

Mining Surprising Patterns and Their Explanations in Clinical Data

Yen-Ting Kuo, Andrew Lonie, Adrian R. Pearce, Liz Sonenberg*
Department of Computing and Information Systems
The University of Melbourne, Australia

March 8, 2013

Short running head: Mining Surprising Patterns in Clinical Data

Abstract

We present a new technique for interactively mining patterns and generating explanations by harnessing the expertise of domain experts. Key to the approach is the distinction between what is unexpected from the perspective of the computational data mining process and what is surprising to the domain expert and interesting relative to their needs. We demonstrate the potential of the approach for discovering patterns and generating rich explanations in a clinical domain. Discovering interesting facts in clinical data is a grand challenge, since medical practitioners and clinicians generally have exceptional knowledge in the problem domain in which they work, however this knowledge is typically difficult to isolate computationally. To identify the desired surprising patterns, we formally record user knowledge, and use that knowledge to filter and constrain the output from an objective data mining technique, with the user making the final judgement as to whether or not a rule is surprising. Specifically, we introduce an *unexpectedness* algorithm based on association rule mining and Bayesian Networks and a *ϵ -explanations* technique for explanation generation to identify unexpected patterns. An implemented prototype is successfully demonstrated using a large clinical database recording incidence, prevalence and outcome of dialysis and kidney transplant patients.

*Corresponding author is Liz Sonenberg, l.sonenberg@unimelb.edu.au

1 Introduction

In this paper, we tackle knowledge mining by taking into account the potential to utilise an expert's own knowledge base for discovering new, *surprising*, knowledge. From the perspective of an end-user, patterns derived during the data mining process are not always interesting. Several promising techniques for mining unexpected patterns have recently been shown to have the potential to address this problem, such as the work of Padmanabhan and Tuzhilin (2006) on discovering minimal unexpected patterns and the unexpectedness mining algorithm of Jaroszewicz and Simovici (2004; 2009). However, such unexpected patterns are not necessarily surprising to the user. On the other hand, techniques have been developed for evaluating interesting patterns based on user expectations e.g. (Liu *et al.*, 1999; Ohsaki *et al.*, 2007).

This paper builds on our preliminary work (Kuo *et al.*, 2008), by creating a new approach and prototype that addresses the challenge of mining and identifying patterns that are interesting to a domain expert. We focus on knowledge that may inform clinical practice. Thus our challenge is not in mining patterns from the data based on criteria defined *a priori*, but in interactively discovering patterns that are surprising to the expert and, importantly, interesting and useful in clinical practice.

In medical practice, clinicians typically diagnose, prognosticate and treat patients based on their medical knowledge; for instance, nephrologists combine knowledge of the anatomy and function of kidneys, risk factors for kidney disease and knowledge on kidney disease progression in treating patients with renal failure. The well-known evidence-based medicine paradigm is based on the ideal that the essence of the aggregated data about past patients provides the empirical basis for decisions about current and future patients (Kuhn *et al.*, 2008). Clinical record databases generally contain diverse information about patient history, such as physiology, treatment and disease progression. These clinical databases frequently contain significant implicit knowledge about the domain and are excellent targets for data mining. However, much of the knowledge implicit in such a database is likely to be only known by clinicians expert in the domain.

An example of a clinical database is the Australia and New Zealand Dialysis and Transplant (ANZDATA) registry¹. Since 1980, the Australian and New Zealand governments have funded this registry to record the incidence, prevalence and outcome of dialysis and transplant treatment for patients with end stage renal failure. Although statistical analysis of data within this registry has yielded some useful insights into the trends in kidney disease

¹<http://www.anzdata.org.au/v1/index.html>

progression, it has required a significant investment in time in order to follow the process of hypothesis selection and validation (McDonald *et al.*, 2002; Lim *et al.*, 2005; Haysom *et al.*, 2009).

In this paper, we demonstrate that it is possible to automate the discovery of surprising facts without requiring a user-generated-hypothesis approach. We implement a two-step process of firstly capturing domain expert knowledge in an explicit formal model and leverage this model to derive unexpected patterns from the data, and secondly derive data-driven explanations to assist the expert in interpreting the new patterns. This work constitutes a promising new approach for discovering surprising patterns in clinical datasets, it is important to note that the associations we present are exploratory in nature.

Our approach relies on an unexpectedness algorithm based on filtering association rules mined by Bayesian representations of a users knowledge, which is achieved through modification of an approach first described by Jaroszewicz and Simovici (2004), but modified to accommodate mining surprising patterns and explanation.

The structure of this paper is as follows. Section 2 sets out background material: a description of the ANZDATA registry; a brief account of the terminology we use in regard to *surprising*, *unexpected*, and *interesting*, set in the context of related literature. In Section 3 we survey related work on unexpected pattern mining. In Section 4 we introduce a mechanism for explanation generation that supports our approach to mining unexpected patterns. Section 5 describes our proposal for an iterative process of mining unexpected rules, combining association rules with a updateable probabilistic knowledge representation based on Bayesian Networks (BNs). Our experiments with this approach using the ANZDATA registry and with involvement of a domain expert are presented in Section 6. The lessons learned from the experiments point to types of gaps in mining surprising patterns, as catalogued in Section 6.3. We conclude with a discussion and some observations on limitations of the evidence gathered so far for the approach (Sections 7 and 8).

2 Background

The ANZDATA registry contains a wealth of information on kidney disease treatment and progression in Australian and New Zealand renal patients. As at 2010, the database recorded some 96 attributes on each patient, including demographic, clinical and physiological data, on a yearly basis; 19,220 patients were represented for a total of 217,803 records. Each patient has multiple entries, corresponding to the outcome of each dialysis and transplant

treatment. There are 96 variables that are categorized into 7 types: record tags, clinical attributes, clinical findings, procedure, disease/syndrome, behavior, and health care activity. The attributes in the dataset are either categorical or numerical, and some of the categorical attributes encode rich terminologies, and potentially knowledge, in their values: for example, the allowable values of the `TreatmentModality` attribute are: [1=Haemodialysis; 2=Peritoneal dialysis; 3=Transplant(Graft); 4=Return of native function; 5=Loss to follow up]. Additionally, as each patient is likely to have multiple entries, further information is implicitly encoded in the progression of time variant attributes. The dataset also contains a large percentage of missing values.

A general goal in data mining is finding interesting facts (Silberschatz and Tuzhilin, 1996). In this paper, we use *interesting* as a subjective user-perspective qualifier, and, to be consistent with the data-mining literature, *interestingness* as a computational metric of data mining output, either independent of, or partially dependent on, a computational representation of the user's domain knowledge. It is important to note that although interestingness metrics are generally developed as computable approximations of interesting, the two terms are not equivalent.

In preliminary experiments on deriving knowledge from the ANZDATA dataset, we applied conventional association rule mining (Agrawal *et al.*, 1993) and emerging pattern mining techniques (Ramamohanarao and Bailey, 2003) to a subset of the data (in which a targeted subset of attributes was selected for a reduced set of records). Unsurprisingly, the first problem we encountered was that an overwhelmingly large set of rules was generated, and furthermore many of the rules were either already known, or seemed *reasonable* (that is, simple to explain in the context of the user's knowledge) to a domain expert (in this case, a senior nephrologist). In other words, uninformed data mining was generally unable to provide interesting or surprising patterns to the expert, perhaps because these patterns were obscured by the sheer number of rules.

A straightforward observation is that *interesting* is a relative term: clearly, any newly presented fact will be assessed as interesting or not with respect to the knowledge that a user already has, and a domain expert will have much more specific domain knowledge than a non-expert. So, before we can identify potentially interesting facts, we need to know what the user knows. Here we use the term *domain knowledge (DK)* as the set of internal knowledge that an individual user interacting with a data mining system has; *DK* refers to terms, concepts, relationships and facts relevant to the dataset being mined; it will vary from user to user, approaching zero in naïve users. To enable automation of the process of discovering interesting facts, we need to

record some part of the *DK* of the user in a computable form, which we refer to as the system's *knowledgebase (KB)*.

In previous work we demonstrated that it is possible to reduce the number of association rules presented to the user by applying a filtering step in which a model of domain knowledge - in that case a domain ontology derived from the UMLS (Unified Medical Language System) - was used to define semantic templates for attributes under which rules were aggregated (Kuo *et al.*, 2007). Although the technique effectively reduces the possible set of rules, it does not necessarily make the set more interesting to the user; in fact, the rules selected for presentation were effectively biased towards being unsurprising, as they had to conform to a template defined by the most common domain knowledge - that recorded in a domain ontology. That technique, while reducing the total rule count and minimising duplication caused by attribute synonyms, is not effective for discovering interesting rules.

There are two types of interestingness: *objective* and *subjective*. Objective interestingness metrics are independent of a user's knowledge, and are defined using statistical significance measures such as conciseness, coverage, reliability, precision, accuracy - that is, they depend solely on the dataset being mined (Geng and Hamilton, 2006; McGarry, 2005). Subjective interestingness metrics are dependent on a user's knowledge; one common type of subjective interestingness is *unexpectedness*.

In data mining, 'unexpectedness' generally refers to a computational property of knowledge with respect to an identified knowledgebase. That is, unexpectedness is a mathematical metric applied to mined rules or patterns, and is a derivative of both a specific knowledgebase *KB* and a particular representation of the related domain knowledge *DK*. Importantly, the value of unexpectedness will change if *KB* changes.

In some earlier work, the term unexpected was used in two ways: (a) a pattern is computationally unexpected to the data mining system or (b) a pattern is unexpected to the user. To avoid this ambiguity, we adopt nomenclature that distinguishes the human ('user') and computational ('system') perspectives (Fig. 1). For the subjective user-perspective term to correspond to the computational property of unexpected, we adopt the term *surprising*. So, relative to a system's knowledge base, a pattern may be 'unexpected'; relative to a user's domain knowledge, a pattern may be 'surprising'.

[Figure 1 about here.]

From the perspective of a user, if we assume that one aspect of being interesting is being surprising to the user - that is, new facts/patterns are likely to be interesting - then it is reasonable to suggest that unexpectedness

metrics should be better approximations of interesting than objective interestingness metrics, purely because they can take some approximation of the user’s knowledge into account in the ranking process.

In our usage, rules/facts are surprising to a user, and unexpected to a data mining algorithm. So, another assumption of the research described here is that, as the computational representation of the user’s knowledge KB used by an unexpectedness algorithm converges to the user’s actual DK , so the set of unexpected facts identified by the metric are more likely to be surprising to the user, and therefore more likely to contain interesting facts. Importantly, we use the terms *interesting* and *surprising* in reference purely to the users’ perspective. This definition differs from earlier work, reviewed in Section 3, in which interesting and surprising have been used to describe both the users’ impressions and the computational properties of rules.

Below, to identify surprising facts, we develop a technique for formally recording user knowledge, and using that knowledge to filter and constrain the output from objective data mining methods, with the user making the final judgement on whether a rule is surprising. Specifically, we use an unexpectedness algorithm based on filtering association rules mined by BN representations of a users knowledge. A similar approach has been described by Jaroszewicz and Simovici (2004; 2009) and we extend that technique to mine surprising patterns and to generate explanations.

3 Related work on unexpectedness in pattern mining

A well recognized problem in data mining is the generation of enormous numbers of patterns which can overwhelm a user (Liu *et al.*, 1999). A general solution to this problem is to measure the interestingness of patterns so that the results can be pruned or ranked. Ohsaki *et al.* (2007) studied the utility of 40 objective interestingness metrics on meningitis and hepatitis data, and found some metrics useful in estimating the interest of medical experts. In terms of subjective interestingness, which incorporates the knowledge of a user, the most studied metric is unexpectedness. The general goal of unexpectedness based pattern mining is discovering surprising patterns to users.

One of the difficulties in mining unexpected patterns comes from the innate character of the entangled interaction between the user and the data mining algorithm during the mining process, as depicted in Fig. 2: initially, DK is acquired and represented in some computable model KB ; secondly, the mining algorithm calculates the degree of unexpectedness for the mined

patterns against the *KB* and presents results to a user; third, the user learns some knowledge from the unexpected patterns; and finally, the user refines the *KB* and performs the next iteration, starting from the second step.

[Figure 2 about here.]

Ideally, the *KB* (that is, the formal representation of the users’s knowledge), converges on the user’s knowledge, and the suggested patterns become more and more relevant. The dynamic character of mining unexpected patterns introduces many issues that are different from those of common data mining tasks, including the complexities of acquiring and representing the *KB*, aligning the *KB* representation with *DK*, and aligning the *KB* representation with the data model in the dataset being mined. Abstractly, the unexpectedness of a pattern is the difference between *KB* and the pattern:

$$Degree(Unexpectedness) = Difference(KB, pattern). \quad (1)$$

The *Difference()* function will be defined according to the type of knowledge representation used. For example, rule based representations may define *Difference()* as a scoring function which estimates the difference between a rule in the *KB* and the input pattern; probabilistic representations may define *Difference()* as the degree of belief change (Jaroszewicz *et al.*, 2009; Malhas and Aghbari, 2009).

Various different knowledge representations have been trialled to capture users’ expectations: rules, fuzzy rules, fuzzy taxonomy, probabilities and BNs. These representations can be categorised into two groups: rule based and probability based representations.

Rule based representations utilise terminologies of subjects and relations for defining users’ expectations about data. Approaches include: *General Impressions*: rules involving impression descriptors: $ID \in \{<, >, <<, |, [aset]\}$, which describe the impression over attributes (Liu *et al.*, 1997); *Fuzzy Linguistic Variables* and *Knowledge Rules*: in which users define semantic meanings over possible values of each attribute (Liu *et al.*, 1999; Wang *et al.*, 2003); *Expectations*: association rules expressed in logical literals (Padmanabhan and Tuzhilin, 1998, 2008, 2002); and *Fuzzy Taxonomy*: tree-based representation with items represented as leaf nodes and concepts represented as non-leaf nodes, where **is-a** relationships between parent nodes and child nodes can be used to derive rules (Shekar and Natarajan, 2004).

There are three main approaches for calculating unexpectedness for rule based representations: *Rule matching*: based on the concept that a rule and a belief are different if either the consequents of the rule and the belief are ‘similar’ but the antecedents are ‘far apart’, or vice versa. Unexpectedness is

generally calculated as the degree of difference (Padmanabhan and Tuzhilin, 1998); *Logical contradiction*: defined in a mined rule $A \rightarrow B$ and a belief $X \rightarrow Y$ satisfying the following conditions: (a) B and Y logically contradict each other, (b) $A \wedge X$ have high *support* (the joint probability in association rules) and (c) the rule $A, X \rightarrow B$ holds (Padmanabhan and Tuzhilin, 1998, 2008, 2002, 2006); and *Item-relatedness*: defined by *highest-level node membership* (HM), *highest-level relatedness* (HR), and *node separation relatedness* (NSR) from the structural context of the fuzzy taxonomy (Shekar and Natarajan, 2004).

Probabilistic representations of knowledge have been used to define formal models of user’s beliefs. Silberschatz and Tuzhilin (1995; 1996) have described a probabilistic representation where KB is defined in probabilistic beliefs; interestingness (unexpectedness) is then defined as: $I(p, B, \xi) = \sum_{\alpha_i \in B} w_i |d(\alpha_i|p, \xi) - d(\alpha_i|\xi)|$. The interestingness of a pattern p relative to a belief system B and prior evidence ξ is defined in terms of how much degrees of beliefs (α_i) change as a result of a new pattern p weighted by weight factors w_i . Various authors have used BNs for representing background knowledge, e.g. (Jaroszewicz *et al.*, 2009; Malhas and Aghbari, 2009). Each variable is represented as a node in the network and the edges are assigned manually so the user can express their belief of the dependencies among variables. Unexpectedness is calculated as $\mathcal{S}(I, i) = |P_I^D(i) - P_I^{BN}(i)|$ for the itemset I with values i . $P_I^D(i)$ and $P_I^{BN}(i)$ represent the probabilities in the data and BN respectively.

In this work we use a probabilistic knowledge representation (BNs) for two important reasons: because they can be graphically visualised and therefore facilitate interaction with the user; and because they provide a mechanism for inferencing probabilities for unexpectedness. It should be noted, however, that one major advantage to logic-based algorithms is that they do not require eliciting probabilities associated with beliefs from users, as eliciting probabilities is a difficult task in acquisition of user knowledge.

4 Explanation generation

In our experience, when presenting unexpected rules to a user, that user will try to understand the rules by explaining them based on their own knowledge; yet, this reasoning task should be difficult because these rules should be surprising to the user. Again, in our experience, frequently rules would be regarded as surprising at first glance, then become unsurprising in retrospect if a user comes up with a rational explanation involving related factors, al-

though the user would still wonder whether his/her explanation is supported by the data. On some occasions the user could not come up with any explanation for a rule, but wondered whether there is some possible explanation(s) in the data. Finding explanations seems to play a critical role in pattern interpretation; potentially, then, computed 'explanations' relevant to a particular rule and supported by the data might provide a way for an expert to interpret that rule, in addition to their own knowledge.

Our approach builds on our earlier work for explanation generation based on the context of probabilistic dependencies and modeled in BNs, and included this approach in the overall data mining process for finding unexpected patterns (Kuo *et al.*, 2008). As an example of this type of explanation generation, one particular mined association rule from ANZDATA stated that when a kidney failure patient is not diagnosed with cancer, they are more likely to die of cardiovascular disease ($\neg\text{Cancer} \rightarrow \text{CardiovascularDeath}$). The domain expert interpreted this rule by giving the following explanation: when patients are diagnosed with cancer, they usually die of cancer; when patients are not diagnosed with cancer they are unlikely to die of cancer and therefore are more likely to die of cardiovascular disease (this is the mortality pattern of kidney disease patients). A matching 'explanation' could be generated from a BN created from the ANZDATA dataset: $\neg\text{Cancer} \rightarrow \neg\text{DieOfCancer} \rightarrow \text{CardiovascularDeath}$; providing this (and other) explanations to the expert may well have aided the pattern interpretation process.

In terms of mining surprising rules, we propose that explanation generation helps users assess whether a rule is *implicitly reasonable* or not. Prior to empirical evidence, we surmise that if a user cannot explain a rule, and the generated explanations do not make sense (at least initially), the rule should be truly surprising to the user, and potentially more likely to be interesting.

Because unexpected rules should be surprising to a user, the user must make some judgement on how the new fact fits with their current understanding and knowledge. It is possible (and in an ideal data mining world, desirable) that a user may find an unexpected fact contradictory to their own knowledge. If a user finds a rule is initially surprising, but the explanation of that rule matches his/her knowledge, then we say that the explanation is reasonable, and the rule would then become unsurprising. In the case that the rule is contradictory to the *KB*, and the user finds the set of presented explanations unreasonable, then the rule would warrant further investigation, and would be potentially interesting. Providing users with a set of potential explanations may allow users to make more informed judgements on unexpected facts in terms of whether they are surprising or not: a fact that initially seems surprising may move to unsurprising status if a reasonable explanation is found (Section 6.2).

We briefly summarise our method for finding explanations introduced in (Kuo *et al.*, 2008), which is based on probabilistic inference techniques. We seek to mimic the way an expert might explain a rule by forming a reasoning chain. In our discussions with domain experts, we found that the explanations provided by the expert can often be effectively captured in terms of conditional probabilities; therefore we choose to define explanations based on conditional probability.

Our approach to explaining association rules, termed ε -Explanations, follows in the spirit of techniques for BNs that utilise the chain rule for factorizing variables. The approach stems, at least in part, from the original idea of factoring variables as a sum-product of local functions as a basis for effective algorithms (Kschischang *et al.*, 2001).

Definition 1. (ε -Explanations) Given $\varepsilon > 0$ and an association rule $A \rightarrow B$ with *confidence* = $P(B|A)$, we say a variable X is a factor in an explanation if X can approximate the *confidence* of the rule via the chain rule in probability:

$$|P(B|A) - \hat{P}(B|A)| < \varepsilon, \text{ where } \hat{P}(B|A) = \sum_X P(B|X)P(X|A).$$

The explanation is written $A \rightarrow X \rightarrow B$.

Definition 1 deals with only three variables: A , B and X . However, association rules sometimes have more than one variable on the left or right hand side. Furthermore, there is no reason to limit explanations to only one factor at a time. Therefore, we introduce the use of BNs for modeling explanations; a BN is capable of modeling the dependencies amongst variables, and it is not limited to three variables.

Let $\mathcal{A} = \{a_1, a_2, \dots, a_m\}$ be a set of variables (attributes) at the left-hand-side of a rule and B be a variable (attribute) at the right-hand-side of the rule (In this paper, we consider only one variable on the right-hand-side.); $\mathcal{X} = \{x_1, x_2, \dots, x_r\}$ be a set of variables (attributes) serving as the factors in an explanation. The definition of a Bayesian Network as an explanation is given below.

Definition 2. (BNs as ε -explanations) Given an association rule $\mathcal{A} \rightarrow B$ with *confidence* = $P(B|\mathcal{A})$, a BN G with a set of variables \mathcal{X} is a ε -explanation if it satisfies the following requirements:

1. $\mathcal{X} \cap (\mathcal{A} \cup B) = \emptyset$. Every x_i in \mathcal{X} is not in \mathcal{A} or B .
2. There is no direct connection between nodes in \mathcal{A} and node B .
3. Every node a_i in \mathcal{A} has connections with one or more nodes in \mathcal{X} . There are no arcs among the nodes of \mathcal{A} .
4. There are no isolated nodes in \mathcal{X} . Each node x_i has degree at least two.
5. Let $\hat{P}_G(B|\mathcal{A})$ be the inferred conditional probability from the BN G . $\hat{P}_G(B|\mathcal{A})$ is within ε of the conditional relative frequency in the data:

$$|P(B|\mathcal{A}) - \hat{P}_G(B|\mathcal{A})| < \varepsilon \quad (2)$$

Therefore, the problem of finding explanations is reduced to finding a set of intermediate variables $\mathcal{X} = \{x_1, x_2, \dots, x_r\}$ and a BN G that can approximate the conditional relative frequency $P(B|\mathcal{A})$ when $a_1, a_2, \dots, a_m \in \mathcal{A}$ is known. Several examples of BNs as possible explanations are illustrated in Fig. 3; the orientation of arcs can be assigned freely as long as the network G remains acyclic.

[Figure 3 about here.]

5 Mining unexpected rules

In this section we describe the development of an interactive process of *unexpected* rules mining combining association rules with a updateable probabilistic knowledge representation based on BNs. We have developed an interactive process so that the user can iteratively add knowledge to the Bayesian Network representation of KB , allowing for a gradual building of the background KB . The data mining process is depicted in Fig. 4 and a screen capture of an application implemented in MATLABTM that presents the domain expert with ranked unexpected rules, their explanations, a KB and an interface for modifying KB is shown in Fig. 5.

[Figure 4 about here.]

[Figure 5 about here.]

Initially, dependent association rules are mined via a χ^2 test on association rules from the dataset (steps *a, b*), while (independently) a *KB* is created by the user in the format of an association graph (*AG*) (step *c*). Local BNs are then generated for every dependent association rule as described in Algorithm 2 (step *d*). Following this, unexpected rules are mined according to the algorithm described (steps *e, f*), to generate a set of possible explanations (step *g*) for assisting pattern interpretation (section 4) (step *h*).

In the next iteration, a user may update the *KB* by refining the *AG* based on the mined unexpected rules and their explanations (step *i*); then repeat the procedure to generate another set of unexpected rules. In terms of termination criterion of the iterative procedure, a user stops this process when the maximal unexpectedness (Eq. 5) is lower than a particular threshold.

Association rules, as the name suggests, represent associations rather than correlations. Correlation analysis is typically used to determine dependence between two or more variables of a population from experimental data, where as in association rules *A* and *B* are associated if $P(A, B) > k1$ and $P(B|A) > k2$, for thresholds *k1* and *k2*, however *A* and *B* might be independent from a statistical perspective, i.e. $P(B|A) = P(B)$. Therefore the need for explicit *dependence* is the key difference.

Since our knowledge is represented as a BN, it is important to present truly correlated rules to a user so that the user can update the network accordingly. An intuitive idea to resolve this problem is to apply a classical test for independence to the association. The use of independence tests for associations has been explored previously in data mining (Bay and Pazzani, 1999; Bayardo, Jr. and Agrawal, 1999; Brin *et al.*, 1997; Silverstein *et al.*, 1998). Of particular interest here, Silverstein *et al.* (1998) defined *dependence rules*, which are a kind of generalisation of association rules that identify statistical dependence in itemsets. The significance of dependence of rules is measured via a chi-squared test (χ^2 test) for independence. To ensure the presented association rules do not represent independent relationships, we applied the χ^2 test to filter mined association rules; effectively, presenting dependent association rules to the user. Dependent association rules are slightly different from dependence rules in that dependent association rules exhibit both high confidence and statistical dependence.

5.1 Measuring correlation in association rules

The purpose of testing rules' independence is to identify rules with attributes that are truly dependent. As proposed by Silverstein *et al.* (1998), the chi-squared (χ^2) test can be applied to test independence of mined rules. Given an association rule $A \rightarrow B$, we can build a 2-way contingency table of *A, B*

as Table 1 shows.

[Table 1 about here.]

The expectation of each cell in Table 1 is calculated as

$$E_{i,j} = \sum_{i'} r_{i',j} \frac{\sum_{j'} r_{i,j'}}{N}. \quad (3)$$

Then the χ^2 value for attributes A, B is calculated by

$$\chi^2 = \sum_{i,j} \frac{(r_{i,j} - E_{i,j})^2}{E_{i,j}}. \quad (4)$$

We can set the null hypothesis as: A and B are independent. When χ^2 is near zero, the null hypothesis is not rejected. On the other hand, if $\chi^2 > \chi_{\alpha;df}^2$, the null hypothesis will be rejected. $\chi_{\alpha;df}^2$ is the quantile of χ^2 distribution at α *significance level* and df *degree of freedom*. The α is usually set at 0.05 or 0.01 and the df is determined by the numbers of rows and columns in a table: $df = (Rows - 1)(Columns - 1)$. For association rules with 2 left hand attributes, e.g. $A, B \rightarrow C$, the χ^2 test is performed by testing B, C under all possible values of $A = \{a_1, a_2, \dots, a_i\}$.

A number of representations are available to formally represent knowledge; as discussed earlier, we chose BNs because they can be graphically visualised and have straightforward mechanisms for inferring probabilities for unexpectedness.

A BN consists of a set of nodes and a set of arcs (or links) which forms a directed acyclic graph (*DAG*). Each node denotes one variable and an arc between two nodes represents the direct dependency between two variables. In the case of discrete variables, a child node (a child node is at the end point of an arc) of an arc is associated with a conditional probability table (*CPT*) representing its distribution under all values of the parent nodes. A *skeleton* of a BN is a graph of the network's structure where the arcs are not directed. To represent knowledge in BNs, two kinds of knowledge need to be acquired: the structure of the network and the parameters of the network. However, it is impractical to require a user to manually assign BN parameters. To minimise the burden on the user, the only knowledge we ask a user to explicitly provide is knowledge about associations between variables in the dataset to be mined. For each variable in data, the user is asked to assign other variables that he/she thinks **relates-to** the given variable. For example, an expert provides the following associations between variables for the

ANZDATA dataset: smoking (age,race,sex,...), diabetes (age,race,weight,...) hypertension (age,bmi,diabetes,...). Additionally, since association rules deal only with discrete variables, the BNs discussed in this paper consist only of nodes corresponding to variables with discrete values.

The acquired table of associations is equivalent to an association graph, *AG*. Because the *AG* is manually assigned, it inevitably contains cycles that are not allowed in BNs. Intuitively, for a cyclic path in a graph, flipping any edge would break the cycle. Now we consider a whole graph with many cycles. If we randomly flip an edge of a cycle, the graph would remain cyclic because there are still other cycles and the flipped edge might incur new cycles; however, if we iteratively apply this strategy to the graph until no more cycles, then the graph becomes acyclic. We can flip edges because our basic descriptor, **relates-to**, is symmetric, so the converted directed acyclic graph, *DAG*, represents the same knowledge as the original *AG*. Note that this argument overlooks the causality or other advanced issues, e.g. d-separation, in research on BNs for simplifying the design of knowledge acquisition, e.g. (Boneh *et al.*, 2006). A simple algorithm for converting an *AG* to a *DAG* is presented below (Algorithm 1).

```

Input: AG, Association Graph
Result: DAG, Directed Acyclic Graph
DAG  $\leftarrow$  AG;
stop  $\leftarrow$  False;
while stop = False do
    | path = FindCyclicPath(DAG);
    | if path= $\emptyset$  then
    | | stop  $\leftarrow$  True
    | else
    | | edge = RandomPickEdge(path);
    | | DAG = FlipEdge(DAG, edge);
    | end
end
return DAG

```

Algorithm 1: *AG*-to-*DAG* conversion algorithm.

The algorithm iteratively and randomly flips an edge of every cycle until done. There is a FindCyclicPath(*G*) function that returns the first cyclic path found in graph *G*. The RandomPickEdge(*path*) function returns an edge that is randomly picked from the edges in *path*. The flipping function FlipEdge(*G*, *e*) changes the orientation of *e* in *G*. The graphs, *AG* and *DAG*, are in a matrix format.

The exponentially growing CPTs of BNs impose a practical memory-based restriction on the number of parent nodes of a child node. We found it impractical to build a full BN representation of the acquired DAG, which was based on all of the ANZDATA variables, as the degree of vertices would be over 30. We decided to use part of the full *KB* graph for each mined association rule - that is, build a local BN for each rule.

A local BN of a rule $A \rightarrow B$ is extracted from the full BN in the *KB* by removing all nodes that are not in the pathways between A and B. More precisely, the local BN consists of (1) node A, (2) node B, (3) all the nodes in all the paths from A to B (denoted as set X), and (4) all the arcs amongst A, B and X. For an association rule $A \rightarrow B$, what we need to know from the *KB* is the expectation of $P(B|A)$, based on the definition of unexpectedness in Eq. 5. Inferring $P(B|A)$ from a BN involves three possible pathways: (1) paths from A to B and vice versa, (2) common causes of A, B and (3) their common effects. A local BN loses no information regarding $A \rightarrow B$ and will return the same inferred conditional probability for $P(B|A)$ as the full BN. Additionally, because the only evidence is the left-hand-side of the association rule, i.e. A, any path between A and B provides sufficient information for inference. Consider the following example:

[Figure 6 about here.]

Either $A \rightarrow X \rightarrow B$, $A \rightarrow Y1 \rightarrow Y2 \rightarrow Y3 \rightarrow B$ or $A \rightarrow Z1 \rightarrow \dots \rightarrow B$ provides a pathway for inferring $P(B|A)$.

Definition 3. (local BN) A local BN loses no information regarding $A \rightarrow B$ and will return the same inferred conditional probability for $P(B|A)$ as the full BN when conditioning over (1) node A, (2) node B, (3) all the nodes in all the paths from A to B (denoted as set X), and (4) all the arcs amongst A, B and X.

Below we present a simple algorithm for generating a local BN based on a given association rule. Let *AR* be the association rule having one left hand side (LHS) attribute or two LHS attributes: $A \rightarrow B$ or $A_1, A_2 \rightarrow B$ and *DAG* be the structure of the BN converted by Algorithm 1. The *FindAllPaths*(G_{ud}, A_i, B) function searches all possible paths between A and B and returns found paths. The parameters of the local BN are estimated by the *LearnParameters*(\cdot, \cdot) function which implements maximum-likelihood-estimation (MLE).

```

Input:  $D$ : data,  $AR$ : association rule;  $DAG$ : directed acyclic graph;
Result:  $BN_{local}$ : local BN
 $G_{ud} \leftarrow OR(DAG, DAG^T)$  ( $G_{ud}$  is a unidirectional graph);
 $\mathbb{A} \leftarrow AR$ 's LHS;
 $B \leftarrow AR$ 's RHS;
 $nodes \leftarrow \emptyset$ ;
forall the  $A_i \in \mathbb{A}$  do
    |  $paths = FindAllPaths(G_{ud}, A_i, B)$ ;
    |  $v = \bigcup_{vertices \in paths} vertices$ ;
    |  $nodes = nodes \cup v$ ;
end
 $nodes = nodes \cup \{A, B\}$ ;
 $BN_{local} \leftarrow DAG(nodes)$ ;
LearnParameters( $BN_{local}, D$ );
return  $BN_{local}$ 

```

Algorithm 2: Local-BN generating algorithm. The first step converts the input DAG to a unidirectional graph by applying a Boolean OR operation to every cell of the DAG and DAG^T .

Here we define a measure of unexpectedness in the context of conditional probability, inspired by the work of Jaroszewicz and Simovici. Utilising a BN as KB , we can infer the conditional probability (*confidence*) from knowing $A = a$ to $P(B = b|A = a)$. The inferred conditional probability, $\hat{P}(B = b|A = a)$, serves as the computational *belief* for the association rule $A = a \rightarrow B = b$. Unexpectedness is then defined as the difference between the confidence of a rule and its inferred conditional probability $\hat{P}(B = b|A = a)$.

Definition 4. (Unexpectedness of an association rule) Given a Bayesian Network BN and data D , the unexpectedness of an association rule $A = a \rightarrow B = b$ is defined as:

$$U(A = a \rightarrow B = b) = \left| P_D(B = b|A = a) - \hat{P}_{BN}(B = b|A = a) \right|, \quad (5)$$

where P_D is the relative frequency in data D , and \hat{P}_{BN} is the conditional probability inferred from BN .

Algorithm 3 demonstrates the process for mining unexpected association (and dependence) rules. Each rule is iteratively evaluated by calculating the difference between the actual probability of the rule (from the data) and the inferred probability (from the local BN); if the difference is greater than a threshold, the rule is appended to the set of unexpected rules.

<p>Input: D:data, BN_{local}: local BN, AR: association rules, ϵ_U:threshold for unexpectedness</p> <p>Result: R_U:Unexpected rules</p> <p>$R_U \leftarrow \emptyset$;</p> <p>forall the $r = \{A_1 = a_1, A_2 = a_2 \rightarrow B = b\} \in AR$ do</p> <table border="0"> <tr> <td style="border-left: 1px solid black; padding-left: 10px;"> $U(r) = \left P_D(B = b A_i = a_i) - \hat{P}_{BN_{local}}(B = b A_i = a_i) \right$;</td> </tr> <tr> <td style="border-left: 1px solid black; padding-left: 10px;"> if $U(r) > \epsilon_U$ then</td> </tr> <tr> <td style="border-left: 1px solid black; padding-left: 20px;">Append r to R_U;</td> </tr> <tr> <td style="border-left: 1px solid black; padding-left: 10px;">end</td> </tr> </table> <p>end</p> <p>return R_U;</p>	$U(r) = \left P_D(B = b A_i = a_i) - \hat{P}_{BN_{local}}(B = b A_i = a_i) \right $;	if $U(r) > \epsilon_U$ then	Append r to R_U ;	end
$U(r) = \left P_D(B = b A_i = a_i) - \hat{P}_{BN_{local}}(B = b A_i = a_i) \right $;				
if $U(r) > \epsilon_U$ then				
Append r to R_U ;				
end				

Algorithm 3: Unexpected-AR mining algorithm.

6 Experimental results

We preprocess the ANZDATA dataset in three aspects: attribute selection, record merging and discretization. The term ‘preprocess’ we use here actually consists of data *selection*, *preprocessing* and *transformation* under the framework of Fayyad *et al.* (1996). In terms of attribute selection, meaningless attributes such as the sequence number of each record were removed. This is informed by advice from the expert user. There were many paired attributes with the same meaning but different encoding; in such cases, we chose only one of them for the processed dataset. In terms of entry merging, we merged multiple entries from individual patients into single rows because we do not address temporal relations in this research. The entry merging procedure, however, is a compromise in the data mining setting because most of the expert’s knowledge has some temporal aspect.

[Table 2 about here.]

Finally, the discretization process equally divides numerical attributes into 4 bins by their values. The equally problematic discretization procedure overlooks the potential of discretization based on medical knowledge, though it is available from the domain expert, because the data preprocessing step is out of the scope of our overall design of unexpected rules mining. In this experiment, we use naïve solutions for data preprocessing because our design focuses on the steps of post-processing and pattern interpretation, but we note that more sophistication in the earlier stages is key to overall success in data mining (Pyle, 1999; Zhang *et al.*, 2003). Preprocessing the dataset reduced the number of variables to 39 (down from 96); some of these are shown in Table 2. As discussed in Section 5, the domain expert inputs

knowledge by assigning an association table; we then processed the acquired knowledge into a DAG.

6.1 Interactive mining of unexpected rules

Following the process described in Fig. 4, initially we mined association rules with minimal *support* = 0.6 and *confidence* = 0.6; under these constraints 1279 rules were identified. In the second step we verified the independence of every association rule via a chi-squared test. Of the 1279 rules, 361 passed the χ^2 test of being dependent under significance level $\alpha = 0.05$. The *KB*, assigned by the medical expert, has 375 links among the 39 variables; it is 49.3% connected in comparison to a complete graph. The dependent association rules are then ranked by their unexpectedness values. The rules with the highest unexpectedness score (U) are listed in Table 3.

[Table 3 about here.]

6.2 Explanation generation and expert interpretations

Explanation generation is designed to assist pattern interpretation; therefore we assess its effectiveness through the expert's interpretations of the unexpected rules. Table 4 lists the expert's feedback on surprisingness and reasonableness of rules and their explanations. Hence our approach successfully discovers two surprising rules – rules 5 and 6 from Table 3.

[Table 4 about here.]

Rule 5 is `lungb=No → txwlcacat = NotonTransplantList`, i.e. “patients without lung disease are not on the kidney transplant list”. The local BN and reasonable explanation of rule 5 are depicted in Figs. 7(a) and 7(b). Our expert commented on rule 5 in the following way: *“Rule 5 is not expected, as we would expect these patients to be on the list if there are no other problems. The generated explanation to Rule 5 relates late referral to no lung disease. A possible explanation is that the late referrals have more complications and higher risk of infection which may delay them being put on the transplant list, or may even be a permanent factor why they are not on the list.”*

Rule 6 is `hfdeath=AliveorNon-HFdeath, mideath=AliveorNon-MIdeath → txwlcacat=NotonTransplantList` i.e. “patients who are alive or don't die from heart failure and myocardial infarction are not on the transplant list”. The local BN and reasonable explanation of rule 6 are depicted in Fig. 8(a) and Fig. 8(b). Our expert commented on rule 6: *“Rule 6 is also not expected, as we would expect these patients to be on the transplant list because*

they should be in good health. The generated explanation suggests late referral is being associated with these group of patients. It is possible that later referral and its associated risks may be a reason why patients are not on or are delayed being put on the transplant list. Interestingly, the explanation rediscovers *agequart* as an intermediate factor the same as the local BN in Fig. 8(a).”

[Figure 7 about here.]

[Figure 8 about here.]

6.3 Identified gaps

Whilst the result is encouraging, we have identified specific ‘knowledge gaps’ that significantly impair our ability to mine meaningful and unexpected rules: they fall into two categories, which we term *literal* gaps and *reasoning* gaps.

Literal gaps occur when the data is encoded differently to the user’s expectation. As an example, rule 1 consists of attribute **dryweight** in the range between 53.2 – 101.5 kg; however, the expert is used to reasoning about a patient’s weight as overweight or underweight, rather than in terms of an absolute value. Examples and categories of literal gaps include:

Improper discretization Continuous data is not discretized into a meaningful range. Example: [Rules 1-3] The expert assesses **dryweight** = 53.2-101.5kg as not meaningful because adults usually have weights in that range.

Temporal information loss Information loss when summarizing multiple records from different time periods. Examples: [Rules 1-3] the expert considers the change in **dryweight** is more meaningful than the value (that is, gaining or losing weight is the important metric rather than absolute weight); [Rules 4-6] the values of **txwlc**(transplant waiting list) do not provide sufficient information because the expert also needs to know the change of status of a transplant list.

Exhaustive and mutual exclusive property of variables If a variable is exhaustive, all possible values are enumerated; if values of variables are mutually exclusive, they have no overlapping meanings. Example: [Rule 2] value **mideath=AliveorNon-MIdeath** is not exclusive because the value has overlapping meaning (alive or non-myocardial infarction death).

Semantic gap Encoding of data is not semantically consistent with the user’s knowledge. Example: [Rules 1-3] the expert considers **dryweight** = 53.2 - 101.5kg does not match her knowledge because she reasons a patient’s weight as overweight or underweight.

Reasoning gaps are caused by the difference between a user’s reasoning and the nature of association rules. Examples and categories of reasoning gaps include:

Partial alignment of mined rules and experts’ knowledge Example: [Rules 1-3] The **dryweight** attribute is partially aligned to an expert’s knowledge as the expert’s perspective of ‘weight’ is relative to sex and age, neither of which are shown in the rule.

Cause-effect context A consequence of the tendency of users to interpret rules in the context of cause and effects, when association rules do not represent causality. Example: [Rule 4] the attribute ‘cause of death’ is on the left hand side of the rule, and the expert determines that this rule is not correct as they know that the status of transplant list is not decided by ‘cause of death’; however, this is not necessarily what the rule is expressing.

As a result of literal and reasoning gaps, the domain expert felt that the rules were not meaningful enough for gaining new knowledge, and therefore categorised rules 1 to 4 as meaningless. It is noteworthy that the various reasons underlying these gaps correspond to different stages in a knowledge discovery and data mining (KDD) process. In the schema of Fayyad *et al.* (1996), there are five steps in a KDD cycle - namely data selection, pre-processing, transformation, mining, and interpretation. Table 5 lists the alignment between the identified gaps and the five steps.

[Table 5 about here.]

In summary, the experiment demonstrates that the proposed design is capable of discovering unexpected patterns, and the method for finding explanations has generated reasonable explanations for the expert. More importantly, we have identified literal and reasoning gaps which play a critical role in mining of unexpected patterns.

7 Discussion

In this paper, we have presented an approach to unexpectedness-based knowledge discovery from a clinical dataset, to address the challenge of identifying

candidate patterns that may be surprising for a domain expert. The approach generates unexpected association rules based on a representation of the user’s knowledge, proposes explanations, and interactively updates its understanding of the user’s knowledge based on user feedback on the presented rules and explanations. We adopted BNs, populated by expert-derived node associations and data-derived conditional probabilities, as a knowledge representation platform for our purposes, and developed an unexpectedness metric based on calculating the difference between the confidence of a rule and its inferred conditional probability $\hat{P}(B = b|A = a)$ from KB .

The advantage of using BNs for knowledge representation is that by encoding a joint distribution among variables, conditional probabilities can be precisely inferred. A user can simply assign links among variables where a link represents a **relates-to** relation. However, the drawback of the design is that it does not provide any interface for a user to input more specific knowledge. For example, the expert knows that when a patient’s urea reduction ratio is low, his/her risk of cardiac death will be higher; but can only assign a simple ‘urea reduction ratio’ **relates-to** ‘cardiac death.’ Another limitation is that a user typically knows a large number of concepts which are not represented in the data; therefore, the BN cannot fully represent a user’s knowledge.

Computational complexity and scalability

Our approach aligns with well known techniques for tackling complexity in BNs, where the full joint distribution is typically factorized and defined in terms of local conditional distributions obtained via the chain rule. This has the effect of reducing the complexity for the joint probability of n binary variables from $O(2^n)$ to the factorized form, of $O(n * 2^k)$, where k is the maximal number of parents of a node. However, as we report for our experiments in Section 6, even using factorized representations, the exponentially growing conditional probability tables (CPTs) impose a practical memory-based restriction on the number of parent nodes of a child node. We therefore found it impractical to build a full BN representation of the acquired DAG, which was based on all of the ANZDATA variables, as the degree of vertices in the full network would be over 30.

Consequently, our approach achieves tractability by reducing the cardinality of the Bayesian modelling problem by only using part of the full Knowledge Base KB graph for each mined association rule - that is, we build a *local* BN for each rule. Indeed, several techniques are potentially applicable to handling the computational complexity involved in BN inference such as, but not limited to, Pearl’s $\pi - \lambda - propagation$ which has linear complex-

ity for the case of polytrees but exponential complexity for networks with loops (Lacave and Diez, 2002). Moreover, Jaroszewicz *et al.* (2009) resolved this network size issue via sampling from the probability distribution dened by the Bayesian network; their method is to first sort the nodes of the network in topological order and than traverse conditional probability tables of nodes based on the values of its parents.

Complexity of interaction for domain experts

We acknowledge that interactively finding explanations is a complicated mental activity, even for domain experts; therefore, it is difficult to replicate with computational approaches, especially when the search space is limited to the attributes in a dataset. The complication suggests that generated explanations will not necessarily be reasonable to a user. We discussed possible reasons behind various outcomes of a user’s judgement on the surprisingness of rules and reasonableness of explanations.

In our exploratory analysis using the ANZDATA registry, we successfully extracted six unexpected rules out of 1279 association rules, suggesting the *KB*-based filtering step is effective. Two of the six unexpected rules (33.3%) were interpreted by our domain expert as surprising, and the explanations for both rules seemed reasonable to the expert, demonstrating that the approach can successfully identify and explain unexpected rules. We acknowledge that there are many issues that can affect the validity of the rules, including those issues identified by our expert.

Clearly further analysis could be done on each of the association rules to quantify the independent contribution of a particular variable to that association (that is, to account for confounding variables within the dataset) using techniques such as multivariate regression. We consciously make no attempt to quantify the causality of a particular association, but rather present to the expert user a filtered set of statistically dependent associations rules that are not represented in *KB*; the expert then makes a judgement as to whether or not the presented association is interesting and worthy of either adding to the *KB*, or further analysis. The expert is assisted in this evaluation by the explanations generated for the association. Essentially, ‘interesting associations’ are that subset of all high confidence (statistically dependent) associations that are not represented in the current *KB*.

Ideally, we would see gradually increasing knowledge in the *KB*, as commonly ‘known’ associations in the dataset are identified and presented to the expert in early iterations of the mining process, who would then add them to the *KB*. In this way, the approach we present can considered as much an aid to building a *KB* as it is a process of generating new knowledge.

Since ‘interesting’ is relative to a particular KB , initial associations might be expected to be obvious when the KB is small, but would become more *interesting* as the knowledge in the KB increases in subsequent iterations of the mining approach. This remains a target for future research. The scope of our research did not include data preprocessing; nevertheless, it is critical for the overall result, as demonstrated by the literal gaps discussed in Section 6. We propose that there should be a systematic (even automatic) approach to acquire and apply domain knowledge to data preprocessing. One aspect we are particularly interested in exploring further is the temporal interaction present in the dataset as patients in the data are represented by multiple records covering sequential clinical visits.

8 Conclusion

We have demonstrated a promising new technique for discovering surprising patterns in clinical datasets, but acknowledge that the associations we present are exploratory in nature. Further, we note that the results we have presented are derived from the knowledge and beliefs of a single expert; we would expect that different users might have different beliefs and consequently generate different results. We argue that explanation generation is a useful element in the design of unexpectedness-based data mining. The work represents an important development in the field of unexpectedness-based knowledge discovery by addressing the challenge of identifying patterns that are interesting to a domain expert. The scope of this research is on pattern post-processing and pattern interpretation. However, due to the complexity of human knowledge, we found this scope limiting, suggesting the problem of unexpectedness should be addressed at all stages of KDD, i.e. data selection, preprocessing, transformation, data mining, and interpretation.

Acknowledgement

We acknowledge the support to the first author from an Australian Government International Postgraduate Research Scholarship. We thank Kathy Paizis for being so generous with her time and effort as our nephrology expert and James Bailey for insightful discussions.

References

- Agrawal, R., Imieliński, T., and Swami, A. (1993), “Mining association rules between sets of items in large databases,” in *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, ed. S.J.P. Buneman, Washington, D.C., United States, New York, NY, USA: ACM, pp. 207–216.
- Bay, S.D., and Pazzani, M.J. (1999), “Detecting change in categorical data: mining contrast sets,” in *Intern. conf. on Knowledge discovery and data mining (KDD99)*, San Diego, California, United States, New York, NY, USA: ACM, pp. 302–306.
- Bayardo, Jr., R.J., and Agrawal, R. (1999), “Mining the most interesting rules,” in *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, San Diego, California, United States, New York, NY, USA: ACM, pp. 145–154.
- Boneh, T., Nicholson, A., and Sonenberg, L. (2006), “Matilda: A Visual Tool for Modelling with Bayesian Networks,” *International Journal of Intelligent Systems*, 21(11), 1127–1150.
- Brin, S., Motwani, R., and Silverstein, C. (1997), “Beyond market baskets: generalizing association rules to correlations,” in *SIGMOD '97: Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, ed. J. Peckham, Tucson, Arizona, United States, New York, NY, USA: ACM, pp. 265–276.
- Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996), “From Data Mining to Knowledge Discovery in Databases,” *AI Magazine*, pp. 37–54.
- Geng, L., and Hamilton, H.J. (2006), “Interestingness Measure for Data Mining: A survey,” *ACM Computing Surveys*, 38(3), Article 9.
- Haysom, L., Williams, R., Hodson, E., Lopez-Vargas, P., Roy, L.P., Lyle, D., and Craig, J.C. (2009), “Diagnostic Accuracy of Urine Dipsticks for Detecting Albuminuria in Indigenous and Non-Indigenous Children in a Community Setting,” *Pediatric Nephrology*, 24(2), 323–331.
- Jaroszewicz, S., Scheffer, T., and Simovici, D. (2009), “Scalable pattern mining with Bayesian networks as background knowledge,” *Data Mining and Knowledge Discovery*, 18, 56–100.

- Jaroszewicz, S., and Simovici, D.A. (2004), “Interestingness of frequent itemsets using Bayesian networks as background knowledge,” in *KDD*, eds. W. Kim, R. Kohavi, J. Gehrke and W. DuMouchel, ACM, pp. 178–186.
- Kschischang, F.R., Frey, B.J., and Loeliger, H.A. (2001), “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, 47(2), 498–519.
- Kuhn, K.A., Knoll, A., Mewes, H.W., Schwaiger, M., and Bode, A. (2008), “Informatics and medicine,” *Methods of Information in Medicine*, 47(4), 283–95.
- Kuo, Y.T., Lonie, A., Sonenberg, L., and Paizis, K. (2007), “Domain ontology driven data mining: a medical case study,” in *Proc. international workshop on Domain driven data mining (DDDM07)*, San Jose, California, New York, NY, USA: ACM, pp. 11–17.
- Kuo, Y.T., Sonenberg, L., and Lonie, A. (2008), “Finding Explanations for Assisting Pattern Interpretation,” in *IEEE/WIC/ACM International Conf. on Web Intelligence (WI 2008)*, Los Alamitos, CA, USA: IEEE Computer Society, pp. 48–51.
- Lacave, C., and Diez, F.J. (2002), “A review of explanation methods for Bayesian networks,” *IEEE Intelligent Systems Transactions On Knowledge And Data Engineering*, 17(2), 107–127.
- Lim, W.H., Johnson, D.W., and McDonald, S.P. (2005), “Higher rate and earlier peritonitis in Aboriginal patients compared to non-Aboriginal patients with end-stage renal failure maintained on peritoneal dialysis in Australia: Analysis of ANZDATA,” *Nephrology*, 10(2), 192–197.
- Liu, B., Hsu, W., and Chen, S. (1997), “Using General Impressions to Analyze Discovered Classification Rules,” in *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, Menlo Park, CA, USA: AAAI Press, pp. 31–36.
- Liu, B., Hsu, W., Mun, L.F., and Lee, H.Y. (1999), “Finding Interesting Patterns Using User Expectations,” *IEEE Transactions on Knowledge and Data Engineering*, 11(6), 817–832.
- Malhas, R., and Aghbari, Z.A. (2009), “Interestingness filtering engine: Mining Bayesian networks for interesting patterns,” *Expert Systems with Applications*, 36, 5137–5145.

- McDonald, S., Russ, G., Kerr, P., and Collins, J. (2002), “ESRD in Australia and New Zealand at the end of the millennium: a report from the ANZ-DATA registry,” *American Journal of Kidney Disease*, 40(6), 1121–1131.
- McGarry, K. (2005), “A Survey of Interestingness Measures for Knowledge Discovery,” *The Knowledge Engineering Review*, 20(1), 39–61.
- Ohsaki, M., Abe, H., Tsumoto, S., Yokoi, H., and Yamaguchi, T. (2007), “Evaluation of Rule Interestingness Measures in Medical Knowledge Discovery in Databases,” *Artificial Intelligence in Medicine*, 41, 177–196.
- Padmanabhan, B., and Tuzhilin, A. (2002), “Knowledge Refinement Based on the Discovery of Unexpected Patterns in Data Mining,” *Decision Support Systems*, 33, 309–321.
- Padmanabhan, B., and Tuzhilin, A. (2008), “Unexpectedness as a Measure of Interestingness in Knowledge Discovery,” *Decision Support Systems*, 27, 303–318.
- Padmanabhan, B., and Tuzhilin, A. (1998), “A Belief-Driven Method for Discovering Unexpected Patterns,” in *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-98)*, ed. G.P.S. Rakesh Agrawal, Paul E. Stolorz, AAAI Press, pp. 94–100.
- Padmanabhan, B., and Tuzhilin, A. (2006), “On Characterization and Discovery of Minimal Unexpected Patterns in Rule Discovery,” *IEEE Intelligent Systems Transactions On Knowledge And Data Engineering*, 18(2), 202–216.
- Pyle, D. (1999), *Data Preparation for Data Mining*, Morgan Kaufmann.
- Ramamohanarao, K., and Bailey, J. (2003), “Discovery of Emerging Patterns and Their Use in Classification,” in *Australian Conference on Artificial Intelligence*, eds. T.D. Gedeon and L.C.C. Fung, Vol. 2903 of *Lecture Notes in Computer Science*, Berlin, Heidelberg: Springer-Verlag, pp. 1–12.
- Shekar, B., and Natarajan, R. (2004), “A Framework for Evaluating Knowledge-Based Interestingness of Association Rules,” *Fuzzy Optimization and Decision Making*, 3, 157–185.
- Silberschatz, A., and Tuzhilin, A. (1995), “On Subjective Measures of Interestingness in Knowledge Discovery,” in *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, ed. R.U.U.M. Fayyad, Menlo Park, CA, USA: AAAI Press, pp. 275–281.

- Silberschatz, A., and Tuzhilin, A. (1996), “What Makes Patterns Interesting in Knowledge Discovery System,” *IEEE Transactions on Knowledge and Data Engineering*, 8, 970–974.
- Silverstein, C., Brin, S., and Motwani, R. (1998), “Beyond Market Baskets: Generalizing Association Rules to Dependence Rules,” *Data Mining and Knowledge Discovery*, 2(1), 39–68.
- Wang, K., Jiang, Y., and Lakshmanan, L.V.S. (2003), “Mining unexpected rules by pushing user dynamics,” in *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, Washington, D.C., New York, NY, USA: ACM, pp. 246–255.
- Zhang, S., Zhang, C., and Yang, Q. (2003), “Data Preparation for Data Mining,” *Applied Artificial Intelligence*, 17, 375–381.

List of Tables

1	Given association rule $A \rightarrow B$: 2-way contingency table is shown for attributes A and B , where $r_{i,j}$ is the count of itemset $\{A = a_i, B = b_j\}$	29
2	Some examples of the attributes in the preprocessed ANZ-DATA dataset.	29
3	The top six mined unexpected rules, with Support (S), Confidence (C) and Unexpectedness (U) values.	30
4	The user's feedback about the top six mined unexpected rules.	30
5	The alignment between the identified gaps and the five steps of knowledge discovery and data mining.	31

A/B	B			
A	b_1	\dots	b_j	Total
a_1	$r_{1,1}$	\dots	$r_{1,j}$	$\sum_j r_{1,j}$
\vdots	\vdots	\ddots	\vdots	\vdots
a_i	$r_{i,1}$	\dots	$r_{i,j}$	$\sum_j r_{i,j}$
Total	$\sum_i r_{i,1}$	\dots	$\sum_i r_{i,j}$	N

Table 1: Given association rule $A \rightarrow B$: 2-way contingency table is shown for attributes A and B , where $r_{i,j}$ is the count of itemset $\{A = a_i, B = b_j\}$.

Attributes	Definition	Values
agequart	Quartiles of age for patients	0/1/2/3
agestart	Age at first dialysis	0-24.2 / 24.2-48.5 / 48.5-72.7 / 72.7-97.0
bmib	Body Mass Index	1.8-20.4 / 20.4-39.1 / 39.1-57.7 / 57.7-76.3
cig	Smoking?	C / F / N / U / na (not applicable)
canddiag	Cancer ever diagnosed?	N / Y
coronary	Coronary artery disease?	N / S / U / Y / na
cvdb	Cerebrovascular disease?	No / Yes / na
diabetes	Diabetes?	N / T1 / T2
diseaseb	Cause of ESRD	Analgesicnephropathy / Diabetes / GN / Other / PolycysticKD / RefluxNephropathy / Renovascular / Uncertainidiagnosis
dryweight	Base weight in kg	5-53.2 / 53.2-101.5 / 101.5-149.8 / 149.8-198.1
height	Height at entry in meters	0.47-0.86 / 0.86-1.25 / 1.25-1.64 / 1.64-2.03
hypertension	Hypertension requiring treatment	HistoryofHypertension / NotHypertensive / na
lungb	Chronic lung disease?	No / Yes / na
modality	Method or treatment	Graft / HD / PD
referralb	Late referral	LateReferral / Notlatereferral / na
txwlcats	Transplant waiting list status	NotonTransplantList / OnTransplantList / na
hfdeath	Death from heart failure	AliveorNon-HFdeath / DeathduetoHF
middeath	Death from myocardial infarction	AliveorNon-MIdeath / DeathduetoMI

Table 2: Some examples of the attributes in the preprocessed ANZDATA dataset.

No.	Unexpected Rules	S	C	U
1	cvdb=No \rightarrow dryweight=53.2-101.5	0.64	0.70	0.09
2	mideath=AliveorNon-MIdeath \rightarrow dryweight=53.2-101.5	0.62	0.70	0.09
3	lungb=No \rightarrow dryweight=53.2-101.5	0.63	0.71	0.09
4	lungb=No, hfdeath = AliveorNon- HFdeath \rightarrow txwlcatt = NotonTrans- plantList	0.64	0.74	0.07
5	lungb=No \rightarrow txwlcatt = NotonTrans- plantList	0.65	0.74	0.07 e
6	hfdeath=AliveorNon-HFdeath, mideath=AliveorNon-MIdeath \rightarrow txwl- catt=NotonTransplantList	0.63	0.73	0.07

Table 3: The top six mined unexpected rules, with Support (S), Confidence (C) and Unexpectedness (U) values.

Rule No.	Surprisingness of rule	Reasonableness of explanation
1,2,3,4	Meaningless	Meaningless
5,6	Surprising	Reasonable

Table 4: The user’s feedback about the top six mined unexpected rules.

Gap	KDD steps				
	Sel.	Prep.	Trans.	Mining	Inter.
Improper discretization		X	X		
Temporal information loss		X	X	X	
Exhaustive and mutual exclusive property of variables	X	X	X		
Semantic gap		X	X		
Partial alignment of mined rules & experts' knowledge				X	X
Cause-effect context			X	X	X

Table 5: The alignment between the identified gaps and the five steps of knowledge discovery and data mining.

List of Figures

1	The different terms for a user's knowledge and a system's knowledge.	33
2	The dynamics of knowledge in the interactive pattern mining process.	33
3	Some examples of BNs as explanations by conditioning over intermediate variables $\mathcal{X} = \{x_1, x_2\}$	33
4	Process for mining unexpected rules	34
5	The user is presented ranked unexpected rules, their explanations and an interface for modifying the knowledge base. . . .	35
6	Possible pathways for inferring $P(B A)$	35
7	Analysis of Rule 5	36
8	Analysis of Rule 6	36

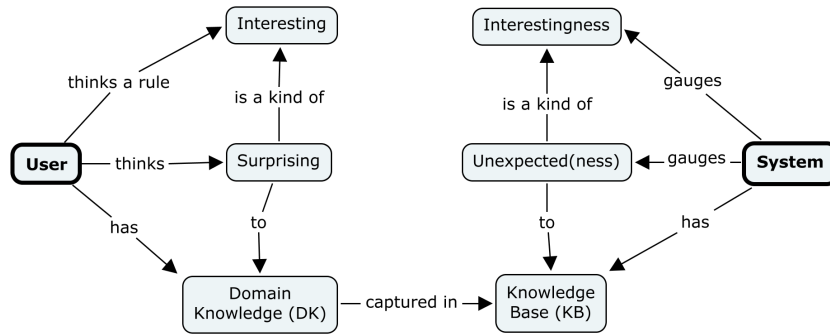


Figure 1: The different terms for a user's knowledge and a system's knowledge.

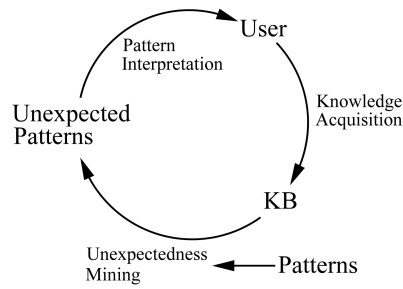


Figure 2: The dynamics of knowledge in the interactive pattern mining process.

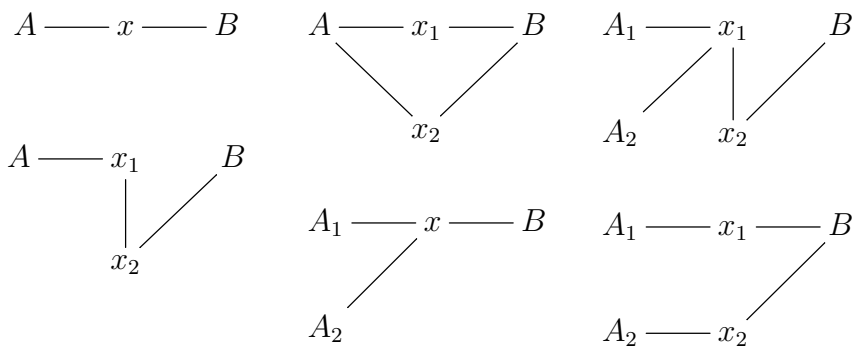


Figure 3: Some examples of BNs as explanations by conditioning over intermediate variables $\mathcal{X} = \{x_1, x_2\}$

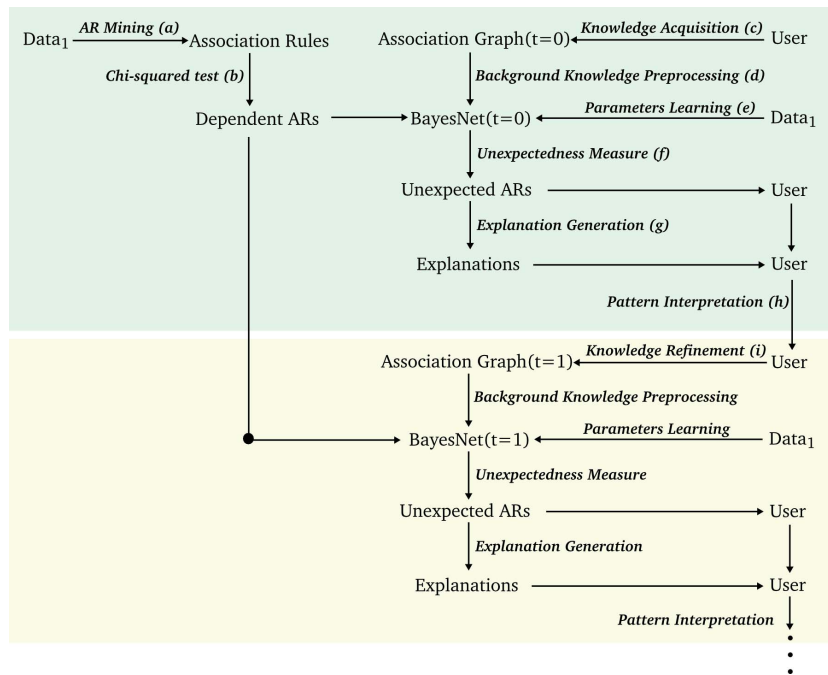


Figure 4: Process for mining unexpected rules

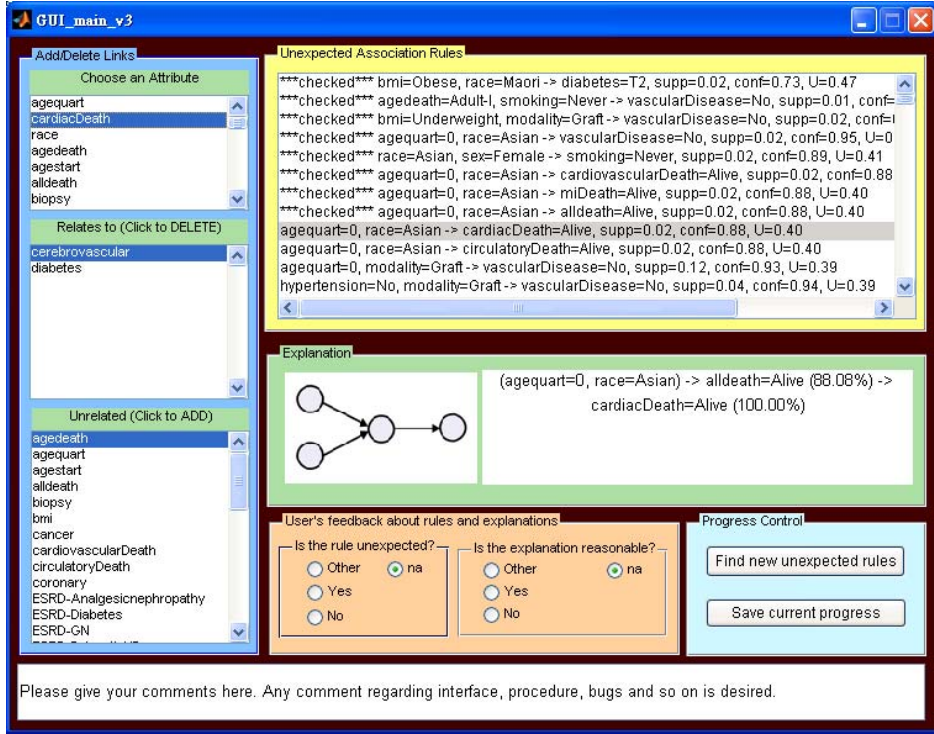


Figure 5: The user is presented ranked unexpected rules, their explanations and an interface for modifying the knowledge base.

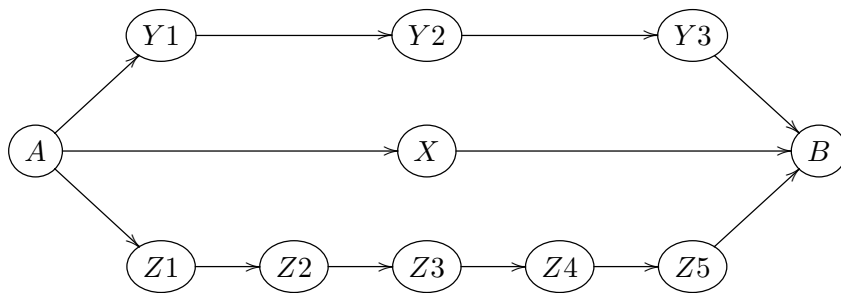


Figure 6: Possible pathways for inferring $P(B|A)$

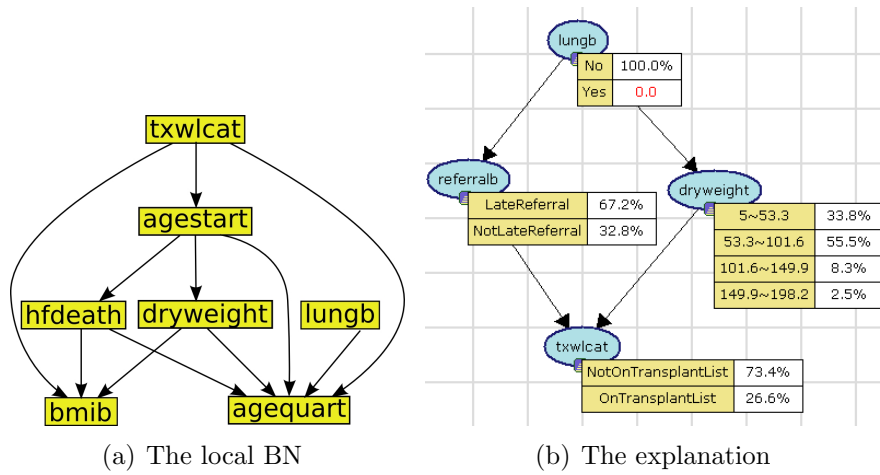


Figure 7: Analysis of Rule 5

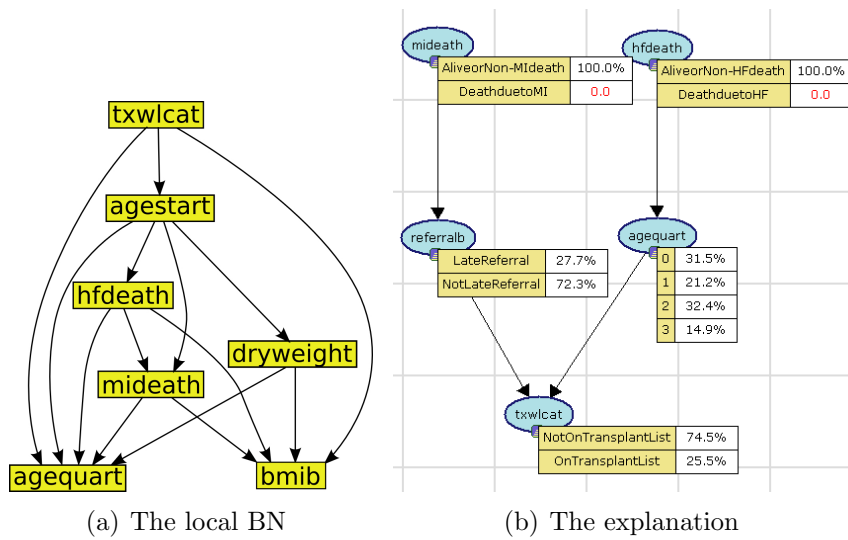


Figure 8: Analysis of Rule 6