Towards a Method for Data Accuracy Assessment Utilizing a Bayesian Network Learning Algorithm

V. SESSIONS
Charleston Southern University
and
M. VALTORTA
University of South Carolina

This research develops a data quality algorithm entitled the Accuracy Assessment Algorithm (AAA). This is an extension of research in developing an enhancement to a Bayesian Network (BN) learning algorithm called the Data Quality (DQ) algorithm. This new algorithm is concerned with estimating the accuracy levels of a dataset by assessing the quality of the data with no prior knowledge of the dataset. The AAA and associated metrics were tested using two canonical BNs and one large-scale medical network. The article presents the results regarding the efficacy of the algorithm and the implications for future research and practice.

Categories and Subject Descriptors: K.6.4 [**Management of Computing and Information Systems**]: System Management—*Quality assurance*; G.3 [**Mathematics of Computing**]: Probability and Statistics—*Probabilistic algorithms*

General Terms: Management, Measurement, Performance

Additional Key Words and Phrases: Bayesian networks, PC algorithm, accuracy levels, data quality assessment

ACM Reference Format:
Sessions, V. and Valtorta, M. 2009. Towards a method for data accuracy assessment utilizing a Bayesian network learning algorithm. ACM J. Data Inform. Quality 1, 3, Article 14 (December 2009), 34 pages. DOI = 10.1145/1659225.1659227. http://doi.acm.org/10.1145/1659225.1659227.

Authors' addresses: V. Sessions, College of Science and Mathematics, Charleston Southern University, P.O. Box 118087, Charleston, SC 29423; email: vsessions@csuniv.edu; M. Valtorta, College of Engineering and Computing, University of South Carolina, Columbia, SC 29210; email: mgv@cse.sc.edu.

1. INTRODUCTION

Data quality assessments are becoming an integral part of information systems, records management systems, and database development [Kaplan et al. 1998; Laudon 1986; Ballou and Tayi 1999]. The authors were previously involved in researching methods to incorporate these data quality assessments into Bayesian learning algorithms [Sessions and Valtorta 2006]. During this research, it was discovered that many researchers in the field of artificial intelligence, and specifically Bayesian research, do not consider data quality elements when creating and using learning algorithms. It was the authors' belief that altering the Bayesian learning algorithms to account for known or suspected data quality problems (specifically the accuracy dimension of data quality) could improve the results of the learning algorithms. A new method for using these data quality assessments in the PC (named by its creators, Peter Spirtes and Clark Glymour, after the initials of their first names) Bayesian network learning algorithm was developed, tested, and proven to be of use to the Bayesian learning community. Due to the success of this initial research, it was theorized that by reversing the algorithm, it could be harnessed to create meaningful data quality assessments. These assessments could help estimate the level of accuracy of a given dataset. These new algorithms and methods could be particularly useful in situations in which there is no known assessment regarding the accuracy of the data or prior knowledge of the context of the data. The authors' original research used data quality assessments to improve the output of the PC algorithm. In this research, the algorithm is reversed in order to assess the accuracy of the data itself.

The remainder of this article is presented as follows. Section 2 presents a literature review of the current state of the field of data quality and data quality assessment techniques. Section 3 then presents background information regarding Bayesian Networks (BNs). Within this section previous research in incorporating data quality assessments into BNs and particularly the DQ algorithm is also reviewed. Sections 4 and 5 present the AAA test methods, and experimental results. Sections 6 and 7 conclude with implications of the research and future avenues for experimentation.

2. LITERATURE REVIEW

Data quality is a young field with several facets and avenues for research. One area of data quality focuses on the dimensions, or measures, of data quality and their formal definitions. One pivotal study in this area was completed by Lee et al. [2002] and categorized the various views of data quality from both a practitioner's view and an academic view. These researchers then consolidated the two views into the PSP/IQ (Personal Software Process/Information Quality) model. This model has both objective and subjective dimensions for data quality, as shown in Table I.

Some objective measures include free-from-error, concise representation, completeness (missing data fields), and timeliness. Some subjective measures include relevancy, understandability, reputation, and ease-of-use. While there are many important dimensions of data quality, we will focus solely on data

Table I. PSP/IQ Model from Lee et al. [2002]

The PSP/IQ model		
	Conforms to specifications	Meets or exceeds consumer expectations
Project Quality	Sound information IQ dimensions Free-of-error Concise representation Completeness Consistent representation	Useful information IQ dimensions Appropriate amount Relevancy Understandability Interpretability Objectivity
Service Quality	Dependable information IQ dimensions Timeliness Security	Usable information IQ dimensions Believability Accessibility Ease of operation Reputation

accuracy/free-from-error for this research. The PSP/IQ model was developed from a variety of other models such as the department of defense's data quality guidelines [Cykana et al. 1996] and these were consolidated in Lee et al.'s [2002] work in developing the PSP/IQ model. Most researchers include accuracy in their list of data quality components, and we will follow the definition and ideas of Wand and Wang [1996] for creating our definition of accuracy. Accuracy is considered how close a measurement, or data record, is to the real-world situation it represents. It is normally considered an intrinsic data quality component, independent of its context within the system. The authors chose to focus on the element of accuracy for many reasons. First, one of the authors' main practical research areas is in assessing the quality of Law Enforcement (LE) datasets. Of traditional concern to LE professionals are the quality dimensions of completeness, timeliness, accuracy, and consistency as defined by formatting errors. Completeness can be categorized in a straightforward manner by assessing the number of missing fields. The timeliness of the data is measured in many ways, notably by the timestamp of the entry of the data, and can be determined readily from the dataset. Consistency of the data can also be categorized in a straightforward manner by assessing the number of data fields that do not meet a standard format. The authors are aware of no algorithms or tools that can assess the accuracy of LE data without completing sampling of the data and fact checking against other sources. The authors therefore see a need in this field for such a tool and concentrated their efforts in this research on the element of accuracy assessment. Secondly, accuracy is an easily manipulated field and by controlling the amount of inaccurate data in our test sets we ensure that the algorithm's output is the true dependent variable in our tests' setup.

Adding to our understanding of accuracy, particularly as it applies to the field of Bayesian learning algorithms, is complementary work in evidential update by Vomlel [2004]. In the field of BNs, evidential update refers to the updating of a probability based on new evidence. In his work, Vomlel defines accuracy as

$$P(A = T) = \frac{tp + tn}{tp + tn + fp + fn},$$

where T is a data source, A is the event T is reporting on, *tp* is the number of true positive data points, *tn* is the number of true negative data points, *fp* is the number of false positives, and *fn* the number of false negatives. He further defines two criteria, that is, sensitivity and specificity, that are important for determining how data sources should update the probability table of A. Sensitivity is the test's true positive rate, *tp*, and specificity the true negative rate, tn. While Vomlel [2004] uses these metrics for evidential update, we harness it to evaluate the BNs generated using data sources of low accuracy.

Another area of data quality research lies in data quality assessment methods. The focus in the last 5–10 years in data quality studies has been to determine a method for assessing our data in a more uniform and consistent way. Most of these assessment methods can be incorporated into our research, but our methods meld most closely with the Total Data Quality Management (TDQM) assessments. Following Lee et al. [2002], the most common functional form of accuracy assessments involves a ratio of the desired outcomes divided by total outcomes subtracted from 1. This gives us a ratio between 1 and 0 that describes our data quality. We interpret a number closer to 1 as more accurate and closer to 0 as inaccurate. In our research, it is not necessary to perform an assessment of the quality of our data, because we use test datasets from BNs of high quality and corrupt them with inaccurate data. Therefore, we already know the quality of our data because we have created that ratio and quality level. This is necessary for our development effort, and without knowing the data quality with certainty, the effect on the machine learning algorithm would not be the true dependent variable in our experiments.

A complementary field to data quality is data cleansing. There are two main categories of data cleansing methods: manual and automatic. Manual techniques for correcting data inaccuracies involve a user reviewing the data and correcting any errors manually. For large datasets this is certainly impractical and we must turn to automatic techniques such as those found in Maletic and Marcus [2000]. These researchers review the general methods for error detection: statistical, clustering, pattern based, and association rules.

Statistical methods identify outliers in the dataset (potential errors) by finding those data points that are outside of a range of a few standard deviations from the mean. The number of standard deviations can be customized by the users. In their study, Maletic and Marcus [2000] found that five standard deviations were optimal. Once outliers are identified they can be manually or automatically removed or corrected. Clustering methods work similarly, but instead of identifying outliers using standard deviation and mean, it clusters the points based on Euclidean distance and uses the resulting clusters to identify outliers. Pattern-based approaches determine groups of records with similar characteristics and group them into patterns. Those records which do not conform to the patterns are then grouped based on various metrics such as distance from mean for further investigation. Association rules work in a similar manner, finding how often two records, or data items of the record, occur together in a relationship such as != or =. Once these associations are discovered, we can identify those records that are outliers, in this case those that would normally have been associated with each other, but for some reason (possibly an error) they are not.

Statistical data cleansing methods are commonly used to find errors such as a wrong birthday, based on normal format of MM-DD-YYYY, name misspellings, etc. A large portion of data cleansing research has been focused towards entity resolution, that is, making sure two records that represent the same real-world event are represented by the same record in the dataset. For this problem all of the preceding broad techniques can be customized to form a solution for this category of error. Sung et al. [2002] review many of the techniques used for entity resolution. The standard method is to sort the database and then use the preceding techniques and others to check if two records are identical. The systematic checking of every record against every other in the dataset can be very inefficient, leading to a runtime of $O(N)$, where N is the number of records. There are several methods to reduce this running time, including the sorted neighborhood method and various customizations of this algorithm. This is discussed in detail in Sessions and Valtorta [2006] and will not be furthered studied here.

While there are many methods for dealing with faulty data, we will assume in our research that all of these methods have been employed and that our datasets are still inaccurate. This assumption is important because it allows us to start our research at the point of data entry so that the dataset does not first need to be cleansed, and also because in many instances the data cannot be cleansed any further.

3. BACKGROUND REVIEW OF BAYESIAN NETWORKS

We will attempt to cover the fundamentals of BNs without overwhelming the reader with complex underlying formulas. For a more detailed mathematical description, we recommend Cowell et al. [2002], Jensen and Nielsen [2007], and Neapolitan [2004].

A BN is used to model a domain of knowledge using a set of nodes (representing variables) and a set of directed edges between the nodes. The directed edges represent a set of dependencies between the nodes. For example, we can represent the state of a wet lawn using the BN in Figure 1.

Mr. Holmes deduces whether his lawn is wet because it has rained or because his sprinkler is working. He gathers evidence by looking to see if Mr. Watson's and Mrs. Gibbon's lawns are also wet. The strength of these dependencies is modeled as a probability, normally represented by a conditional probability table. The conditional probability tables for three nodes in our wet lawn example are shown in Figure 2.

As we see from this example, Holmes's wet lawn is dependent upon either the sprinkler or the rain. If it has not rained or if the sprinkler has not been working, the lawn will not be wet. If it has either rained or the sprinkler has

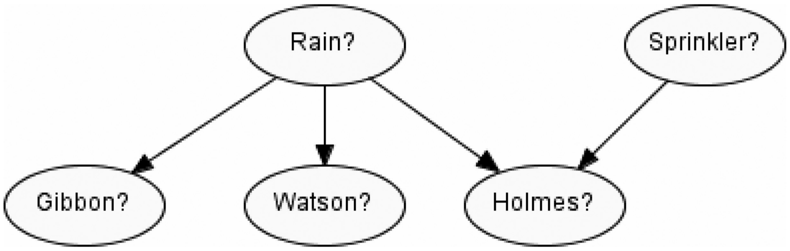
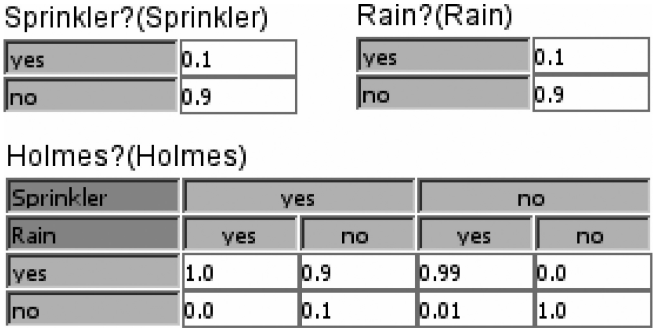


Fig. 1. Wet lawn.



Sprinkler?(Sprinkler)

yes	0.1
no	0.9

Rain?(Rain)

yes	0.1
no	0.9

Holmes?(Holmes)

Sprinkler	yes		no	
Rain	yes	no	yes	no
yes	1.0	0.9	0.99	0.0
no	0.0	0.1	0.01	1.0

Fig. 2. Wet lawn probability tables.

been working it is a 90%–99% chance that it is wet, and if it has both rained and the sprinkler is working, it is 100% chance that the lawn is wet.

Using Bayes rule, or the chain rule, we can also update our beliefs about the network. Bayes rule is stated as

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)},$$

where $P(A|B)$ is the probability of A given B, $P(A)$ and $P(B)$ are the prior probabilities of A and B, respectively, and $P(B|A)$ is the probability of B given A. Based on a combination of evidence and prior probability of a variable, the probability of certain results will increase or decrease.

This is a small sample BN, but we can see both the power of the BN and how quickly its complexity grows as more nodes and dependencies are added. These networks are developed in a variety of ways. One method is to create the network by interviewing Subject Matter Experts (SMEs). The SMEs draw on years of experience in their field to develop a BN consisting of variables and dependencies, and also populate the probability tables which represent the strength of these dependencies. This method works well, but is reliant upon the experts used. We would prefer to learn directly from domain data itself, and thus a set of algorithms has been developed for this purpose. One algorithm is the PC algorithm explained in detail in Neapolitan [2004]. This algorithm is used by the Hugin^{TM} Decision Engine [Olesen et al. 1992; Madsen et al. 2005] which was an integral part of our research. This algorithm draws directly on the data itself to create the nodes and directed edges which make up the structure of the BN. It then uses a second algorithm, the Expectation Maximization (EM) algorithm, to create the probability tables or the strength of the connections between variables. We include a simplified description of the PC algorithm here and refer the reader again to Neapolitan [2004] for more detailed information.

The PC algorithm begins by gleaning the node names and states from the datasets. In a relational database, this would be done by field name and then a process of scanning the fields to determine the possible states of the variable. In our wet lawn example, we would find a node for Holmes that can be in state wet or dry, which would also be related to Watson's lawn which can also be wet or dry. The algorithm then creates a complete graph from the node, that is, all nodes are then connected to each other by undirected edges. An independence test is then performed to determine whether the variables are actually dependent. This is done by using a score based on I-divergence (also called KL-divergence or cross-entropy), a nonsymmetric measure of the distance between two distributions, which we will call the G^2 score [Spirtes et al. 2000]. If X and Y are random variables with joint probability distribution P, and sample size m, then the G^2 score is defined as

$$G^2 = 2m * I(X, Y) = 2m * \sum_{x,y} P(X, Y) \log \frac{P(X, Y)}{P(X)P(Y)}.$$

For simplicity, we denote by $P(X)$ and $P(Y)$ the marginal of $P(X, Y)$ on variables X and Y, respectively. This score measures the degree of dependence between two variables and has value 0.0 only for independent variables. Therefore any edges with a cross-entropy score of 0.0 are removed.

Once all of the independent edges are removed, the algorithm orients the edges in a series of steps.

(A) Head to Head Links: If there exists three nodes X, Y, Z , such that $X - Z - Y$ are connected, and X and Y are independent given a set not containing Z , then orient the nodes as $X \rightarrow Z \leftarrow Y$.

(B) Remaining Links: Three more rules govern the remaining links, each use the assumption that all head-to-head links have already been discovered.
   (i) If there exists three nodes X, Y, Z , that are connected as $X \rightarrow Z - Y$, and X and Y are not connected, then orient $Z - Y$ as $Z \rightarrow Y$.
   (ii) If there exists two nodes X, Y, that are connected $X - Y$ and there exists a path from X to Y, then orient $X - Y$ as $X \rightarrow Y$.
   (iii) If there exists four nodes X, Y, Z , W that are connected $X - Z - Y, X \rightarrow W, Y \rightarrow W$ and X and Y are not connected, then orient $Z - W$ as $Z \rightarrow W$.

(C) If there are any links left, these are oriented arbitrarily, avoiding cycles and head-to-head links.

After orienting the links, the algorithm is complete.

In reality, the cutoff for independence of variables is not set at exactly G^2= 0.0. This is too tight a constraint for "real" data. In Hugin's$^{\text{TM}}$ implementation of the PC algorithm, the determination of independence is partially handled by a value called the Significance Level (SL). Hugin$^{\text{TM}}$ exploits the fact that G^2 is roughly χ^2 and if its tail probability, or α, is less than the significance level, dependence is assumed. This significance level is by default set at 0.05 consistent with work by Madsen et al. [2005]. Therefore, the smaller the significance level, the larger a cross-entropy score that is considered independence.

In our previous research, we sought to determine ways that the PC algorithm could be improved by incorporating data quality assessments into the algorithm. The aforementioned cross-entropy score presented one such opportunity for the incorporation of these assessments. In studying the PC algorithm, the authors noted that it degraded very quickly under inaccurate data. The cross-entropy score and subsequent relationship with the significance level was discovered to be a key part of this degradation. If the two variables are deemed independent by the cross-entropy test, the edge is removed, otherwise it remains. The number of these independence tests is dependent upon the number of nodes and the degree of each node (number of edges connecting the node). We will call the number of nodes n, and the maximum degree of the nodes, k. The upper bound or worst-case number of independence tests is then given by

$$\frac{n^2(k+1)(n-2)^k}{k!}.$$

As is noted in Spirtes et al. [2000], and which we will confirm here, normally (in data of high accuracy) this worse case is not achieved. From early research which we used in the development of our methods, we determined that the average degree of each node in our canonical BNs is 2.22, and the maximum degree of these normal networks is 5. These numbers are empirically derived from examining ten sample canonical BNs and they serve us well in determining why the runtime is greater under situations of inaccurate datasets. Using these average degree and maximum degree numbers for normal BNs created with clean data, this gives us a number of independence tests for Alarm [Beinlich et al. 1989], one large well-known example BN, of

$$\frac{37^2 * 6 * 35^5}{5!} = 3.60 * 10^9.$$

However, when we give the algorithm inaccurate data we hit the worst-case scenario of the number of independence tests needed for a network at 37 nodes and maximum degree of 36 (a complete graph)

$$\frac{37^2 * 37 * 35^{36}}{36!} = 5.25 * 10^{18}.$$

This number of independence tests makes the computation impractical.

Why do we hit the worst case because of our inaccurate data? In order to determine if each of the nodes is independent we use the cross-entropy score (this is based on I-divergence also called KL-divergence).

$$CE(X,Y) = \sum_{x,y} P(X,Y) \log \frac{P(X,Y)}{P(X)P(Y)}$$

```
DQ Algorithm
Input from User: Data Files.
    Create single data file from the imported files.
    Set four different significance levels from ranging 0.05 − 0.00005
For each significance level
    Using Hugin API
        Create empty domain;
        Learn structure(data file); //Hugin uses PC algorithm
        Learn probabilities(data file); //Hugin uses EM algorithm
 Export to .net file;
        Determine average degree per node of network;
        Score = Absolute Value(2.22 − degree of network);
File to Output = Network with lowest score

Output: Hugin .net file
```

Fig. 3. DQ algorithm pseudocode.

This score is 0.0 when the nodes are independent and is positive otherwise. If we have even one data record that points to a dependency, the score is nonzero, and therefore many of the edges from the complete graph remain in the learned structure. The reason the algorithm can complete its calculations under small amounts of inaccurate data is because of the number of significant digits we use for the score; in other words, where we decide to round to zero. In Hugin's$^{\text{TM}}$ implementation of the pc algorithm, where to determine independence is partially handled by the significance level. Inaccurate data affects this algorithm severely because it only takes a few (inaccurate) data records which point to a dependency to keep the edge in the learned graph structure and therefore lead to the worst-case runtime scenario.

Using this knowledge of the cross-entropy score we developed a set of algorithms as described in Sessions and Valtorta [2006]. The most promising of these was the DQ algorithm presented in Figure 3.

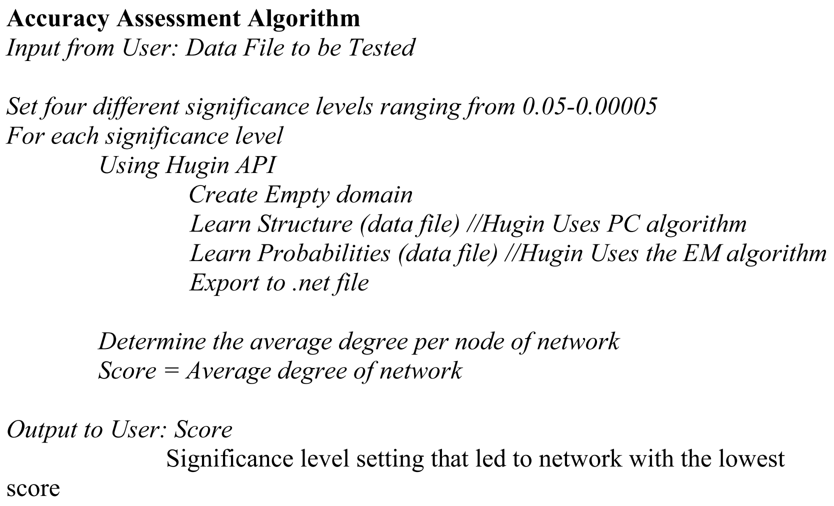
This algorithm uses a different strategy than most. Instead of limiting the datasets given to the algorithm to only those of high quality or weighting the datasets based on quality, we stretch the limits of what the PC algorithm believes is independence by increasing the significance level we use in the cross-entropy score. Remember from our discussion of significance level that the larger the significance level, the smaller the cross-entropy score that is considered independence. So one would assume a significance level of 0.00005 would have fewer remaining dependencies or edges than a significance level of 0.05. In a sense, we are stretching the parameters for what is a meaningful edge and what is not. A good illustration of how this works is to think of how we might target our marketing advertisements under high- and low-quality data. If we have high-quality data we may be able to target a specific region of town for our marketing, maybe even a specific subdivision in which we know the makeup of the residents. If our data is of low quality, we may only be able to limit our marketing region to a certain town so that while we are not missing an important portion of the market because we have low-quality data, we are targeting a larger area than we would need to if we had high-quality data. By decreasing the significance level, we are basically lowering the bar on what is considered a dependency and therefore removing more incorrect edges. Using the algorithm we were able to create networks with 5–6 fewer incorrect edges than by using a baseline method. Full test setup and results of this algorithm and the average degree heuristic are shown in Sessions and Valtorta [2006]. These initial promising results led us to believe that if we reversed the algorithm we could determine the relative accuracy levels of datasets by determining the significance level at which the network created a stable BN. Those sets that created a stable network at a significance level closer to 0.0 would be considered of higher quality that those that created stable networks at higher significance levels. This is discussed in Section 4.

4. ACCURACY ASSESSMENT ALGORITHM

Accuracy is often a difficult dimension to calculate. In small databases it may be possible to manually or automatically check each data element and determine its accuracy. In large databases, however, this is virtually impossible. Therefore we often rely on spot checking or sampling of data fields and extrapolating the results over the entire datasets. This is a common way to approximate the accuracy of the datasets given time and sampling. Companies with large databases can often invest countless man hours and dollars to improve the accuracy of their data. These improvement methods work well for companies and organizations that collect relatively similar types of data, and have the means to invest in these programs. However, as was mentioned previously in the article, the field in which one of the authors' research is based is that of law enforcement datasets. These datasets are compiled from a variety of state, local, tribal, and multinational agencies. These sets are often of differing quality levels and in most cases, no one governing body has the authority or means to correct all of the inaccuracies in the data (even if given authority or made into laws, it is unclear whether the manpower or money involved in achieving higher levels of accuracy exists). To compound this problem, if crises do occur, the data from various agencies is often pulled into data mining and decision making tools and used to make quick decisions on how to react to a threat. With this fast time table for data usage, it is infeasible to spend hours sampling the datasets to determine which are of high quality and fit for use in the tools. From previous studies related to how problem complexity increases under inaccuracy [Blake and Mangiameli 2008], however, we know that excluding inaccurate sets is essential for better decision tool runtime and for overall results.

For these reasons, the authors determined the need for a faster, automatic approximation of dataset accuracy. Based on previous success with the DQ algorithm, we were hopeful that a reversal of the algorithm might lead to such a method. This new algorithm is called the Accuracy Assessment Algorithm (AAA) and the pseudocode is presented in Figure 4.

First, we set four different significance levels between 0.05 (Hugin$^{\mathrm{TM}}$ default) and 0.00005. As discussed earlier, the 0.05 significance level relates



```
Accuracy Assessment Algorithm
Input from User: Data File to be Tested

Set four different significance levels ranging from 0.05-0.00005
For each significance level
        Using Hugin API
                Create Empty domain
                Learn Structure (data file) //Hugin Uses PC algorithm
                Learn Probabilities (data file) //Hugin Uses the EM algorithm
                Export to .net file

        Determine the average degree per node of network
        Score = Average degree of network

Output to User: Score
              Significance level setting that led to network with the lowest
score
```

Fig. 4. AAA pseudocode.

to a higher cross-entropy score that is considered dependence. The 0.00005 significance level relates to a lower cross-entropy score that is considered dependence. Therefore one would expect the 0.05 level to equate to a stronger set of network dependencies and edges. The choice of four significance levels ranging from the default of 0.05 down to 0.00005 was to give an incremental decrease of significance level to determine if a corresponding increase or decrease in average degree of nodes could be found. The range has no other significance and could also start higher or lower in future testing. For each of the four significance levels, the PC algorithm was used to learn a BN. Then the average degree of nodes within the network was calculated and a score based on the average degree per node was calculated. Output to the user included both the score of the network and the significance level setting that was used in the creation of that network.

Many elements of our previous research led to the development of this algorithm and the reasoning behind it is as follows. Many datasets collected and stored in a relational database are correlated in some fashion and therefore stored together. In a medical database, for example, the correlation between a patient with lung cancer also being a smoker may be high. This would not be the case in every scenario but overall this correlation would be high. In the case of lung cancer and phone number or city of birth, however, there would likely be little correlation. The PC algorithm seeks to learn these sets of correlations by beginning with a fully connected network: all nodes/fields connected via edges. It then systematically eliminates the edges that it determines do not exist: a correlation between phone number and lung cancer, for instance. As we found when testing and developing our DQ algorithm, the PC algorithm has a difficult time eliminating noncorrelated edges under inaccurate datasets. This refers back to our earlier discussion of significance level and its effect on the removal of edges. It was therefore hypothesized that we could approximate accuracy levels of datasets by examining the number of learned edges, or

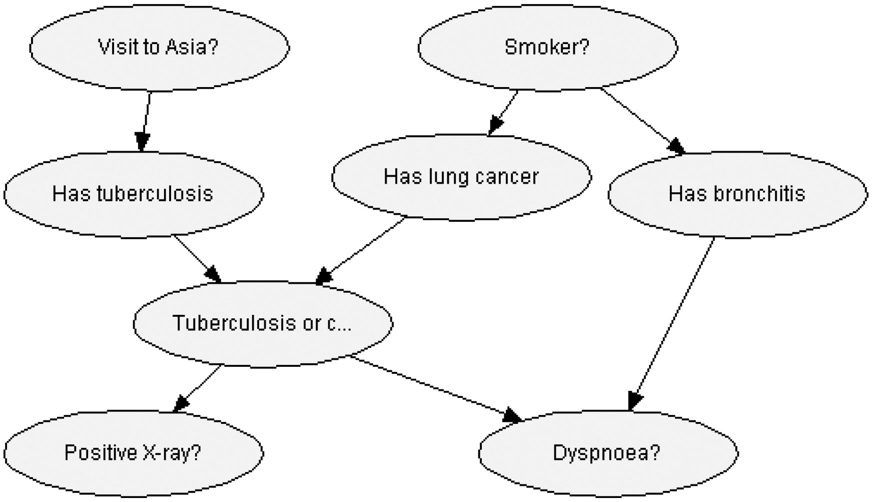


Fig. 5. Visit to Asia.

average degree of nodes, in a network at the default significance level of 0.05, or perhaps by examining the difference in the average degree of nodes at various significance levels. In order to formally test this theory we developed the following null hypotheses.

Hypothesis I. Using the AAA with SL = 0.05 for a given network, the average degree per node will be the same for both accurate and inaccurate data.

At the default PC algorithm significance level (referred to as SL) of 0.05, the AAA will determine no difference between the average degree per node in a network learned with accurate data versus one learned from inaccurate data.

Hypothesis II. Using the AAA with both accurate and inaccurate data for a given network, the average degree per node will be the same for SL = 0.05 and SL = 0.00005.

The AAA will determine no difference between the average degree per node in a network learned with a PC significance level of 0.05 versus a significance level of 0.00005 when presented with accurate and inaccurate data.

For both hypotheses a regression analysis was used in the analysis of the data, and statistical significance is defined as an f-measure or f-significance lower than the standard alpha of 0.05.

These hypotheses were tested using the methodology presented in Section 5.

5. EVALUATION OF THE AAA

5.1 Test Setup

We used two canonical BNs and one larger scale medical BN to test our algorithm. We will describe each network here. Visit to Asia is a fictitious, canonical BN created by Lauritzen and Spielgelhalter [1988] and shown in Figure 5.

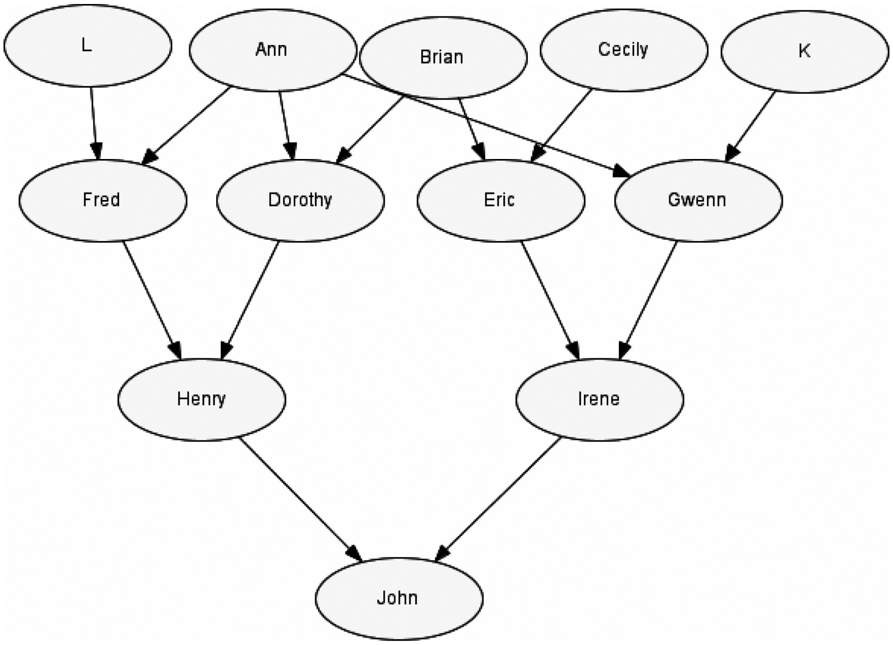


Fig. 6. Stud farm.

The network represents a situation as follows:

> Shortness-of-breath (dyspnoea) may be due to tuberculosis, lung cancer, or bronchitis, or none of them, or more than one of them. A recent visit to Asia increases the chances of tuberculosis, while smoking is known to be a risk factor for both lung cancer and bronchitis. The results of a single chest X-ray do not discriminate between lung cancer and tuberculosis, as neither does the presence or absence of dyspnoea.
> Lauritzen and Spielgelhalter [1988].

As shown in Figure 5 and its description, Visit to Asia is a relatively small BN, consisting of 8 nodes and 8 edges between the nodes. This network is well known in the field of BNs and is a very common test case within the field.

Our second network is Stud Farm, a network modeled on the genealogy of a set of horses [Jensen 1995]. Represented in Figure 6, the network represents the genetic disposition of the horses for a life-threatening genetic disease carried through a recessive gene. A horse is classified as either pure, carrier, or sick. A pure horse has no history of the disease, a sick horse has the disease and subsequently dies, and a carrier has a history of the disease in the family but has not yet shown signs of the disease. A contraction of the disease occurs if two carriers mate and the recessive genes cause the disease to occur. The likelihood of being a carrier increases as descendents are diagnosed with the disease.

This network is larger with 12 nodes and 14 edges and represents a different set of dependencies than Visit to Asia. It is an interesting network because it represents genealogy and is therefore an interesting test set for dependencies.

Finally, the Alarm network is the largest of the BNs. It is ascribed to Beinlich et al. [1989] as a expert system for identifying anesthesia problems in the operating room. There are 37 nodes and 46 edges in the network and due to its size we will not show it here. Of the 37 nodes, 8 are variables that represent diagnostic problems such as insufficient anesthesia or analgesia, 16 are findings such as increased heart rate, and 13 are intermediate variables connecting the diagnostic problems to findings. Each node has from two–four possible values. It is the largest of our sample networks and is a standard test set in the field of BNs. The alarm network was developed in the field and is a practical network. It is representative of one of the larger "real world" networks that one might encounter in the LE arena. The public dataset for the South Carolina Sex Offender Registry, for example, has 17 identifying fields and an additional 9 fields for offense, related data. Therefore Alarm was chosen as a representative test set for the upper limits of the algorithm.

Using these three networks we developed our test datasets by using the dataset generation tool of the Hugin$^{\text{TM}}$ Decision Engine [Olesen et al. 1992; Madsen et al. 2005]. First, we modeled two test networks per sample BN: one that was considered the "true" state of the BN (with Hugin$^{\text{TM}}$ default example network potentials), and another, "false" network in which all of the potentials were set at 0.5 (equal likelihood of either result). The potential tables are included as an Appendix to this article. Partially inaccurate or dirty datasets for Visit to Asia and Stud Farm were then generated in the following way. Using the data generation tool, datasets were generated from both the "true" network and the "false" network. These were then combined in different ratios of true/false data. When testing our DQ algorithm it was discovered that the PC algorithm degraded under what the authors considered small amounts of inaccurate data: less than 5% false data. Therefore we tested the AAA at small amounts of inaccurate data: 100/0, 99/1, 98/2, 97/3, 96/4, 95/5 and then larger increments: 90/10, 85/15, 80/20, 75/25, 70/30, 65/35, 60/40, 55/45, and 50/50, assuming that results at the 75/25 would be similar in nature to the 50/50 sets. If this proved false we would test at smaller increments. It was decided not to test at higher amounts of inaccurate data because most data is not more inaccurate than 50/50 [Blake and Mangiameli 2008]. Also as we discuss in the results section, when testing at 50/50 data the score has maxed out and going farther does not add to the discussion. As an example of dataset generation, in order to generate a 90/10 inaccurate set for 100 records, one would generate 90 cases from the "true" network and 10 cases from the "false" network and combine them for a 100 record set. We used datasets of 10,000 data records in each experiment in order to limit the effects of variance due to the size of the datasets. Table II shows the overall categories developed and a breakdown of true/false data within each category.

We were prohibited from using the larger ratios in the Alarm network due to the size of the network and subsequent size of the datasets. As mentioned in Section 2 of this article, inaccurate datasets cause us to reach worst-case

Table II. Test Datasets

	Number of True BN	**Data Records False BN**
Golden Standard (100/0)	10000	0
99/1	9900	100
98/2	9800	200
97/3	9700	300
96/4	9600	400
95/5	9500	500
90/10	9000	1000
85/15	8500	1500
80/20	8000	2000
75/25	7500	2500
70/30	7000	3000
65/35	6500	3500
60/40	6000	4000
55/45	5500	4500
50/50	5000	5000

runtimes for the independence test of the PC algorithm. Because of this and the size of Alarm, we were unable to run tests for inaccuracy levels greater than 5% inaccurate data, or a 95/5 ratio. Greater inaccuracy levels were prohibitive due to the amount of virtual memory needed compared to that of our machine. Therefore for the Alarm network, we tested sets in ratios of 100/0, 99/1, 98/2, 97/3, 96/4, and 95/5. This limitation is discussed in Section 6.

In order to test our hypotheses we collected metrics for average degree of node at each of four significance levels: 0.05, 0.005, 0.0005, and 0.00005.

5.2 Test Results

We present the results of the AAA testing of Visit to Asia in tabular form in the two tables that follow. In Table III, the score for each significance level (average degree of the nodes) is shown for significance levels of 0.05, 0.005, 0.0005, and 0.00005. Table IV shows the score at 0.05, and 0.00005, and difference between the scores at 0.05, and 0.00005 (in order to test Hypothesis II).

As can be deduced from a quick scan of the tables, score at 0.05 leads to a better measure of inaccuracy rates than a difference in scores at the 0.05 and 0.0005 levels. We provide a graph of the score at 0.05 results here in Figure 7 and test for significance of the findings for Visit to Asia.

The initial results for Visit to Asia at 0.05 suggested the data were asymptotical, that is, rising rapidly as the data become more dirty but leveling off around 4%. Therefore we performed a regression analysis for the results of Visit to Asia and the full results of this analysis are shown in Table V.

The significance *F* level, or *p*-value, is 0.027. This is less than our alpha of 0.05 therefore for our Visit to Asia results we contradict and therefore reject our null hypothesis that there will be no difference in score for inaccurate versus accurate datasets. A similar function could not be found for the result differences between score at 0.05 and score at 0.0005, therefore our second hypothesis is not rejected.

We show now the results of our tests for Stud Farm in both tabular and graphical form in Table VI and VII, and Figure 8.

Table III. Accuracy Assessment Algorithm Results for Visit to Asia

Percent Inaccurate Data	Score at 0.05	Score at 0.005	Score at 0.0005	Score at 0.00005
0	3.00	2.75	2.25	2.25
1	5.00	4.75	4.25	5.00
2	5.75	5.50	5.25	5.00
3	5.75	5.75	5.25	4.75
4	5.75	5.25	5.25	5.25
5	6.25	5.50	5.25	5.00
10	6.25	5.75	5.25	5.25
15	6.25	5.75	5.00	5.00
20	5.97	5.75	5.50	5.25
25	6.50	6.25	5.75	5.75
30	5.97	5.75	5.75	5.75
35	6.25	6.25	5.75	5.75
40	6.50	6.50	6.50	6.25
45	6.75	5.75	5.50	5.25
50	6.25	5.75	5.50	5.50

Table IV. Continued Accuracy Assessment Algorithm Results for Visit to Asia

Percent Inaccurate Data	Score at 0.05	Score at 0.00005	Score Diff. 0.05-0.0005
0	3.00	2.25	0.75
1	5.00	5.00	0.00
2	5.75	5.00	0.75
3	5.75	4.75	1.00
4	5.75	5.25	0.50
5	6.25	5.00	1.25
10	6.25	5.25	1.00
15	6.25	5.00	1.25
20	5.97	5.25	0.72
25	6.50	5.75	0.75
30	5.97	5.75	0.22
35	6.25	5.75	0.50
40	6.50	6.25	0.25
45	6.75	5.25	1.50
50	6.25	5.50	0.75

We performed a regression analysis of the Stud Farm results as well and they are shown in Table VIII. As these results show, the Stud Farm results are not statistically significant.

Last, we show now the results of our tests for Alarm in both tabular and graphical form in Tables IX and X, and Figure 9. As we referred to in the test development section, we were unable to completely run testing for all levels of inaccuracies for the Alarm network. This network is the largest of our test networks, and as we discovered in the development of the DQ algorithm, when met with inaccurate data the calculations become lengthy and the amount of virtual memory needed to conduct the tests was above that of our personal computing device. In the future it would be wise to conduct the testing on a larger machine, but also to do further testing to determine the computing needs of a network of this size. As this was not the goal of our current research

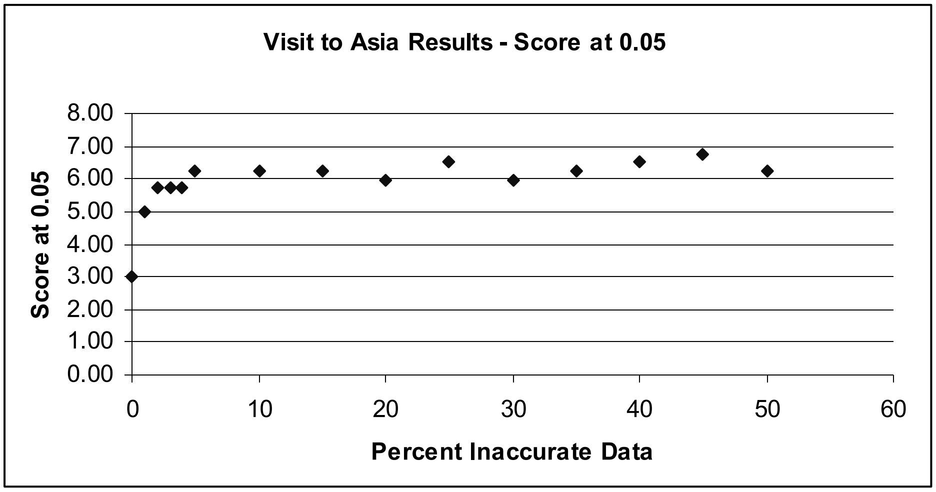


Fig. 7. Graphical representation of visit to Asia results for score at 0.05.

Table V. Regression Statistics for Score at 0.05, Visit to Asia

Regression Statistics	
Multiple R	0.568270568
R Square	0.322931439
Adjusted R Square	0.270849242
Standard Error	14.86029997
Observations	15

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	1369.229	1369.229	6.200419	0.027091497
Residual	13	2870.771	220.8285		
Total	14	4240			

we decided instead to test only up to 5% inaccurate data. Test results that did not complete are labeled N/A.

To be complete we performed a regression analysis on these results as well and they are in Table XI. While the significance f level is 0.042 and below our alpha of 0.05, there are not enough scores in our testing to reject our null hypothesis.

Our findings for Hypothesis II do not lead to a satisfactory metric for determining a difference between datasets of high accuracy and those of low accuracy. For completeness, we show here the graphs for each test as well as the regression analysis, but in summary there is not enough evidence to support rejecting our null hypothesis.

6. DISCUSSION OF RESULTS AND LIMITATIONS

The test results for the AAA, while not conclusive, have some promising elements. If we hold the significance level constant at 0.05, there is a noticeable difference in the average degree of nodes in the learned BNs. Considering only the Visit to Asia network results, there is a statistically significant correlation

Table VI. Accuracy Assessment Results, Stud Farm

Percent Inaccurate Data	**Score at 0.05**	**Score at 0.005**	**Score at 0.0005**	**Score at 0.00005**
0	3.33	2.50	2.36	2.36
1	11.00	11.00	10.67	10.12
2	11.00	11.00	11.00	10.67
3	11.00	11.00	11.00	11.00
4	11.00	11.00	11.00	11.00
5	11.00	11.00	11.00	11.00
10	11.00	11.00	11.00	11.00
15	11.00	11.00	11.00	11.00
20	11.00	11.00	11.00	11.00
25	11.00	11.00	11.00	11.00
30	11.00	11.00	11.00	11.00
35	11.00	11.00	11.00	11.00
40	11.00	11.00	11.00	11.00
45	11.00	11.00	11.00	11.00
50	11.00	11.00	11.00	11.00

Table VII. Continued Accuracy Assessment Results, Stud Farm

Percent Inaccurate Data	**Score at 0.05**	**Score at 0.00005**	**Score Diff. 0.05-0.0005**
0	3.33	2.36	0.97
1	11.00	10.12	0.88
2	11.00	10.67	0.33
3	11.00	11.00	0.00
4	11.00	11.00	0.00
5	11.00	11.00	0.00
10	11.00	11.00	0.00
15	11.00	11.00	0.00
20	11.00	11.00	0.00
25	11.00	11.00	0.00
30	11.00	11.00	0.00
35	11.00	11.00	0.00
40	11.00	11.00	0.00
45	11.00	11.00	0.00
50	11.00	11.00	0.00

between the percent of inaccurate data and the score at 0.05 (or average degree of the nodes at significance level 0.05). This significance does not hold true for the stud farm network because the score simply maxes out at only 1% inaccurate data. This may be an attribute of the genealogical nature of the network. In the Alarm network, there appears to be a relationship between the percent inaccurate data and score, particularly for results at a significance level of 0.00005; however, we do not have enough data to fully test these results. In all cases, in a practical situation, given two datasets, one of 100% accurate data and one of lower accuracy, it would be possible to determine the better of the two by examining the score at 0.05 learned by using the AAA. However, one would not be able to distinguish between a node of 5% inaccurate data versus a 50% inaccurate set. While the AAA did not provide a full solution to the problem, the authors are encouraged by the result that there is indeed some correlation between the two and that further refinement of the

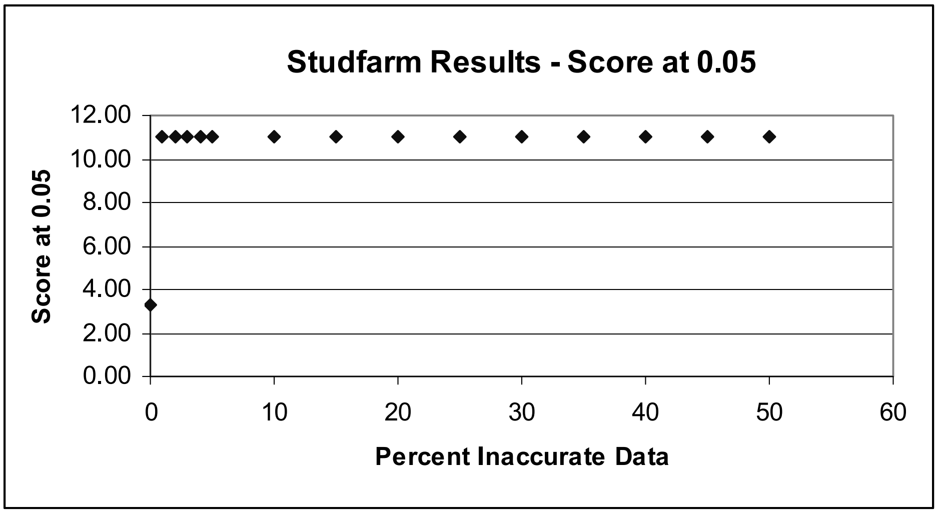


Fig. 8. Accuracy assessment results, score at 0.05 stud farm.

Table VIII. Regression Statistics for Score at 0.05, Stud Farm

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.30203153
R Square	0.09122305
Adjusted R Square	0.02131713
Standard Error	17.2163033
Observations	15

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	386.7857	386.7857	1.30494	0.273916489
Residual	13	3853.214	296.4011		
Total	14	4240			

Table IX. Accuracy Assessment Results, Alarm Network

Percent Inaccurate Data	**Score at 0.05**	**Score at 0.005**	**Score at 0.0005**	**Score at 0.00005**
0	4.60	4.49	4.44	4.44
1	7.90	6.60	5.79	5.30
2	10.60	7.46	7.46	6.00
3	11.03	10.33	8.27	7.79
4	N/A	9.63	9.63	8.71
5	N/A	N/A	N/A	9.73

scoring metric and the algorithm may lead to more conclusive and better practical results. As to the difference between score at 0.05 and score at 0.00005 metric, the results for testing indicates no statistically significant or practical usage for this metric.

The purpose of our testing for the AAA was to determine if there was any promise in these techniques. Therefore this initial research was limited in

Table X. Continued Accuracy Assessment Results, Alarm Network

Percent Inaccurate Data	Score at 0.05	Score at 0.00005	Score Diff. 0.05-0.0005
0	4.80	4.44	0.16
1	7.90	5.30	2.59
2	10.60	6.00	4.59
3	11.03	7.79	3.24
4	N/A	8.71	N/A
5	N/A	9.73	N/A

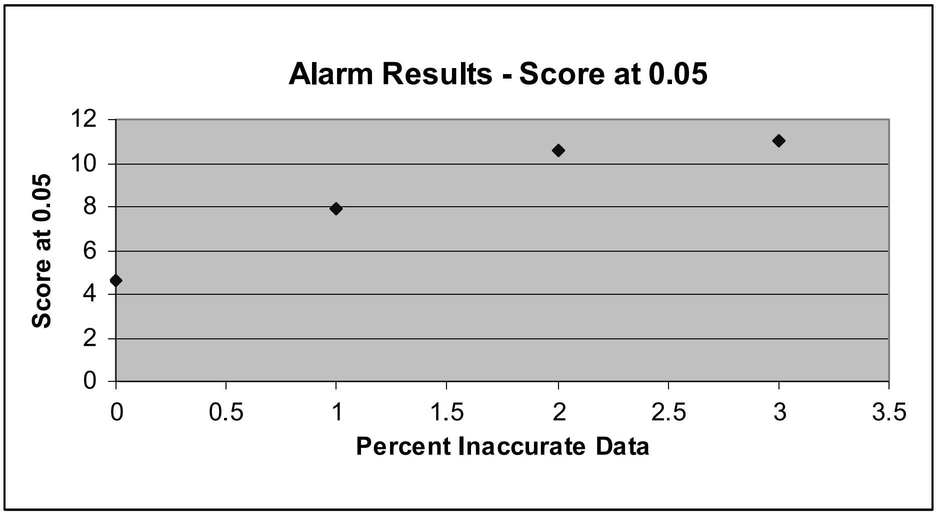


Fig. 9. Accuracy assessment results, score at 0.05 alarm network.

Table XI. Regression Statistics for Score at 0.05, Alarm

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.957415
R Square	0.916643
Adjusted R Square	0.874965
Standard Error	0.456499
Observations	4

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	4.583217	4.583217	21.9933005	0.042585044
Residual	2	0.416783	0.208392		
Total	3	5			

terms testing by the types of data tested, the types of networks used, and the type of scoring metric. If the AAA is to be fully tested, new inaccurate sets must be developed in varying ways (perhaps sets where only one or two nodes are corrupt). Also, while the variety of our chosen networks is advantageous, the AAA needs to be tested further with additional networks and where possible with datasets from practitioners in the field. Also, there may be better scoring metrics that can be tested instead of only the score at 0.05 (or average degree of the nodes). Finally, this testing was limited by the virtual memory and

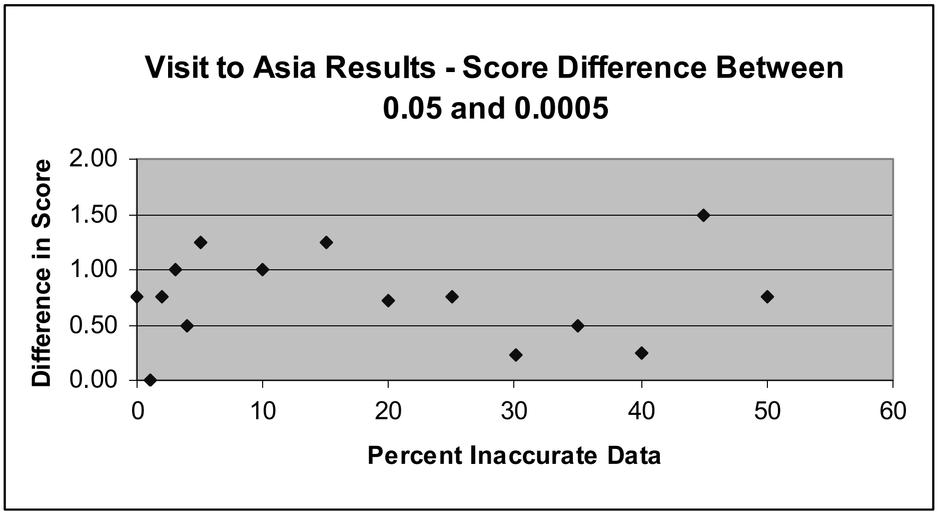


Fig. 10. Visit to Asia results score difference between 0.05 and 0.0005.

Table XII. Regression Statistics for Difference in Score 0.05 and 0.0005, Visit to Asia

SUMMARY OUTPUT					
Regression Statistics					
Multiple R	0.011286				
R Square	0.000127				
Adjusted R Square	-0.076786				
Standard Error	18.05858				
Observations	15				
ANOVA					
	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	0.540024	0.540024	0.0016559	0.968158738
Residual	13	4239.46	326.1123		
Total	14	4240			

computing power of the machine on which the testing was run. The testing for Alarm, a large network of 37 nodes, was unable to be carried out beyond 5% inaccurate data. Further testing in the future would be wise for determining the computing needs of a network of this size. While limited in its depth, the results and research into this algorithm do posit the concept that future research and algorithm development in this area would be fruitful and would lead to practical tools in the field.

7. FUTURE RESEARCH

For the authors, this represents the first of their undertakings into the development of an accuracy assessment tool that needs no prior knowledge of the dataset. We are concerned with investigating a range of artificial intelligence/knowledge discovery techniques for accuracy assessment. The next promising field in which we plan to research is in problem complexity as it relates to data mining. In recent research by Blake and Mangiameli [2008],

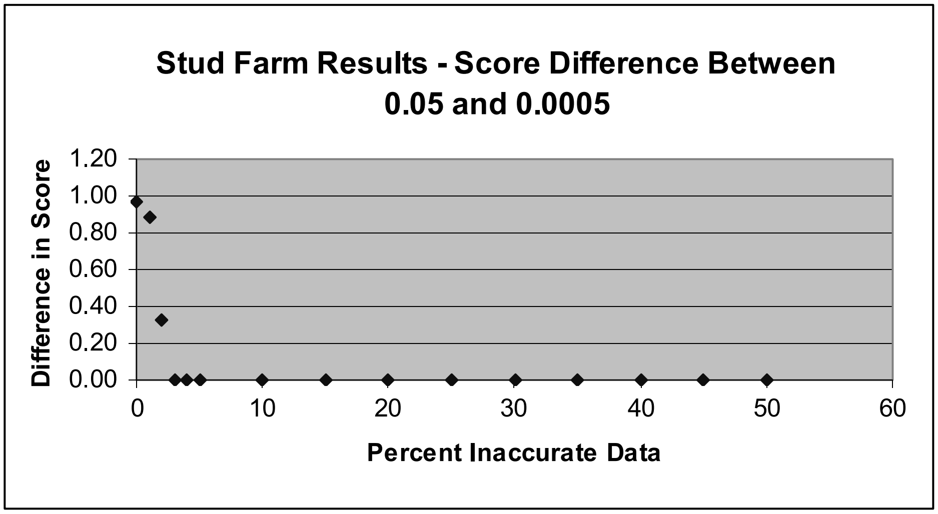


Fig. 11. Stud farm results score difference between 0.05 and 0.0005.

Table XIII. Regression Statistics for Difference in Score 0.05 and 0.0005, Stud Farm

SUMMARY OUTPUT					
Regression Statistics					
Multiple R	0.498841				
R Square	0.248842				
Adjusted R Square	0.191061				
Standard Error	15.65225				
Observations	15				
ANOVA					
	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	1055.09	1055.089	4.306611	0.05837411
Residual	13	3184.91	244.9931		
Total	14	4240			

a correlation was found between problem complexity in data mining and the quality of the data itself. We are hopeful that come useful metric can be found that can pinpoint inaccuracies in the data based on differences in problem complexity. The authors also wish to reinvestigate the literature in neural networks to determine if there may be some technique for performing assessments using these methods. Finally, if the AAA proves to be the most promising of these techniques, we will return to this work and develop new scoring metrics and an enhanced algorithm.

In addition to further research in these areas, one of the authors is involved complementary research in the development of objective and automatic freeware tools for assessing the quality of data, specifically in law enforcement datasets. This work incorporates an overall utility based calculation of total score. In future research and development efforts, this overall data quality score may then be fed into an algorithm such as the AAA. There are many

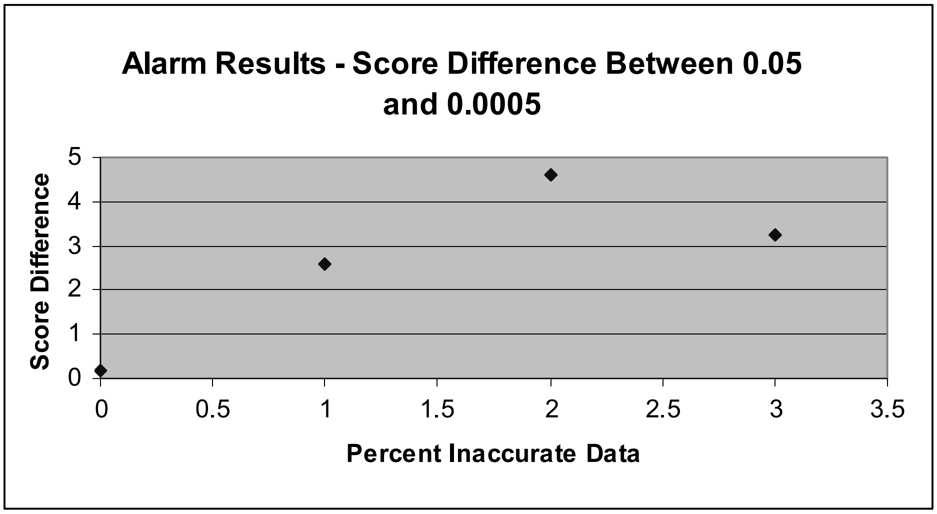


Fig. 12. Alarm results score difference between 0.05 and 0.0005.

Table XIV. Regression Statistics for Difference in Score 0.05 and 0.0005, Alarm

SUMMARY OUTPUT					
Regression Statistics					
Multiple R	0.782547				
R Square	0.61238				
Adjusted R Square	0.41857				
Standard Error	0.984404				
Observations	4				
ANOVA					
	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	3.061898	3.061898	3.159688	0.217453088
Residual	2	1.938102	0.969051		
Total	3	5			

unexplored avenues of research in this field and we are encouraged by the initial promising results presented here.

APPENDIX

Potential Tables: Visit to Asia

Visit to Asia? (A)

yes	0.01
no	0.99

Smokers? (S)

yes	0.5
no	0.5

Has tuberculosis (T)

A	yes	no
yes	0.5	0.01
no	0.95	0.99

Has lung cancer (L)

S	yes	no
yes	0.1	0.01
no	0.9	0.99

Has bronchitis (B)

S	yes	no
yes	0.6	0.3
no	0.4	0.7

Tuberculosis or cancer (E)

T	yes		no	
L	yes	no	yes	no
yes	1.0	1.0	1.0	0.0
no	0.0	0.0	0.0	1.0

Positive X-ray (X)

E	yes	no
yes	0.98	0.0.05
no	0.02	0.95

Dyspnoea (D)

T	yes		no	
L	yes	no	yes	no
yes	0.9	0.8	0.7	0.1
no	0.1	0.2	0.3	0.9

Potential Tables: Stud Farm

L

Carrier	0.01
Pure	0.99

Ann

Carrier	0.01
Pure	0.99

Brian

Carrier	0.01
Pure	0.99

Cicily

Carrier	0.1
Pure	0.99

K

Carrier	0.01
Pure	0.99

Fred

L	Carrier		Pure	
Ann	Carrier	Pure	Carrier	Pure
Carrier	0.666667	0.5	0.5	0.0
Pure	0.333333	0.5	0.5	1.0

Dorothy

Ann	Carrier		Pure	
Brian	Carrier	Pure	Carrier	Pure
Carrier	0.666667	0.5	0.5	0.0
Pure	0.333333	0.5	0.5	1.0

Eric

Brian	Carrier		Pure	
Cecily	Carrier	Pure	Carrier	Pure
Carrier	0.666667	0.5	0.5	0.0
Pure	0.333333	0.5	0.5	1.0

Gwenn

Ann	Carrier		Pure	
K	Carrier	Pure	Carrier	Pure
Carrier	0.666667	0.5	0.5	0.0
Pure	0.333333	0.5	0.5	1.0

Henry

Fred	Carrier		Pure	
Dorothy	Carrier	Pure	Carrier	Pure
Carrier	0.666667	0.5	0.5	0.0
Pure	0.333333	0.5	0.5	1.0

Irene

Eric	Carrier		Pure	
Gwenn	Carrier	Pure	Carrier	Pure
Carrier	0.666667	0.5	0.5	0.0
Pure	0.333333	0.5	0.5	1.0

John(John)

Henry	Carrier		Pure	
Irene	Carrier	Pure	Carrier	Pure
Sick	0.25	0.0	0.0	0.0
Carrier	0.5	0.5	0.5	0.0
Pure	0.25	0.5	0.5	1.0

Potential Tables: Alarm

HREKG(HREKG)

HR	Low		Normal0		High	
ErrCauter	True	False	True	False	True	False
Low	0.333333	0.98	0.333333	0.01	0.333333	0.01
Normal	0.333333	0.01	0.333333	0.98	0.333333	0.01
High	0.333333	0.01	0.333333	0.01	0.333333	0.98

HRBP(HRBP)

ErrLowOutput	True			False		
HR	Low	Normal	High	Low	Normal	High
Low	0.98	0.4	0.3	0.98	0.01	0.01
Normal	0.01	0.59	0.4	0.01	0.98	0.01
High	0.01	0.01	0.3	0.01	0.01	0.98

HRSat(HRSat)

HR	Low		Normal		High	
ErrCauter	True	False	True	False	True	False
Low	0.333333	0.98	0.333333	0.01	0.333333	0.01
Normal	0.333333	0.01	0.333333	0.98	0.333333	0.01
High	0.333333	0.01	0.333333	0.01	0.333333	0.98

Anaphylaxis(Anaphylaxis)

True	0.01
False	0.99

TPR(TPR)

Anaphylaxis	True	False
Low	0.98	0.3
Normal	0.01	0.4
High	0.01	0.3

ErrLowOutput(ErrLowOutput)

True	0.05
False	0.95

ErrCauter(ErrCauter)

True	0.1
False	0.9

HR(HR)

Catechol	Normal	High
Low	0.1	0.01
Normal	0.89	0.09
High	0.01	0.9

FiO2(FiO2)

Low	0.01
Normal	0.99

ArtCO2(ArtCO2)

VentAlv	Zero	Low	Normal	High
Low	0.01	0.01	0.04	0.9
Normal	0.01	0.01	0.92	0.09
High	0.98	0.98	0.04	0.01

ExpCO2(ExpCO2)

ArtCO2	Low				Normal				High	
Ventlung	Zero	Low	Normal	High	Zero	Low	Normal	High	Zero	Low
Zero	0.97	0.01	0.01	0.01	0.97	0.01	0.01	0.01	0.97	0.01
Low	0.01	0.97	0.97	0.97	0.01	0.01	0.01	0.01	0.01	0.01
Normal	0.01	0.01	0.01	0.01	0.97	0.97	0.97	0.97	0.01	0.01
High	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.97

ArtCO2	High	
VentLung	Normal	High
Zero	0.01	0.01
Low	0.01	0.01
Normal	0.01	0.01
High	0.97	0.97

MinVol(MinVol)

VentLung	Zero			Low			Normal			High
Intubation	Normal	Esophageal	OneSided	Normal	Esophageal	OneSided	Normal	Esophageal	OneSided	Normal
Zero	0.97	0.97	0.97	0.01	0.6	0.01	0.01	0.5	0.01	0.01
Low	0.01	0.01	0.01	0.97	0.38	0.97	0.01	0.48	0.01	0.01
Normal	0.01	0.01	0.01	0.01	0.01	0.01	0.97	0.01	0.97	0.01
High	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.97

VentLung	High	
Intubation	Esophageal	OneSided
Zero	0.05	0.01
Low	0.48	0.01
Normal	0.01	0.01
High	0.01	0.97

Press(Press)

KinkedTube	True									
Intubation	Normal				Esophageal				OneSided	
VentTube	Zero	Low	Normal	High	Zero	Low	Normal	High	Zero	Low
Zero	0.97	0.01	0.01	0.01	0.97	0.1	0.05	0.01	0.97	0.01
Low	0.01	0.49	0.01	0.01	0.01	0.84	0.25	0.15	0.01	0.29
Normal	0.01	0.3	0.08	0.01	0.01	0.05	0.25	0.25	0.01	0.3
High	0.01	0.2	0.9	0.97	0.01	0.01	0.45	0.59	0.01	0.4

KinkedTube	True		False							
Intubation	OneSided		Normal				Esophageal			
VentTube	Normal	High	Zero	Low	Normal	High	Zero	Low	Normal	High
Zero	0.01	0.01	0.97	0.01	0.01	0.01	0.97	0.4	0.2	0.2
Low	0.01	0.01	0.01	0.97	0.01	0.01	0.01	0.58	0.75	0.7
Normal	0.08	0.01	0.01	0.01	0.97	0.01	0.01	0.01	0.04	0.09
High	0.9	0.97	0.01	0.01	0.01	0.97	0.01	0.01	0.01	0.01

KinkedTube	False			
Intubation	OneSided			
VentTube	Zero	Low	Normal	High
Zero	0.97	0.01	0.01	0.01
Low	0.01	0.9	0.01	0.01
Normal	0.01	0.08	0.38	0.01
High	0.01	0.01	0.6	0.97

VentMach(VentMach)

MinVolSet	Low	Normal	High
Zero	0.01	0.01	0.01
Low	0.97	0.01	0.01
Normal	0.01	0.97	0.01
High	0.01	0.1	0.97

VentTube(VentTube)

VentMach	Zero		Low		Normal		High	
Disconnect	True	False	True	False	True	False	True	False
Zero	0.97	0.97	0.97	0.01	0.97	0.01	0.97	0.01
Low	0.01	0.01	0.01	0.97	0.01	0.01	0.01	0.01
Normal	0.01	0.01	0.01	0.01	0.01	0.97	0.01	0.01
High	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.97

VentLung(VentLung)

VentTube	Zero			Low			Normal			High
Intubation	Normal	Esophageal	OneSided	Normal	Esophageal	OneSided	Normal	Esophageal	OneSided	Normal
Zero	0.97	0.97	0.97	0.95	0.97	0.95	0.4	0.97	0.5	0.3
Low	0.01	0.01	0.01	0.03	0.01	0.03	0.58	0.01	0.48	0.68
Normal	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
High	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01

KinkedTube	True		False							
VentTube	High		Zero			Low			Normal	
Intubation	Esophageal	OneSided	Normal	Esophageal	OneSided	Normal	Esophageal	OneSided	Normal	Esophageal
Zero	0.97	0.3	0.97	0.97	0.97	0.01	0.97	0.01	0.01	0.97
Low	0.01	0.68	0.01	0.01	0.01	0.97	0.01	0.97	0.01	0.01
Normal	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.97	0.01
High	0.01	0.1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01

KinkedTube	False			
VentTube	Normal	High		
Intubation	OneSided	Normal	Esophageal	Onesided
Zero	0.01	0.01	0.97	0.01
Low	0.01	0.01	0.01	0.01
Normal	0.97	0.01	0.01	0.01
High	0.01	0.97	0.01	0.97

VentAlv(VentAlv)

Intubation	Normal				Esophageal				OneSided	
VentLung	Zero	Low	Normal	High	Zero	Low	Normal	High	Zero	Low
Zero	0.97	0.01	0.01	0.01	0.97	0.01	0.01	0.01	0.97	0.03
Low	0.01	0.97	0.01	0.01	0.01	0.97	0.01	0.01	0.01	0.95
Normal	0.01	0.01	0.97	0.01	0.01	0.01	0.97	0.01	0.01	0.01
High	0.01	0.1	0.01	0.97	0.01	0.01	0.01	0.97	0.01	0.01

Intubation	OneSided	
VentLung	Normal	High
Zero	0.01	0.01
Low	0.94	0.88
Normal	0.04	0.1
High	0.01	0.01

SaO2(SaO2)

Shunt	Normal			High		
PVSat	Low	Normal	High	Low	Normal	High
Low	0.98	0.01	0.01	0.98	0.98	0.69
Normal	0.01	0.98	0.01	0.01	0.01	0.3
High	0.01	0.01	0.98	0.01	0.01	0.01

Catechol(Catechol)

InsuffAnesth	True									
SaO2	Low									Normal
TPR	Low			Normal			High			Low
ArtCO2	Low	Normal	High	Low	Normal	High	Low	Normal	High	Low
Normal	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
High	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99

InsuffAnesth	True									
SaO2	Normal								High	
TPR	Low		Normal			High			Low	
ArtCO2	Normal	High	Low	Normal	High	Low	Normal	High	Low	Normal
Normal	0.01	0.01	0.01	0.01	0.01	0.05	0.05	0.01	0.01	0.01
High	0.99	0.99	0.99	0.99	0.99	0.95	0.95	0.99	0.99	0.99

InsuffAnesth	True							False		
SaO2	High							Low		
TPR	Low	Normal			High			Low		
ArtCO2	High	Low	Normal	High	Low	Normal	High	Low	Normal	High
Normal	0.01	0.05	0.05	0.01	0.05	0.05	0.01	0.05	0.05	0.01
High	0.99	0.95	0.95	0.99	0.95	0.95	0.99	0.95	0.95	0.99

InsuffAnesth	False									
SaO2	Low						Normal			
TPR	Normal			High			Low			Normal
ArtCO2	Low	Normal	High	Low	Normal	High	Low	Normal	High	Low
Normal	0.05	0.05	0.01	0.05	0.05	0.01	0.1	0.1	0.1	0.95
High	0.95	0.95	0.99	0.95	0.95	0.99	0.9	0.9	0.9	0.05

InsuffAnesth	False									
SaO2	Normal					High				
TPR	Normal		High			Low			Normal	
ArtCO2	Normal	High	Low	Normal	High	Low	Normal	High	Low	Normal
Normal	0.95	0.3	0.95	0.95	0.3	0.95	0.95	0.3	0.99	0.99
High	0.05	0.7	0.05	0.05	0.7	0.05	0.05	0.7	0.01	0.01

InsuffAnesth	False			
SaO2	High			
TPR	Normal	High		
ArtCO2	High	Low	Normal	High
Normal	0.99	0.95	0.99	0.3
High	0.01	0.05	0.01	0.7

CO(CO)

HR	Low			Normal			High		
StrokeVolume	Low	Normal	High	Low	Normal	High	Low	Normal	High
Low	0.98	0.95	0.3	0.95	0.04	0.01	0.8	0.01	0.01
Normal	0.01	0.04	0.69	0.04	0.95	0.3	0.19	0.04	0.01
High	0.01	0.01	0.01	0.01	0.01	0.69	0.01	0.95	0.98

PCWP(PCWP)

LVEDVolume	Low	Normal	High
Low	0.95	0.04	0.01
Normal	0.04	0.95	0.04
High	0.01	0.01	0.95

CVP(CVP)

LVEDVolume	Low	Normal	High
Low	0.95	0.04	0.01
Normal	0.04	0.95	0.29
High	0.01	0.01	0.7

LVEDVolume(LVEDVolume)

Hypovolemia	True		False	
LVFailure	True	False	True	False
Low	0.95	0.98	0.01	0.05
Normal	0.04	0.01	0.09	0.9
High	0.01	0.01	0.9	0.05

StrokeVolume(StrokeVolume)

LVFailure	True		False	
Hypovolemia	True	False	True	False
Low	0.98	0.5	0.95	0.05
Normal	0.01	0.49	0.04	0.9
High	0.01	0.01	0.01	0.05

PAP(PAP)

PulmEmbolus	True	False
Low	0.01	0.05
Normal	0.19	0.9
High	0.8	0.05

Shunt(Shunt)

PulmEmbolus	True			False		
Intubation	Normal	Esophageal	OneSided	Normal	Esophageal	OneSided
Normal	0.1	0.1	0.01	0.95	0.95	0.05
High	0.9	0.9	0.99	0.05	0.05	0.95

KinkedTube(KinkedTube)

True	0.04
False	0.96

Disconnect(Disconnect)

True	0.05
False	0.95

MinVolSet(MinVolSet)

Low	0.01
Normal	0.98
High	0.01

Intubation(Intubation)

Normal	0.92
Esophageal	0.03
OneSided	0.05

PulmEmbolus(PulmEmbolus)

True	0.01
False	0.99

InsuffAnesth(InsuffAnesth)

True	0.2
False	0.8

History(History)

LVFailure	True	False
True	0.9	0.01
False	0.1	0.99

LVFailure(LVFailure)

True	0.05
False	0.95

Hypovolemia(Hypovolemia)

True	0.2
False	0.8

REFERENCES

BALLOU, D. AND TAYI, G. 1999. Enhancing data quality in data warehouse in environments. *Comm. ACM*, 73–78.

BEINLICH, I., SUERMONT, H., CHOVEZ, R., AND G. COOPER, G. 1989. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the 2nd European Conference on Artificial Intelligence in Medical Care*. Springer, 247–256.

BLAKE, R. AND MANGIAMELI, P. 2008. The effects and interactions of data quality and problem complexity on data mining. In *Proceedings of the 13th International Conference on Information Quality*. 160–175.

COWELL, R. DAWID, G., LAURITZEN, S., AND SPIEGALHALTER, D. 1999. *Probabilistic Networks and Expert Systems*. Springer, New York.

CYKANA, P., STERN, M., AND PARUL, A. 1996. DoD guidelines on data quality management. In *Proceedings of the Conference on Information Quality*. 154–171.

ENGLISH, L. 2003. Plain english about information quality: Defining and measuring accuracy. DM Review. July 2003.

JENSEN, F. V. AND NIELSEN, T. 2007. *Bayesian Networks and Decision Graphs*. 2nd Ed. Springer, New York. Berlin, Germany.

JENSEN, F. V. 1995. *An Introduction to Bayesian Networks*. UCL Press and Springer, New York.

KAPLAN, D., KRISHNAN, R., PADMAN, R., AND PETERS, J. 1998. Assessing data quality in accounting information systems. *Comm. ACM*, 72–78.

LAUDON, K. 1986. Data quality and due process in large interorganizational record systems. *Comm. ACM*, 4–11.

LAURITZEN, S. L. AND SPIELGELHALTER, D. J. 1988. Local computation with probabilities in graphical structures and their applications to expert systems. *J. Royal Statist. Soc. B 50*, 2.

LEE, Y., STRONG, D., KAHN, B., AND WANG, R. 2002. AIMQ: A methodology for information quality assessment. *Inform. Manag.,* 133–146.

MADSEN, A. L., JENSEN, F., KJÆRULFF, U., AND LANG, M. 2005. The Hugin tool for probabilistic graphical models. *Int. J. Artif. Intell. Tools 14*, 3, 507–543.

MALETIC, J. AND MARCUS, A. 2000. Data cleansing: Beyond integrity analysis. In *Proceedings of the Conference on Information Quality*. 200–209.

NEAPOLITAN, R. E. 2004. *Learning Bayesian Networks*. Pearson Education, Upper Saddle River, NJ.

OLESEN, K., LAURITZEN, S., AND JENSEN, F. 1992. aHUGIN: A system creating adaptive causal probabilistic networks. In *Proceedings of the 8th Conference on Uncertainty in Artificial Intelligence*. 223–229.

PIPINO, L. AND KOPSCO, D. 2004. Data mining, dirty data, and costs. In *Proceedings of the 9th International Conference on Information Quality*. 164–169.

SESSIONS, V. AND VALTORTA, M. 2006. The effects of data quality on machine learning algorithms. In *Proceedings of the 11th International Conference on Information Quality*. 485–498.

SPIRTES, P., GLYMOUR, C., AND SCHEINES, R. 2000. *Causation, Prediction, and Search*. MIT Press, Cambridge, MA.

SUNG, S., LI, Z., AND SU, P. 2002. A fast filtering scheme for large database cleansing. 76–83.

VOMLEL, J. 2004. Thoughts on belief and model revision with uncertain evidence. In *Proceedings of the Conference Znalosti*. 126–137.

WAND, Y. AND WANG, R. 1996. Anchoring data quality dimensions in ontological foundations. *Comm. ACM 39*, 86–95.

Received August 2007; revised June 2009; accepted August 2009