiblings.add(*v*₂) //new sibling was created

hasChild = *true*

else //no need to expand search

if (!*hasChild*) //super-set known to exist

addCompleteSet(*n*) //n is maximal so far

addCompleteSet(*Node* *n*)

for each (*Node* *n*₂ in *maximalSets*)

if (*n*₂ subsetof *n*) //simple linear traversal

maximalSets.remove(*n*₂) //not maximal

maximalSets.add(*n*)

double *corr*(*int* *v*₁, *int* *v*₂)

if (*v*₁ = ∞) return 1 //root of *PrefixTree*.

else return *corr*[*v*₁ - 1][*v*₂ - 1]

4 Selecting a Representative Set: an Application to Feature Subset Selection

Recall that maximal sets of correlated variables can be presumed to capture sets of variables that are interchangeable with each other and therefore can be represented by one member of the set. In this Section, this idea is developed for the purpose of feature subset selection. The goal is to select variables in such a way that they “cover” (represent) the original dataset, but at the same time are not correlated with each

other. The primary complication is the overlap between maximal sets of variables, which requires some care. The approach is as follows:

1. Mine all maximal sets, where variables are connected if $|corr(A, B)| \geq t$, using Algorithm 1. Call the result – a set of such sets – M . Note that a set containing a single variable may be maximal. Clearly, all variables will be present in at least one element of M and in that sense, the data set is completely “covered”.
2. Select a representative variable from each maximal set $C \in M$. This is a two step procedure, complicated by overlap between elements of M :
 - (a) Recall that the weight of each edge (v_i, v_j) is the correlation between the variables. For each $C \in M$ select the representative variable $v \in C$ as follows, breaking ties arbitrarily:

$$v = \arg \max_v \left(\sum_{v_j \in C} |corr(v, v_j)| \right)$$

In other words, the most central variable is chosen, measured by it being the most correlated with all the other variables in the set C . The variable v is taken to represent the other variables and to capture the underlying factor of the set. The remaining variables $C - \{v\}$ are assumed to be redundant.

- (b) Due to the frequent overlap between the $C \in M$ (different maximal sets often share a common subset), it is not possible to treat each C in isolation, as a redundant variable in one maximal set may be the representative (non-redundant) variable of another – overlapping – maximal set. The problem with this is that two or more variables that are in fact in the same maximal set (and therefore correlated with each other) can be chosen.

To partially remedy this, assign each variable $v \in V$ an integer weight. When considering each $C \in M$ as above, the chosen variable $v \in C$ has its weight incremented by the number of variables it replaces in C – that is, $|C|$. Every other (redundant) variable $v' \in C - \{v\}$ has its weight *decremented* by $|C| - 1$. Note that a variable may be determined to be a representative (redundant) variable for some $C \in M$, but a redundant (representative) variable in other C (’s).

Only variables with a positive weight after the procedure has completed are retained. This

means that a variable is only retained if it is more representative than non-representative, measured by the number of variables it represents minus the number of variables that it does *not* represent. The reason for decrementing by $|C| - 1$ rather than C is to avoid variables “canceling” each other out when representing two overlapping sets of equal size.

Call the resulting set of variables V_c . Generally, V_c contains fewer variables than V and so the number of features has been reduced. However, V_c is only considered a candidate set of selected features, as it’s elements may still be correlated with each other. This can occur, for example, when two sets of equal size overlap, or when two sets are connected to each other. In the latter case they don’t overlap, but some elements of one set may be correlated with elements of the other. This is undesirable, as the selected variables should not be correlated with each other.

3. This step ensures that none of the selected variables are correlated with each other. First, the “cumulative sum” of correlations is computed for each variable:

$$cum_sum(v) = \sum_{C \in M} \sum_{v_j \in C, v_j \neq v} |corr(v, v_j)|$$

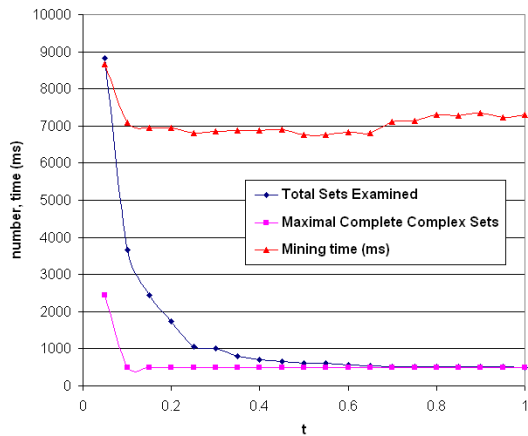
Note that this can be done as part of step 2a. A variable with a higher cumulative sum is more representative, and therefore is more desirable. This is used to decide between pairs of correlated variables. The procedure is as follows;

Loop through each $v \in V_c$, and check if it is correlated with another variable $v' \in V_c$. If not, add v to V_s . If it is, add it to V'_c *if* the cumulative sum of it’s correlations (as described above) is higher than that of v' . Set V_c to V'_c and repeat the procedure until V_c is empty. The final set of selected variables is V_s .

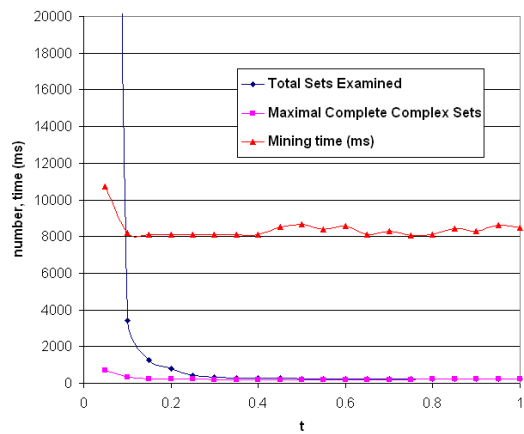
Note that complex relationships are not applicable for feature selection. That is, of interest is only whether there is a high correlation – the sign of the correlation is irrelevant. Therefore, Algorithm 1 can be used as Step 1 of the feature selection procedure **for any value of t** .

Note that this is an unsupervised approach. If a variable to be predicted is present, it must be removed from $|V|$ prior to applying the procedure.

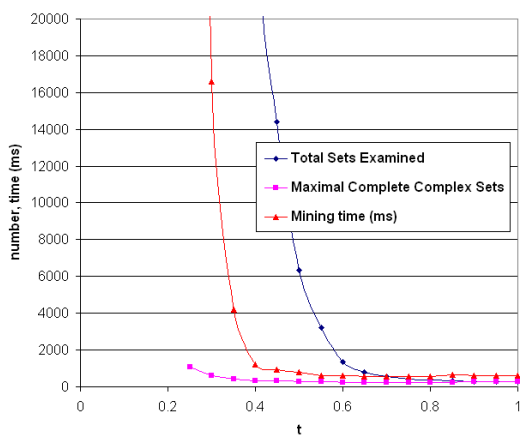
The approach ‘covers’ the data set, in the sense that every variable is taken into account by the final



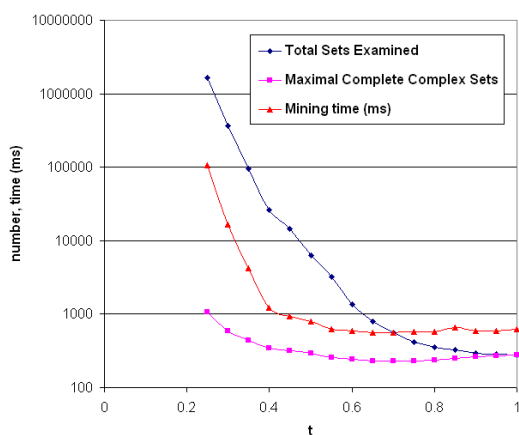
(a) MADELON Dataset



(b) SYLVA dataset



(c) Arrhythmia Dataset



(d) Arrhythmia Dataset, vertical axis in log scale

Figure 4: Run time Results. Total Sets Examined is the exact number of sets that the search has examined. That is, the size of the space examined. Maximal Complete Complex Sets is the number of such sets mined. Mining time is the run time of the entire algorithm in milliseconds.

selected set – *provided that* this does not lead to selected attributes being correlated with each other.

The threshold t functions in two ways: First, it allows the user to define the minimum correlation magnitude between variables that signifies that variables can be considered redundant. Secondly, no variables in the final selected set will be correlated with each other (have a correlation magnitude greater than t).

The technique therefore generates a *representative* subset of the original variables while guaranteeing that the selected variables are uncorrelated.

An advantage of this feature selection approach, in addition to the guarantees provided on the correlations and redundant features, is its simplicity.

5 Experiments

An implementation of Algorithm 1 is first evaluated on some large data sets for the purpose of run time analysis. Then, the approach is applied to feature selection using the technique described in Section 4.

5.1 Run Time Performance

Experiments were performed on three data sets: MADOLEN, SYLVA and Arrhythmia. The MADOLEN data set was obtained from [6] and SYLVA was obtained from [10]. The datasets were part of feature selection and performance prediction challenges respectively. No pre-processing was done on them and the “training data” sets were used. The Arrhythmia data set was obtained from the UCI repository [1], and all missing values replaced by the mean of the correspond-

Algorithm	Accuracy, original data set	Accuracy, after PCA	Accuracy, reduced data set	Accuracy improvement over original data set	Accuracy improvement over using PCA
J48	76.99	66.81	65.04	-11.95	-1.77
J48graft	78.32	67.48	65.71	-12.61	-1.77
NaiveBayes	76.99	71.68	72.12	-4.87	0.44
IBK, K=5	63.72	57.74	63.94	0.22	6.19
IBK, K=1	63.72	57.30	62.39	-1.33	5.09
DecisionStump	65.93	60.62	64.60	-1.33	3.98
ZeroR	54.20	54.20	54.20	0.00	0.00
OneR	54.20	59.07	54.87	0.66	-4.20
DecisionTable	71.68	68.14	66.37	-5.31	-1.77
ADtree	79.65	71.02	67.92	-11.73	-3.10
BaysianNet	77.21	72.12	70.35	-6.86	-1.77
Jrip	65.27	61.50	65.93	0.66	4.42
SimpleCart	77.88	68.36	69.03	-8.85	0.66
RandomForest	75.00	66.81	69.25	-5.75	2.43
Kstar	57.52	53.98	63.05	5.53	9.07
Logistic	63.27	67.48	69.47	6.19	1.99
SimpleLogistic	75.22	74.78	71.68	-3.54	-3.10
PART	76.77	72.12	68.81	-7.96	-3.32
Average	69.64	65.07	65.82	-3.82	0.75

Table 1: Accuracy results for various Classifiers on the Arrhythmia data set.

ing attribute. In all data sets, the class variable was omitted. All data sets were chosen for a large number of numeric features and high density in order to attempt to challenge the algorithm. In particular, the Arrhythmia data set is one of the larger data sets in the UCI repository, and due to the problem domain, many variables are related. Properties of the data sets are listed in Table 2.

The run time results for various levels of t are shown in Figure 4. For the SYLVA and MADELEN data sets the run time remains relatively constant. It is only when the threshold becomes very small that the search space expands significantly. In the Arrhythmia data set on the other hand, many more correlations are exhibited. Indeed, this is expected as the variables in the data set are related in the domain. A threshold of $t = 0.2$ took over 10 minutes, at which point the experiment was stopped.

The results also show that on these data sets, which are presumed to be at worst typical, there are relatively

Data set	Attributes	Instances
MADOLEN	500	2000
SYLVA	216	13086
Arrhythmia	279	452

Table 2: Data set properties.

few complete maximal sets when t is above about 0.4. This means that the enumeration approach considered is ideal, as it allows heavy pruning of the search space and therefore allows it to progress quickly.

5.2 Feature Selection Performance

The approach of Section 4 is used here to perform feature selection on the Arrhythmia data set. t was set to 0.5, resulting in 111 attributes being selected out of the 279 original attributes. If only positive correlations are considered, 135 attributes would have been selected.

In addition to comparing classification results on the reduced dataset to the original data set, a comparison to PCA was also performed. PCA was performed using the algorithm from WEKA [11], and options were set so the same number of attributes – 111 – were chosen. The 111 principle components cover 96% of the variance of the data set.

Table 1 shows the results on various classifiers in WEKA [11] (version 3.5.7), evaluated over the original data set, the data set with features extracted using PCA, and the subset of the attributes selected using the approach in Section 4. Unless otherwise stated, default values were used in the ML algorithms. The 16 classes in the original dataset were amalgamated into two classes, representing normal heart rhythms (245 instances) and cardiac arrhythmia (207 instances).

10-fold cross validation was used for the evaluation of classification accuracy in all cases. The approach in this paper performs comparably to PCA, having only a 0.75% better accuracy on average. On average, the accuracy is 3.82% lower than on the original dataset. Therefore, not only can this approach compete well against PCA, but it maintains the interpretability of the model. That is, the rules and decision trees built on the data set retain the actual attributes, in contrast to when PCA is used.

6 Related Work

A complete set and a clique are equivalent. The latter is often used in social network situations or in spatial data sets. In spatial applications, the space in which variables exist is usually low dimensional so enumeration approaches to mine them are not appropriate. Also, distances are used, rather than correlations (angles). Complex cliques have been considered [5], but this is in relation to *absence* of objects.

Graph based clustering approaches are also related. In some sense, the approach described in this paper is related to agglomerative clustering [8]. The desire for *complete* sub-graphs (sets) is the same as the *clique* pattern, or in distance based approaches, the MAX approach [8]. The *maximal* set idea could be considered as the highest level in a hierarchy defined over subsets, but the method does not fit into hierarchical clustering. In particular, the threshold is fixed. The approach in this paper is not really a clustering method. It is best described as a method of mining interactions between variables, with those interactions having a specific structure and being defined by correlation – rather than what would traditionally be called ‘distance’ measures. The consideration of *complex* interactions in particular sets it well apart from clustering approaches.

The algorithmic approach actually has a closer relationship to item-set mining than it does to clustering. Items are a special type of variable, and item-sets are sets of variables possessing some interesting property – usually that they occur frequently (frequent item-set mining). The similarity to *item enumeration* approaches is that the enumeration is over sets of variables, from the bottom up. The most related of such techniques is GLIMIT [9], which, unlike Apriori [2], is able to function in a depth first fashion like the algorithm presented here. The fundamental difference to item-set mining is that the item-set mining problem cannot be mapped to graph mining, as it cannot be reduced to pairwise relationships. Complex relationships therefore also don’t mean the same thing. While the *absence* of an item can be considered, this is completely different to the complex relationships considered in this paper.

It should also be emphasised that the use of correlation is a core component of this work, in particular, the lemma and corollaries that are developed under the completely connected sub-graph structure. Correlation is generally not used for clustering, and it cannot be used for item-set mining, as it does not translate to more than two variables at a time. Unlike distance measures, it has both positive and negative values – therefore techniques based on it necessarily have different semantics.

Feature subset selection comes in three flavours; wrapper, embedded or filter [8]. In the wrapper or embedded approaches, it is used in conjunction with a data mining or machine learning algorithm in some form of supervised or semi-supervised approach. The wrapper approach uses the DM or ML algorithm as an objective function, while in the embedded approach the DM or ML algorithm decides what features to discard as part of its operation. The filter approach selects a subset independently of the subsequent DM/ML algorithm. It may or may not be supervised. The approach described in Section 4 fits into the unsupervised filter category. A filter approach using correlation for feature subset selection is [4]. However, this is a hill climbing, supervised, optimizing approach that, in short, is completely different. It is also based on entropy – not statistical correlation.

Finding representative sets is considered in [7] using an entropy based approach on binary data. The algorithm in [7] mines a representative set directly (this work performs it as a second step), and overall it is also quite different.

As earlier mentioned, the idea of maximal complex sets *representing* underlying factors of the data set has similarities to the way principle components can be applied. But as also mentioned earlier, these are completely different approaches.

In summary, the work in this paper is related to various bodies of work in Data Mining, but to the author’s knowledge, is quite different to each. To the author’s knowledge, the approaches and theory presented in this paper are novel.

7 Conclusion

This paper has presented and exploited useful results about the possible correlation structures between variables. Additionally, it proposes the ‘complete, complex and maximal sub-graphs or sets of highly correlated variables’ pattern. This approach is useful as a Data Mining technique in its own right, or, as also demonstrated in this paper, as the core component of an unsupervised feature subset selection procedure.

Future work will improve or replace the feature subset selection technique.

References

- [1] D.J. Newman A. Asuncion. UCI machine learning repository, 2007.
- [2] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proceedings of 20th International Conference on Very Large Data Bases VLDB*, pages 487–499. Morgan Kaufmann, 1994.
- [3] Joseph Hair, Bill Black, Barry Babin, Rolph Anderson, and Ronald Tatham. *Multivariate Data Analysis (6th Edition)*. Pearson, November 2005.
- [4] M.A. Hall and L.A. Smith. Feature subset selection: a correlation based filter approach. In N. Kasabov and et al., editors, *Proc Fourth International Conference on Neural Information Processing and Intelligent Information Systems*, pages 855–858, Dunedin, New Zealand, 1997.
- [5] Rob Munro, Sanjay Chawla, and Pei Sun. Complex spatial relationships. In *Proceedings of the 3rd IEEE International Conference on Data Mining, ICDM 2003*, pages 227–234. IEEE Computer Society, 2003.
- [6] Nips 2003 workshop on feature extraction. <http://clopinet.com/isabelle/projects/nips2003>.
- [7] Feng Pan, Wei Wang, Anthony K. H. Tung, and Jiong Yang. Finding representative set from massive data. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 338–345, Washington, DC, USA, 2005. IEEE Computer Society.
- [8] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison Wesley, May 2005.
- [9] Florian Verhein and Sanjay Chawla. Geometrically inspired itemset mining. In *ICDM*, pages 655–666. IEEE Computer Society, 2006.
- [10] Performance prediction challenge, wcci model selection workshop, 2006. <http://www.modelselect.inf.ethz.ch/datasets.php>.
- [11] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques, 2nd Edition*. Morgan Kaufmann, 2005.