Ontologies for Data Mining and Knowledge Discovery to Support Diagnostic Maturation

Timothy J. Wilmering
The Boeing Company
P.O. Box 516 M/C S106 3075
St. Louis, MO. 63166
timothy.j.wilmering@boeing.com

John W. Sheppard
The Johns Hopkins University
3400 N Charles Street
Baltimore, MD 21218
jsheppa2@jhu.edu

Abstract

Initial test and maintenance solutions that are deployed to support new complex systems are generally imperfect and are initially liable to contribute substantially to system ownership costs. This is because development of effective health management solutions requires prediction of complex systemic interactions and the effect of presupposed external stimuli. It is nearly always the case that unforeseen emergent behavior (those that result from unpredicted system interactions) of fielded systems within their operational context create deviations from anticipated health management system performance. This suggests a need for processes and tools to monitor the effectiveness of product health management solutions in their application domains, collect data that validates and documents system performance, and pinpoint and analyze relevant patterns that can help mitigate the issues that arise. The process of identifying and implementing corrective actions as required to satisfy customer support requirements is known as diagnostic maturation and often draws on techniques from data mining and knowledge discovery.

Most current approaches to data mining involve analyzing low-level data elements to attempt to induce previously unknown relationships among those data elements. Techniques such as clustering, association rule learning, feature extraction, and classification pervade the data mining literature. Unfortunately, most data mining implementations are either “blind” in that they consider all available data, or they depend on a human expert to identify the types of relationships of interest. Recent developments in defining ontologies for a domain provide significant potential in data mining in general and diagnostic maturation in particular. In this paper, we discuss a framework and approach for utilizing health management ontologies to guide the maturation process and better streamline automatic (or semi-automatic) knowledge discovery to improve diagnostics.

Diagnostic Maturation Process

Initial test and maintenance solutions that are deployed to support new complex systems are generally imperfect and are initially liable to contribute substantially to system ownership costs. This is because development of effective health management solutions requires prediction of complex systemic interactions and the effect of presupposed external stimuli. It is nearly always the case that unforeseen emergent behavior (those that result from unpredicted system interactions) of fielded systems within their operational context create deviations from anticipated health management system performance. This suggests a need for processes and tools to monitor the effectiveness of product health management solutions in their application domains, collect data that validates and documents system performance, and pinpoint and analyze relevant patterns that can help mitigate the issues that arise. The ability to mature the effectiveness of fielded system test, diagnostic, and maintenance procedures is a critical factor in an overall system support posture. The process of identifying and implementing corrective actions as required to satisfy customer support requirements is known as diagnostic maturation.

Recognizing that data mining in general and diagnostic maturation in particular are difficult, the purpose of this paper is to discuss one approach to streamlining the process by using domain ontologies. Specifically, in this paper, we discuss a framework and approach (one of many possible) for utilizing health management ontologies to guide the maturation process and better streamline automatic (or semi-automatic) knowledge discovery to improve diagnostics. The focus is on using the ontology to focus data mining and reduce the size of the search space to only those portions directly relevant to a specific maturation question.

The initial frameworks to support health management system maturation are put in place in the conceptual design stage and should be developed throughout the system life cycle. Model-based approaches (e.g., logic models, dependency models, qualitative models, and physics-based

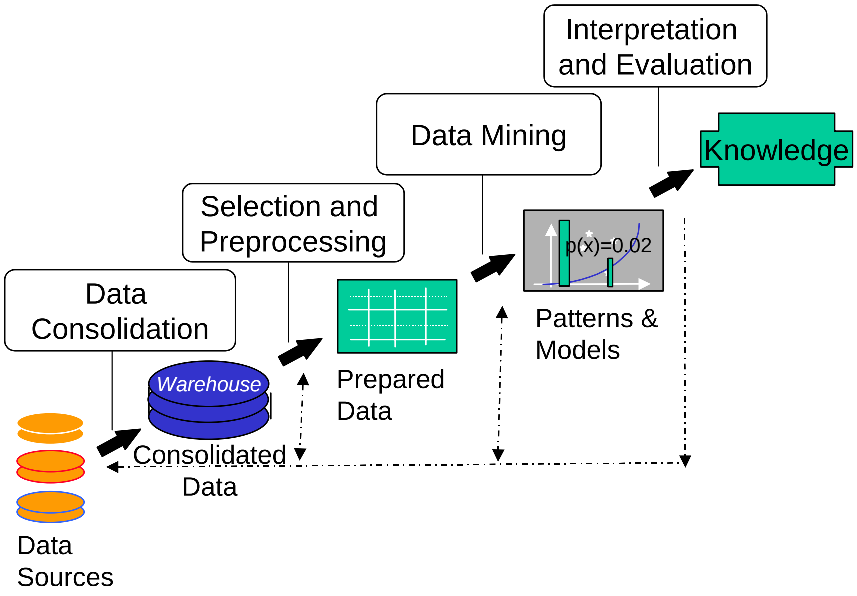

Figure 1. Traditional Knowledge Discovery Process (Han and Kamber, 2006)

models) provide excellent representational tools for developing and documenting system design choices that support traceability of design decisions and can be refactored during the corrective action analysis and development process. When unexpected and unplanned system level design interactions, operational and environmental stresses, or other influences create a system readiness issue or cost of ownership problem, the models will be in place to support the remedial actions that must be taken. This represents an iterative closed loop process of root cause analysis, corrective action deployment and reevaluation called a Maturation Cycle, or more formally in some circles, the FRACAS (Failure Reporting and Corrective Action System) process. In either case, the goal is to determine a corrective action that prevents or minimizes recurrence of the reported problem in subsequent use of the product (Wilmering, 2001).

Maturation is hard because of two fundamental issues (Wilmering, 2003):

1. While it is generally perceived that data collection is a prerequisite for the maturation process, data collection is in fact a difficult issue. The product data that is typically required for maturation analysis is generally stored in disparate heterogeneous data systems - this makes access, retrieval, and integration of the requisite information a costly and often incomplete process at best.
2. There are several categories of analysis required to support maturation: correlation analysis (identifying the problem and correlating the data which supports and characterizes that identification), root cause analysis, and analysis of the support system that provides a framework for the processes in question and corrective action. At issue is the fact that the data required to perform these analyses is scattered among many different data sources and systems – pulling it all together can be challenging, to say the least.

Knowledge Discovery

We propose that formal Knowledge Discovery techniques may offer significant benefit to the diagnostic maturation process. Suppose a Maturation Cycle is triggered by some event or series of events during the operation of a system. Perhaps a repeated test failure is determined to be unfounded – this may first be noticed by repeated False Alarms or Cannot Duplicates involving a system component. An analyst may first characterize the problem based on the information as it is initially presented: what is the part whose failure is misdiagnosed, what information is already available that helps to characterize the problem (times, locations, maintenance scenarios, etc). Once the initial problem statement is formalized, the Knowledge Discovery (KD) process is described in Han and Kamber (2006) can begin as follows (Figure 1):

Data collection and consolidation: Identify relevant data and factors, find relevant data sources, locate data in sources and formulate queries.

Prepare Data (Data Integration): Develop data correlation keys, retrieve data, integrate data, clean data, examine data.

Data Mining: Data mining is the heart of the KD process. Select the appropriate methods for pattern extraction from the large data sets collected in the previous step, then apply any of numerous classification and pattern matching techniques to extract relevant relationships from the data sets.

Interpretation and Evaluation (Analysis of results): Interpretation of results may employ further data reduction mechanisms, use of visualization techniques, or other methods to enhance presentation and interpretation of the data for human consumption.

This may be an iterative process as patterns emerge, facts are discovered, or experiments are designed and carried out.

Knowledge discovery is hard for two fundamental reasons:

1. There is a wide variety of available tools, techniques, and representations for data that must be considered throughout the knowledge discovery process.
2. Given the typically large amount of data that must be mined, the process of finding interesting relationships in a combinatorially large data space is itself computationally hard.

Given the range of possible tools and techniques available, as well as the range of possible combinations of data elements to be considered for analysis, the problem of knowledge discovery in general (and data mining in particular) is difficult. The focus in this paper is providing an approach to reduce the scope of the search required in knowledge discovery by using ontologies. Currently, the common approaches to data mining include (among others).

- On Line Analytical Processing (OLAP)—A multi-dimensional analysis technique involving the aggregation of "cubes" (actually, hypercubes) of data to determine relationships among the data. The aggregation process is analogous to marginalization in probability theory, except counts are returned rather than probabilities.
- Statistical Analysis—A collection of tools range from correlation analysis, to trending, to regression where models of data are constructed uses traditional tools from statistics. In fact, all of the techniques used in data mining can (and often do) benefit from applying combinations of statistical tools.
- Cluster Analysis—An unsupervised learning technique where the data is grouped by common attributes, often based on a distance function. Clustering can be used to identify a set of candidate classes for the data that might not have been known previously.
- Association Rule Analysis—An unsupervised learning technique where correlations between data elements are examined in an attempt to extract logical relationships between those attributes. Learning association rules is similar to learning decision rules using supervised learning techniques such as FOIL (Quinlan, 1990).
- Pattern Classification—A class of algorithms, often considered under supervised learning, where data with associated classes or labels are used to derive a compact model of those classes for future classification of new data.

As we see, determining which of these tools to apply can, itself, be a significant challenge. Interesting work at NYU (Bernstein, Hill, and Provost, 2002) is applying an ontology of the data mining process itself to guide a miner to select approach tools and consider appropriate data elements.

A second, possibly more significant challenge facing the data mining process arises from the so-called "curse of dimensionality." In fact, this is the challenge we attempt to address here. While studying control processes, Richard Bellman (the inventor of dynamic programming) noted that, with the exponential increase in volume of a sample space as the number of attributes (or dimensions) increases, we are faced with an accompanying requirement to have exponentially increasing data to characterize the higher dimensional space (Bellman, 1961). In traditional data mining problems, the data exists (i.e., there are large numbers of data records, even in a high-dimensional space), but the challenge comes in analyzing this huge amount of data to extract interesting and useful knowledge.

Mediated Data Collection and Integration

The diagnostic maturation process requires ready access to design, maintenance, and other logistic support information sources. Data essential to analysis of maturation issues may be generated by system producers (e.g., engineering, product support organizations, etc), or by system users (maintenance and supply chain data, etc). Each of these organizations is also multifaceted in nature. Engineering organizations, for example, are composed of sub-disciplines such as design, reliability and maintainability, etc. Compounding the problem is the fact that the data of interest resides in multiple systems each with different owners where it does exist – and it should be recognized that some data that is desirable to have might not be captured in data systems at all. The heterogeneous nature of these sources present issues having to do with access, accuracy, semantic understanding, completeness, and correlation of relevant design features with performance data, and spans hardware platforms, Data Base Management Systems (DBMS), and software standards. More specific data structural concerns may include general data representation issues (e.g., methodologies, hierarchical dominance across systems, uniqueness identification, data types) and more specific data format disparities (e.g., units of measure, code sets, "intelligent" values), and semantic differences in the relationship between data labels (terms) and their intended meaning (concepts). This heterogeneity occurs quite naturally as multiple systems are developed and evolved as a function of independent decisions and design within the development lifecycle of a system. The physical constraints having to do with access are easing, but generally accepted approaches to content integration are only recently appearing in actual applications.

Federated data servers, or query engines, can address many of these heterogeneity issues. The goal of a federated data server is to provide real-time, integrated access to diverse and distributed data as if it were a single source, regardless of format, location, or operating environment (Hayes and Mattos, 2003). Federated servers address differences in DBMS, physical data structure, structured vs. semi- or unstructured data, data type casting, and query performance across systems. These sort of mechanical or structural issues are essential to integration of the information required for maturation analysis, but they do not address the conceptual issues involving semantic heterogeneity.

Mediated approaches to semantic integration can help by enumerating archetypes of the concepts of interest in the domain of interest – in our case health management and related maintenance and logistic information – then detailing the relationships and constraints between these concepts in a unified ontology – thereby reducing information requests to operations on a single model of the requisite information and mapping those requests between the ontology and logical models of the data sources which can instantiate the information requests.

An inherent advantage of a mediated implementation is in the knowledge that can be synthesized from the models and mappings. An ontology is developed to explicitly represent the semantics of the target information domain, then mappings are created to correlate source data with the terminology and concept relations in the ontology, providing a basic semantic interpretation of the data being accessed and its relation and relevancy to the domain of interest.

The domain ontology subsumes the semantics of the data across multiple data systems but is otherwise independent of the system data representations. From end users' perspectives, a mediated information integration system looks like a single information resource, containing all of the available information within a specific domain.

Ontology-Directed Diagnostic Maturation

We proceed from the assumption that the diagnostic maturation process is fundamentally a data mining/knowledge discovery process. Specifically, we claim that there are three major types of maturation steps to be performed relative to system diagnosis, all with the goals of either improving the accuracy of diagnosis or improving the efficiency of the diagnostic process.

Refining cost estimates of actions or tests being performed in the maintenance process. These cost estimates are used to optimize the maintenance process and, if inaccurate, can substantially weaken the benefits expected from a strong optimization algorithm.

Refining the probabilities of occurrence of various maintenance events. Again, these probabilities are essential to the optimization process since we are attempting to minimize expected cost over the system being maintained. Inaccurate probabilities can skew the process in ways that can yield significant, sub-optimal maintenance procedures.

Correcting errors in an underlying system model. In fact, many types of errors can exist in these types of models; however, most diagnostic models tend to focus on capturing relationships between observable information (i.e., tests) and the diagnoses to be drawn (i.e., faults). If we restrict ourselves to these types of errors, then our concern becomes determining if there are relationships that are either missing from the model or if relationships need to be added to the model that currently do not exist. A variation of these two extremes is the case where relationships have qualifying information, and that qualifying information is not completely accurate.

Historically, refining cost or probability estimates has been relatively straightforward, so we focus our discussion in this paper on improving the accuracy of the system diagnostic models.

Ontologies for Diagnostic Maturation

For the following discussion, we draw on the development of a standard ontology for diagnostic maturation being developed by the Institute of Electrical and Electronics Engineers (IEEE), recognizing that alternative ontologies are possible. The approach we describe in the following should work, in principle, with any of these alternative ontologies.

Currently, the IEEE Standards Coordinating Committee 20 (SCC20) of the IEEE Standards Association is developing two families of standards that together define an ontology of the diagnostics problem domain. IEEE 1232 Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE) defines an ontology specifically covering the diagnostic process itself (IEEE, 2007a). Given the fact the intent is to improve diagnostics, we need the diagnostic ontology as a framework; however, we also need an ontology for the maintenance data collection process so we can mediate these processes to mature the diagnostics. To support this, IEEE P1636 Software Interface for Maintenance Information Collection and Analysis (SIMICA) is being developed (IEEE, 2007b). Currently, SIMICA consists of two "component" standards—IEEE P1636.1 Test Results and IEEE P1636.2 Maintenance Action Information. The Test Results standard provides ontological information about the test process and provides a framework for capturing specific measurements and outcomes of actual tests (IEEE, 2006). The Maintenance Action Information standard provides ontological information about the maintenance process, given supporting components for test and diagnosis (IEEE 2007c). The combination of the three—test, diagnosis, and maintenance—provide an integrated, mediated view of information necessary for process improvement and maturation.

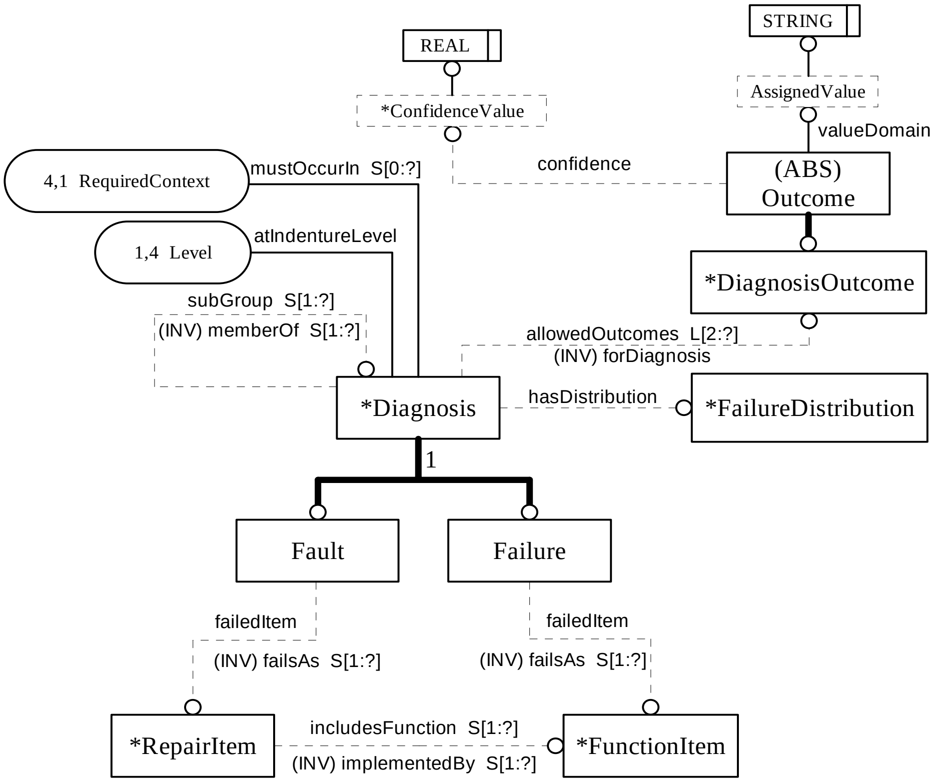

Figure 2. AI-ESTATE Ontology Excerpt: Diagnosis

As an example of how the ontology can guide the data mining process, we will consider the problem of correcting the relationships in a diagnostic model. Currently, AI-ESTATE defines three different types of diagnostic models—the fault tree, the diagnostic inference model, and the Bayes model. For our purposes, we will focus on the Bayes model. Second, AI-ESTATE includes a Dynamic Context Model (DCM), which was designed to provide an information interface to a diagnostic reasoner and represent historical information captured during an actual diagnostic process. All of the entities defined in the DCM are tied back to the third type of model in AI-ESTATE. This model—the Common Element Model (CEM)—provides a top-level ontology for diagnostic information to characterize relationships and constraints between different elements in the diagnostic domain. In fact, the specific diagnostic models are also tied back to the CEM.

Figure 2 shows a small portion of the AI-ESTATE CEM ontology corresponding to a diagnosis. Of interest here is the fact that diagnosis is defined to have a lattice organization where a particular diagnosis can have multiple children and multiple parents. The parent-child relationship is defined here to represent a grouping of diagnoses into, for example, replaceable unit groups or common functional groups. The diagnosis entity also has two attributes of interest to us—mustOccurIn (which is of type RepairContext) and atIndentureLevel (which is of type Level). A similar structure appears for the Test, Action, and Repair entities. In fact, Test and Repair are represented as subtypes of Action.

Remember that our concern is managing the curse of dimensionality. In this case, we can restrict our analysis to a set of diagnoses that are at the same level of indenture, part of the same repair context, and at the same depth within the lattice structure by constructing ontological queries against the mediated information, thus restricting the information to be limited to the subset of interest. Alternatively, we may find that there is a group of diagnoses at the same level of indenture and in the same repair context but appearing at different depths in the lattice structure. We may decide to restrict search to the subset of those elements having a common ancestor in the structure. One final example arises by noting that there are additional entities within the ontology that have attributes of type Level and RepairContext. These common attributes can be used to consider relationships among entities other than tests or diagnoses but appearing within a common context.

In the following, we will assume we have received a data set to be mined corresponding to data collected using

```
function Apriori($D$, $s$)
  $L_1 \leftarrow$ set of large 1-itemsets
  $k \leftarrow 2$
    while $L_{k-1}$ not empty do
      $C_k \leftarrow select(L_{k-1})$ // candidate itemset
      for all $d$ in $D$ do
        $C_d \leftarrow subset(C_k, d)$ // candidate contained in data
        for all $c$ in $C_t$ do
          $count[c] \leftarrow count[c] + 1$
      $L_k \leftarrow \{c \in C_k \mid count[c] \geq s\}$ // test for minimal support
      $k \leftarrow k + 1$
  return $\cup_k L_k$
```

Figure 3. *Apriori* Algorithm for Finding Large Itemsets (Agrawal and Srikant, 1994)

P1636.1 Test Results coupled with the P1232 Dynamic Context Model (for diagnostic history) and P1636.2 Maintenance Action Information (for actual repair information). Using our ontology, we restrict analysis to the set of test results, diagnoses, and repairs occurring at a particular level of indenture with a common repair context that have been verified through the repair certification process.

Guided Association Rule Mining

Given the now restricted set of data to consider, we start by deriving association rules from the data (Agrawal and Srikant, 1994). An association rule is a rule written as in implication of the form $X \Rightarrow Y$ where X is the conjunction of a set of variables and Y is the conjunction of a different set of variables. Given an association rule and a training data set, we say that the *support s* of the rule is the percentage of examples in the training set for which the conjunction $X \wedge Y$ holds, and the *confidence c* of the rule is the percentage of those examples for which X holds where $X \wedge Y$ also holds (i.e., $|X \wedge Y| / |X|$). Note that support can be interpreted as "coverage" (i.e., the number of examples that match all of the variables in the rule) and confidence can be interpreted as "accuracy" (i.e., the percentage of examples for which the $X \Rightarrow Y$ holds).

Agrawal and Srikant provide an algorithm, called *Apriori*, with an efficient variant, called *AprioriHybrid* for finding association rules in a data set. Both algorithms require that minimum support and confidence thresholds be set by the user. The algorithm finds so-called "large itemsets" which are sets of variables for which the minimum support has been obtained. We refer to a large itemset with k variables to be a k-itemset. A basic version of the *Apriori* algorithm can be found in Figure 3. In this algorithm, let L_k be the set of large k-itemsets, C_k be the set of candidate k-itemsets, d is some subset of "literals" or "items" in the data (called transactions in the original), and s be the user-specified minimal percent of transactions containing some candidate itemset c. This algorithm then finds the itemsets of various sizes meeting the minimum support threshold. Once the itemsets are found, it is a simple matter to construct the rules where the left hand sides are itemsets that are subsets of itemsets covering the entire rule.

The *AprioriHybrid* algorithm implements the *Apriori* algorithm more efficiently. Specifically, by using breadth-first search and a hash tree, the itemsets exceeding the minimum support threshold (and subsequent association rules satisfying the minimum confidence threshold) can be found in time linear in the number of examples. Given the *AprioriHybrid* algorithm, diagnostic maturation proceeds by applying the algorithm to the data set that has been restricted according to our ontology. Association rules can be constructed relating tests to one another, tests and diagnoses, or even tests/diagnoses with other factors in the maintenance process (such as a repair action). While the general *AprioriHybrid* algorithm permits construction of rules of the form $P_1 \wedge \ldots P_n \Rightarrow Q_1 \wedge \ldots \wedge Q_m$, we will restrict the rules of interest to use to those containing only one consequent. These association rules can then be ranked by their confidence and used to refine the diagnostic model as long as the addition of such rules improve the accuracy of diagnosis.

Augmenting Bayesian Classifiers for Diagnosis

Consider a specific case where we have a Bayesian diagnostic model as defined in (IEEE, 2007a). Previous work by Sheppard *et al.* has demonstrated that naïve Bayesian classifiers can provide a relatively simple method for capturing diagnostic relationships that also yields relatively accurate diagnostics (Sheppard, Butcher, Kaufman and MacDougall, 2006). However, naïve Bayes models are only capable of representing linearly separable concepts, and even then, only a subset of all linearly separable concepts can be represented (Zhang, Ling, and Zhao, 2004). It is also interesting to note that the popular *D*-matrix of model-based diagnosis suffers from the same problem (Sheppard and Butcher, 2007).

Formally, the naïve Bayes network finds the class label (i.e., diagnosis) that maximizes the *a posteriori* probability of the specific class given the set of observations (i.e. tests):

$$D = \arg\max_{d_i \in \mathbf{D}} P(d_i \mid o(T_1),\ldots,o(T_n))$$
$$= \arg\max_{d_i \in \mathbf{D}} P(o(T_1),\ldots,o(T_n) \mid d_i)P(d_i).$$

where d_i is the ith diagnosis, T_j is the jth test, and $o(T_j)$ is the observed outcome for T_j. But we have a problem that the joint distribution over the tests is exponential in the number of tests. The naïve Bayes assumption states that we can treat each of these observations as if they are conditionally independent given the diagnosis, and this leads to the classification rule:

$$D = \arg\max_{d_i \in \mathbf{D}} P(d_i)\prod_{j=1}^{n} P(o(T_j) \mid d_i).$$

Given a set of training data mapping test results (outcomes) to actual faults repaired, we can "learn" the naïve Bayes network by observing that $P(d_i)$ is simply the frequency of occurrence of a particular fault in the data set and similarly, $P(o(T_j) \mid d_i)$ is the frequency of test outcome $o(T_j)$ considering only the particular diagnosis d_i.

We proceed under the assumption the initial diagnostic model that has been deployed is a naïve Bayes network. This is reasonable for an initial deployment given empirical evidence of frequent effectiveness of such a model; however, the theoretical limitations indicate that the model can be improved as data is collected. One of the primary methods for improving naïve Bayes networks is by "augmenting" the networks with additional conditional dependence relationship that had previously been assumed away.

One popular approach augmenting naïve Bayes networks is through the tree-augmented naïve Bayes (TAN) algorithm (Friedman, Geiger, and Goldszmidt, 1997) Empirical evidence has shown that the improvements from TAN can be substantial; however, the experiments in (Sheppard, J., Butcher, S., Kaufman, M., and MacDougall C. 2006) suggest that such improvements may, on average, be marginal. We hypothesize that one reason for such marginal improvement is the fact that the augmentation algorithm can still miss important dependencies while including lesser dependencies that can actually deceive the classifier. This hypothesis is being explored in separate work.

Proceeding from the above hypothesis, we further hypothesize that using association rules can provide a better indication of needed augmentations for the network. The task then becomes, for a particular association rule, how do we incorporate that rule into the current diagnostic model? Specifically, for an association rule of the form $P_1 \wedge \ldots P_n \Rightarrow Q$, we include (or update) the probability table corresponding to $P(Q \mid P_1,\ldots,P_n)$. There are three cases to consider.

Case 1—A new association rule relating tests and diagnoses: This case is the simplest of the three since it simply revises an existing conditional probability table in the current network. Specifically, if we have a rule $D \Rightarrow T$, we modify the conditional probabilities for $P(o(T) \mid D)$ according to the data in the training set. Note that, if we have the opposite rule of $T \Rightarrow D$, then we can consider the contrapositive and model $\neg D \Rightarrow \neg T$, which still leads to a simple update of the current conditional probability table.

Suppose we have a more complex rule of the type $D_i \wedge D_j \Rightarrow T$. This, in fact, corresponds to a multiple fault, which can now be added to the model by including the conditional probability table for $P(o(T) \mid D_i, D_j)$. This constitutes an augmentation to the network.

Case 2—A new association rule relating multiple tests: In this case, any rule identified, whether $T_i \Rightarrow T_j$ or the more general $T_1 \wedge \ldots \wedge T_m \Rightarrow T_j$ results in an augmentation of the network because we now must add a new conditional probability table to the network. Note that this can be done directly by following the template of $P(T_j \mid T_1,\ldots,T_m)$. Note, however, that the actual probability tables would correspond to $P(T_j \mid T_1,\ldots,T_m,D_k)$ for all diagnoses D_k because the information needs to be merged with the existing model.

Case 3—A new association rule relating a test or diagnosis to some other variable: Finally, this case is interesting because it not only augments the network with an additional conditional probability table, but it incorporates an additional random variable into the network. The new random variable corresponds to the factor not currently captured by the set of tests or diagnoses, such as an in-process repair action. The first step, then, is to add the new variable to the network and then add the associated probability table.

In considering the above approach, there are several computational issues to be considered. First, as augmentations are included in the network, the size of the corresponding conditional probability tables grows exponentially. This is one of the reasons for selecting the minimum support and confidence values carefully. Second, even controlling the number of augmentations, the computational expense of processing augmenting networks also grows exponentially. Therefore, utilizing resulting networks may require alternative processing approaches such as conversion to join trees, Monte Carlo simulation, or some alternative form of network transformation. Third, collecting further historical information may determine that a previously learned association rule does not, in reality, apply. At such time, the rule and associated portions of the probability tables affected should be removed.

Future Directions

The presented algorithm is an initial approach to maturing diagnostic models based on an associated ontology for the test, diagnostic, and maintenance process. This approach is work in progress, and the first step is to run several experiments to validate the approach. It is reasonable to expect the model to perform relatively well given that it proceeds from an existing model and only modifies the model based on statistical information collected in the field. The key advantage to the approach is that it provides a directed strategy for using the ontology to guide the maturation process. Additional future work would include extending the idea to other parts of the maintenance process and other parts of the ontology.

In addition to augmented Bayes learning, there are many machine learning methods available for creating and revising classification models. These methods apply directly to the problem of maturing diagnostic models, which are also classification models. Therefore, a rich area of related research is to use the ontology to direct learning and revision of models such as decision trees (P1232 fault tree model), rule sets (P1232 diagnostic logic model) or companion models such as neural networks, support vector machines, or even hidden Markov models (for system prognosis).

Another interesting question is whether the approach can be used from an alternative perspective where the ontology directs search in areas where no relationships in the ontology itself exist. The purpose here would be to determine if there are correlations from the data indicating a previously unknown relationship, thus permitted maturation of the ontology itself. Specifically, the KD process might identify a useful relationship that was not explicitly modeled in the original ontology.

Finally, one of the authors is exploring the utility of constructing separate system-level diagnostic models corresponding to specific or subsets of a fleet of systems given empirical evidence to show that such models can yield more accurate diagnosis than a single model covering all instances of a given system. Based on contextual information (which is included in the ontologies), work is ongoing to utilize this contextual information to better determine what historical data to apply to mature or develop a given model.

Conclusion

The purpose of this paper was to present a framework for utilizing ontologies combined with machine learning to mature diagnostic models. The approach focuses on using the ontologies to restrict data sets in a meaningful way to manage the curse of dimensionality and still yield useful model revisions. While the research is still too early to report experimental results, theoretical analysis suggests the approach has promise.

References

Agrawal, R. and Srikant, R. 1994. Fast Algorithms for Mining Association Rules. *Proceedings of the 20th VLDB Conference*, Santiago, Chile.

Bellman, R. 1961. *Adaptive Control Processes: A Guided Tour*, Princeton, NJ: Princeton University Press.

Bernstein, A., Hill, S., and Provost, F. 2002. "Intelligent Assistance for the Data Mining Process: An Ontology-Based Approach," CeDAR Working Paper IS-02-02, Center for Digital Economy Research, Stern School of Business, New York University,.

Friedman, N., Geiger, D., and Goldszmidt, M. 1997. Bayesian Network Classifiers. *Machine Learning*, 29:131–163.

Han, J. and Kamber, M. 2006. *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers.

Hayes, H. and Mattos N. 2003. Information On Demand. *IBM DB2 Magazine*, Quarter 3, Vol. 8, Issue 3.

IEEE 2007a. P1232 *IEEE Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE)*, Draft 1.0, Piscataway, NJ: IEEE Standards Press.

IEEE 2007b. P1636 *IEEE Standard Software Interface for Maintenance Information Collection and Analysis (SIMICA)*, Draft 1.0, Piscataway, NJ: IEEE Standards Press.

IEEE 2006. P1636.1, *IEEE Trial Use Standard Software Interface for Maintenance Information Collection and Analysis (SIMICA): Exchanging Test Results and Session Information via the Extensible Markup Language (XML)*, Draft 1.0, Piscataway, NJ: IEEE Standards Press.

IEEE 2007c. P1636.2, *IEEE Trial Use Standard Software Interface for Maintenance Information Collection and Analysis (SIMICA): Exchanging Maintenance Action Information via the Extensible Markup Language (XML)*, Draft 1.0, Piscataway, NJ: IEEE Standards Press, 2007.

Quinlan, J. R. 1990. "Learning Logical Definitions from Relations," *Machine Learning*, 5:239–266.

Sheppard, J., Butcher, S., Kaufman, M., and MacDougall C. 2006. Not-So-Naïve Bayesian Networks and Unique Identification in Developing Advanced Diagnostics. *Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana, March.

Sheppard, J. and Butcher, S. 2007. A Formal Analysis of Fault Diagnosis with D-Matrices, to appear in *Journal of Electronic Testing: Theory and Applications*, Springer.

Wilmering, T. 2001. Semantic Requirements on Information Integration for Diagnostic Maturation. *IEEE AUTOTESTCON Conference Record*.

Wilmering, T. 2003. When Good Diagnostics Go Bad – Why Maturation is Still Hard. *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT.

Zhang, H., Ling C., and Zhao Z. 2004. The Learnability of Naïve Bayes. *Advances in Artificial Intelligence: 13th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, Quebec, Canada, Springer, LCNS Vol 1822, pp. 432–441.