Web-based Knowledge Acquisition to Impute Missing Values for Classification

Na Tang
Computer Science Dept.
University of California, Davis
Davis, CA 95616
natang@ucdavis.edu

V. Rao Vemuri
Computer Science Dept.
University of California, Davis
Davis, CA 95616
rvemuri@ucdavis.edu

Abstract

Machine learning is the science of building predictors from data while accounting for the predictor's accuracy on future data. Many machine learning classifiers can make accurate predictions when the data is complete. In the presence of insufficient data, statistical methods can be applied to fill in a few missing items. But these methods rely only on the available data to calculate the missing values and perform poorly if the percentage of missing values exceeds a threshold. An alternative is to fill in the missing data by an automated knowledge discovery process via mining the WWW. This novel procedure is applied by first restoring missing information and next learning the parameters of the classifier from the restored data. Using a Bayesian network as a classifier, the parameters, i.e., the probabilities associated with the causal relationships in the network, are deduced using the knowledge mined from the WWW in conjunction with the data available on hand. The method, when tested with heart disease data sets from the UC Irvine Machine Learning Repository [2], gave satisfactory results.

1. Introduction

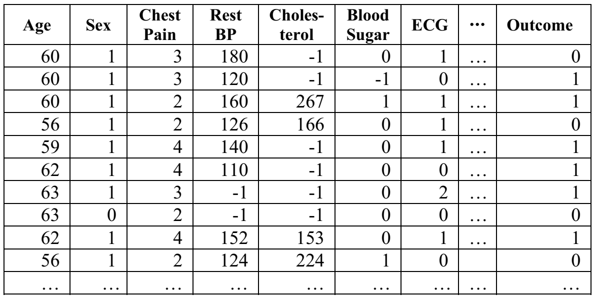
Machine learning methods are popularly used in many knowledge-based systems and applications to build predictors. Many of them classify data into a number of predefined categories and make predictions. For example, classifiers can be used to diagnose a disease from symptoms. Here, the training data for the classifier is a set of symptoms (which can be present or absent) and diseases, while the parameters to be estimated are the relationships among them. Once the parameters are known, the diseases can be diagnosed given a new patient's symptoms.

One widely-used classifier is the Bayesian network [15], a probabilistic model. However, training Bayesian networks to make accurate predictions can be difficult or even impossible when there is insufficient information. Today's WWW is a source of huge amount of information, and it can be used to fill the gaps. However, people have to manually get the information and fill in the data because information on the web is understandable only to humans. Existing knowledge discovery techniques can help retrieve the needed information automatically and effectively.

In this paper, a methodology is described to automatically retrieve the relevant knowledge from the web and restore the missing data. Bayesian networks built from the restored data are then used to make predictions. More advanced information extraction techniques are used instead of handcrafted rules used in an earlier paper [18] to make effective and efficient extractions (see detail at section 3.2). As a result, the web serves as an automated information resource to build models for systems and applications. The basic assumption here is that the knowledge obtained from the web can correctly reflect the relationship among data items. This assumption is validated in our experiments - at least for the problem domain considered.

The heart-disease data sets from the UCI Machine Learning Repository [2] are used as the initial experimental data to validate the approach. Figure 1 shows a segment from the data sets (All "-1"s are missing values). There are fourteen attributes including the class variable *Outcome* that indicates whether a person has heart disease or not. All the other attributes describe the conditions and symptoms of the patient such as *Age*, *Gender*, *Cholesterol*, etc.

A Bayesian model can be built from complete training data sets and can then be used to make a diagnosis, given one's conditions. Methods of building Bayesian networks from complete data, such as Naive Bayes (NB) and Tree Augmented Naive Bayes (TAN), are discussed in [8, 9]. A NB model assumes that attributes are independent from each other. Figure 2 shows the topology of a NB model, explicitly showing the independence of the factors causing heart disease. A TAN model has more complex topology, which is obtained by searching through a restricted space of possible network structures. Both NB and TAN can make satisfactory predictions in the presence of complete data.



Age	Sex	Chest Pain	Rest BP	Choles-terol	Blood Sugar	ECG	…	Outcome
60	1	3	180	-1	0	1	…	0
60	1	3	120	-1	-1	0	…	1
60	1	2	160	267	1	1	…	1
56	1	2	126	166	0	1	…	0
59	1	4	140	-1	0	1	…	1
62	1	4	110	-1	0	0	…	1
63	1	3	-1	-1	0	2	…	1
63	0	2	-1	-1	0	0	…	0
62	1	4	152	153	0	1	…	1
56	1	2	124	224	1	0		0
…	…	…	…	…	…	…	…	…

Figure 1. "Va" Data Set (one of the four heart disease data sets) from UCI repository.

However, in many practical circumstances, data entries of some attributes may be partially or completely missing. An attribute whose values are completely missing is called an *incomplete attribute*. *Incomplete attributes* can arise, (in the heart disease problem), for example, when some recently acquired new knowledge points to the possibility of a newly discovered cause to a disease condition.

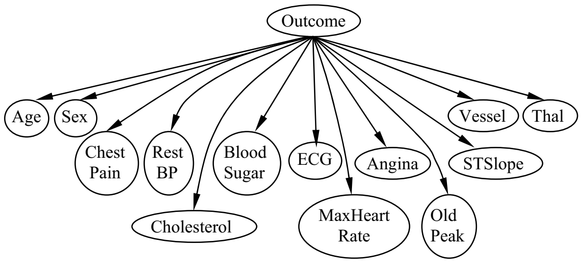


Figure 2. NB model for heart-disease data

The rest of paper is organized as follows. We review some related work in Section 2. In Section 3, we describe our approach to fill in missing data for constructing Bayesian classifiers. Experimental results are reported in Section 4. We conclude and discuss our future work in Section 5.

2. Related work

Several statistical methods are available to construct Bayesian networks from incomplete data. Some of them are described below:

(1) A simple method is to fill in missing data values using available data as a guide [17]. A missing entry can be filled in with the most probable value of the *incomplete attribute* or with the most probable value in the corresponding predicting class. This method is not feasible when the values of an attribute are completely missing.

(2) Expectation-maximization (EM) algorithm [4] estimates the parameters by iteratively finding the expectation of a parameter and then finding the maximum likelihood estimate (MLE) using the parameter from the expectation step. It provides more reliable estimates of the parameters than (1) in the presence of missing data. However, it relies on the assumption that data items are missing at random (MAR): That is, with each configuration, the available data is a representative sample of the complete data, which is not true in our case because sometimes the available data can not reflect the relationship between the *incomplete attribute* (whose values are totally missing) and the values of the other known attributes.

(3) Other statistical methods seek to simultaneously fill in the missing data while searching for an optimal structure of the Bayesian network model. Two of these methods are briefly outlined here. Structural EM [7, 6] starts with an initial structure and passes the structure to the EM algorithm. The MLE returned by EM is considered as the score for the structure. A new structure is generated by adding, deleting or reversing an edge in the previous structure and the new structure is passed to the EM algorithm again and the score for the new structure is returned and compared to the previous score. The process is repeated until there is no improvement in the score. The Evolutionary Algorithm (EA) [14], uses a genetic algorithm to evolve both network structures and missing values to find an optimal Bayesian network.

Another limitation of these methods is that they lead to not-so-accurate predictions when a large percentage of data is missing or when data is non-randomly missing. For example, EA finds good predictive networks at 5%, 10% and 15% missing data while the predictive accuracy degrades sharply at 30% missing data. To address these limitations, a novel approach is proposed in this paper to deduce the missing information via a knowledge discovery process in conjunction with mining the WWW.

Some of the other papers in web mining research are closer in concept to the work presented in this paper. Reference [3] shows a methodology for extracting useful information to build a knowledge base. This method, WWW→KB, retrieves relevant documents from certain web sites via Naive Bayes text classifier, then extracts useful symbolic information from the retrieved documents via an information extraction technique called SRV (Sequential Rules with Validation), and finally builds the knowledge base from the extracted data. In our paper this idea is extended to WWW→KB→BN in the sense that the knowledge extracted from the web is used to help construct Bayesian networks (BN) from incomplete data. Also, the method differs from the WWW→KB method in three important respects. First, the method described here uses search engines to search a large portion of the WWW whereas the WWW→KB method performs content mining within a specified context. Two, the proposed method extracts probabilistic information to fill in missing values rather than symbolic information. Three, the proposed method performs further processing to handle the uncertainty associated with the extracted information.

Web query systems [10, 11] are also related to our mining approach. They analyze the queries, use search engines to search the web, and employ information retrieval and extraction techniques to get the answers to user queries. However, they focus on the query categorization. Furthermore, most of the sought-after information is very simple and directly resides in the text, while the knowledge needed in this paper is implicitly indicated by the text and requires further processing to be converted to probabilities.

3. Restoring missing data via knowledge discovery from the WWW

The crux of the idea here is to look for patterns of relationship between the *incomplete attribute* and other available attributes of the problem. It is assumed that the sought-after relationship information appears in some web document either as a natural language sentence or as an item in a table. In either case the relationship indicates how the available attributes influence the *incomplete attribute* or how the *incomplete attribute* influences the other attributes. The implementation architecture is shown in Figure 3. For simplicity, three-way relationships and cascade relationships are not examined in this paper; only binary relationships (i.e., relationships between two attributes at a time) are considered to make the classification step (Step C1) and the extraction process (Step C2) simple to handle.

The three steps C1, C2 and C3 are explained in the following subsections. It is assumed that *Cholesterol* and *RestBP* are two *incomplete attributes* in the heart data table. *Cholesterol* is the attribute indicating people's cholesterol values and *RestBP* is the attribute indicating people's resting blood pressure values.

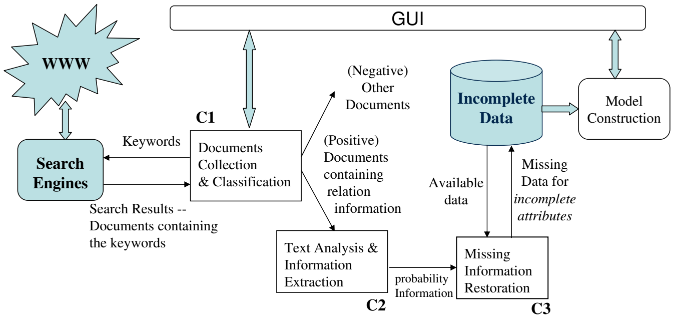

GUI
WWW
C1
Keywords
Search Engines
Documents Collection & Classification
Search Results -- Documents containing the keywords
(Negative) Other Documents
(Positive) Documents containing relation information
Incomplete Data
Model Construction
Available data
Missing Data for *incomplete attributes*
Text Analysis & Information Extraction
probability Information
Missing Information Restoration
C2
C3

Figure 3. Implementation architecture

3.1. C1: Documents collection and classification

Document Collection Phase: N (= 300 or more, typically) documents are collected from Google using a compound search phrase comprised of one *incomplete attribute* and one completely specified attribute. For example, if *Cholesterol* is an *incomplete attribute*, the probability information between *Cholesterol* and other attributes would be of interest. To extract documents about *Cholesterol* and *Outcome*, we use the keyword set: "cholesterol heart disease" (Similarly, the keyword set "cholesterol age" is used to get documents containing relationship information between *Cholesterol* and *Age* and so on.) Google is chosen because it is a widely available general-purpose search engine. A specialized search engine and a specialized database, such as Medline, may work better for a specialized domain, say, the health-care field.

Document Classification Phase: A trained Naive Bayes Classifier (Rainbow toolkit [12]) is used to divide the collected documents into two classes: the positive class containing information on the causes of heart-disease and associated probability data and the negative class comprising of all other documents. As the documents in the positive class (positive documents) contain the needed probability information, they are retained for further processing; the negative documents are discarded. Naive Bayes text classifier [13] is a probabilistic classifier that calculates the probability of a document belonging to each possible class and then assigns the document the class with the highest probability.

The Naive Bayes text classifier needs to be trained before the preceding step can be implemented. This training is performed manually: Using a number of professional web sites that specialize in heart disease [1], 200 documents are collected, manually inspected and hand-labeled as positive and negative (50 for the positive class and 150 for the negative class). Naive Bayes text classifier is used here because of its high accuracy and low complexity. It outperforms k-Nearest Neighbors (k-NN) in our categorization task.

3.2. C2: Text analysis and information extraction

Only text-formed files on the web, e.g., HTML (HyperText Markup Language) files, are examined here because they are one of the most popular formats on the WWW. The text classifier and the extraction system described in this paper can only deal with such format. The embedded knowledge may reside in free text as well as structured text (e.g. tables). For the latter, some HTML tags become very useful. Rules are automatically generated via WHISK [16], a well-known information extraction system , to process and extract the probability information. The length condition for text segments (e.g., the length of the table, the length of the sentence) is added into each rule for more effective extractions. Each sentence and each table from the positive documents is analyzed and examined to see if they matched any of the rules. The outputs of the probability information are collected for processing step C3.

In an earlier paper [18], rules were designed manually from user observations in this step, which took more manual effort than by using WHISK because users need to design the rules again once the problem domain changes. Also, manually designed rules do not extract information as accurately as the rules learned by WHISK. WHISK is more suited for the extraction task here than other information extraction methods because it is designed to handle all types of text, ranging from structured text to free text. Also, WHISK can extract multiple slots and the information targeted for extraction usually consists of multiple slots.

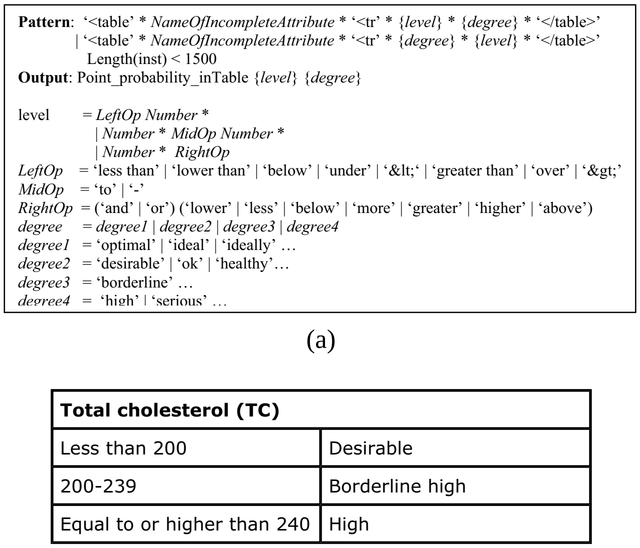
WHISK is a supervised learning method. A set of training instances, e.g., a set of sentences and tables, are annotated by users. WHISK starts with a set of untagged instances and an empty training set. At each iteration a set of untagged instances are selected and presented to users to annotate. Users add tags to the instance that contains the expected information; otherwise the instance becomes a negative instance. An instance can contain multiple tags. Then WHISK selects an instance-tag pair to grow a new rule. The rule is generated via a top-down induction algorithm. First the most general rule that covers the current instance is found and the terms in the instance are then added into the rule one at a time until there is no extraction error or the metric, the Laplacian expected error, reaches the minimum. The Laplacian expected error is defined as $\frac{e+1}{n+1}$, where n is the number of extractions and e is the number of incorrect extractions. When a rule is done, it is added to the rule set. A rule that causes extraction errors on other instances is removed from the rule set. This process is repeated until the rules can cover all the positive instances.

The probability information targeted for extraction, i.e. the relations between the *incomplete attribute* and other attributes, is divided into two categories: *point probabilities* and *qualitative influences*. Other forms of probability information would also be useful but are not considered in this paper. Formal definitions of all these items are given in [5]. Only the above two categories are discussed because of their simplicity.

3.2.1 *Point probabilities*

Point probabilities can be expressed in the mathematical form $P(a_i|a_{j1},...a_{jk}) = c$, where c is a constant. As mentioned before, to simplify the extraction task, we only examine $P(a_i|a_j)$, i.e. the probabilistic relationship between two attributes instead of many attributes. Take the relation between *Cholesterol* and *Outcome* as an example. The degree of risk for heart disease for different levels of cholesterol is usually explicitly described in the relevant documents.

In the documents examined, the *point probabilities* are typically expressed in tables (structured text) or in regular sentences (free text). Examples of the former are given below and the latter is omitted because of the similarity.



```
Pattern:  '<table' * NameOfIncompleteAttribute * '<tr' * {level} * {degree} * '</table>'
          | '<table' * NameOfIncompleteAttribute * '<tr' * {degree} * {level} * '</table>'
            Length(inst) < 1500
Output: Point_probability_inTable {level} {degree}

level      = LeftOp Number *
             | Number * MidOp Number *
             | Number *  RightOp
LeftOp     = 'less than' | 'lower than' | 'below' | 'under' | '&lt;' | 'greater than' | 'over' | '&gt;'
MidOp      = 'to' | '-'
RightOp    = ('and' | 'or') ('lower' | 'less' | 'below' | 'more' | 'greater' | 'higher' | 'above')
degree     = degree1 | degree2 | degree3 | degree4
degree1    = 'optimal' | 'ideal' | 'ideally' …
degree2    = 'desirable' | 'ok' | 'healthy'…
degree3    = 'borderline' …
degree4    =  'high' | 'serious' …
```

(a)

Total cholesterol (TC)	
Less than 200	Desirable
200-239	Borderline high
Equal to or higher than 240	High

(b)

Figure 4. Extraction rule for *point probabilities* **expressed in tables: (a) Extraction Rule; (b) An instance covered by the rule.**

Figure 4 (a) shows the a rule generated by WHISK to extract the point probabilities from tables. Here, "*" stands for any string and "|" for "or". The italic words in the rule are semantic classes. Some of them are hard coded in WHISK and others defined by the user. *NameOfIncompleteAttribute* = 'total cholesterol' here, where *Cholesterol* is the *incomplete attribute*. Figure 4 (b) shows an instance that covered by this rule, which is a table appearing in the web browser.

Three tags are extracted from the table: ('Less than 200', 'Desirable'), ('200-239', 'Borderline high') and ('higher than 240', 'High'). Each tag has two slots: *level* and *degree*. All (*level*, *degree*) pairs are interpreted in probability format: $P(Outcome = 1|Cholesterol < 200)$ is low, $P(Outcome = 1|200 < Cholesterol < 239)$ is borderline high and $P(Outcome = 1|Cholesterol > 240)$ is very high. The words ('desirable', 'borderline' ...) that describe the probability are finally converted into numerical values.

3.2.2 *Qualitative Influences*

Qualitative influences describe how one attribute influences another in a qualitative way. It consists of positive and negative influences. A positive influence from attribute A_i to A_j means that choosing a higher value for A_i makes a higher value for A_j more likely. A negative influence is defined in a similar way.

The extraction rules for the *qualitative influences* are obtained in a similar way as those for the *point probabilities*. Each tag contains two slots *changeOf*A_i and *changeOfIncompleteAttribute*. If both of the two changes are in the same direction, there is a *positive influence* from attribute A_i to the *incomplete attribute*; otherwise, a negative influence exists. An example is given below:

> "As people get older, their cholesterol levels rise."
> "Cholesterol levels naturally rise as men and women age."

Here, A_i is *Age* and the *incomplete attribute* is *Cholesterol*. The tags extracted from the two sentences are ('older', 'rise') and ('age', 'rise') respectively. Each tag contains two slots: *changeOfAge* and *changeOfCholestrol*. Both of the changes are increasing, which confirms a positive *qualitative influences* from *Age* to *Cholesterol*.

If the probability information extracted from different documents is inconsistent, it is assumed that the probability information published in most web sites is the most reliable information. For example, if most web sites show that it is optimal to have cholesterol less than 200mg/dl while a few web sites regard 160mg/dl as the separation line, then the former is chosen. Also, the obtained information comes from the top of the list returned by the search engine. It is implicitly assumed that the higher-ranked search results are more reliable.

3.3. C3: Missing information restoration

There are two types of outputs from Step C2: *point probabilities* and *qualitative influences*. For example, the *point probabilities* for the relation between *Outcome* and *Cholesterol* can be represented by:

$$P(\text{Outcome} = 1|\text{Cholesterol} = v_1) = v_2, \qquad (1)$$

where v_1 stands for a range of cholesterol levels and v_2 is a constant. The second type of output from Step C2 includes a positive influence from *Age* to *Cholesterol* and a positive influence from *Gender* to *Cholesterol*. The former can be represented by:

$$\begin{aligned} &P(\text{Cholesterol} > v|\text{Age} = a_1) > \\ &\qquad P(\text{Cholesterol} > v|\text{Age} = a_2), \end{aligned} \qquad (2)$$

given $a_1 > a_2$. The latter can be interpreted similarly. Given all these probability outputs from Step C2, and given the probability constrains such as P(Age, $Gender$, $Outcome$) = v calculated from the available data, P($Cholesterol$ | Age, $Gender$, $Outcome$) can be elicited based on the approach described in [5]. This method converts all the probability information into a linear system of equalities and inequalities, from which bounds on the probabilities of interest are calculated. From these bounds, it is possible to elicit the required probabilities, namely, $P(Cholesterol|Age, Gender, Outcome)$. Now the missing values in the data set can be filled in based on these probabilities. For example,

$$\begin{aligned} &P(Cholesterol < 200|pred) = v_1, \\ &P(200 < Cholesterol < 240|pred) = v_2, \\ &P(Cholesterol > 240|pred) = v_3, \\ &pred = \text{“}Age < 50, Gender = female, Outcome = 1\text{”}. \end{aligned}$$

If a female patient is younger than 50 and has heart disease, then her missing cholesterol number is set to one of the values from the set $\{< 200, 200 - 240, > 240\}$ with probabilities $\frac{v_1}{v}$, $\frac{v_2}{v}$, and $\frac{v_3}{v}$, respectively, where $v = v_1 + v_2 + v_3$.

4. Experimental results

We conducted our experiments on the Cleveland data set, which is a complete data set, i.e. a table with no missing values. We randomly chose a portion of the data for training and the remaining for testing. First, we trained our Naive Bayes model and TAN model using the complete training data set and obtained the heart-disease prediction accuracies with the testing data. Then we assumed that all the *Cholesterol* values or *RestBP* values or both in the training data were missing and applied our method to fill in these values. Figure 5 shows the fill-in accuracies with different training data size for *Cholesterol* and *RestBP*. It shows that the knowledge extracted from the web is capable of restoring incomplete data. Also, the fill-in accuracy increases as the training data size gets bigger.

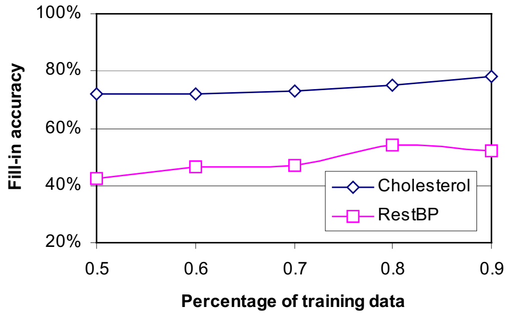


Figure 5. Fill-in accuracies

Based on the training data with the filled-in values, we built our NB and TAN models and evaluate the trained networks with the testing data. We also tested the prediction accuracies without considering the *incomplete attributes*. To compare our approach with the statistical methods, we applied EM algorithm to estimate the NB parameters by considering the missing attributes as hidden variables to test the prediction accuracies. All the heart-disease prediction accuracies are shown in Figure 6 with missing *Cholesterol*, missing *RestBP*, or both, respectively.

The results show that the incomplete data (data with *incomplete attributes*) can deteriorate the NB and TAN models, i.e. it leads them to lower prediction accuracies than the original complete data. Generally the incomplete data has greater influence on the TAN model than on the NB model. An increment in the amount of missing data results in a decrease in the prediction accuracy. The EM algorithm can only improve the prediction accuracy to a small degree. The filled-in data set via our approach outperforms the EM algorithm and performs almost as good as or sometimes even better than the original data. It works better with the TAN model than with the NB model. Also, the percentage of missing data does not have an influence on the performance of the filled-in data via our approach. This is because the missing data is filled in based on the extracted knowledge but not on the available data only. Therefore, it can be a promising method when a large percentage of data is missing.

5. Summary and discussion

In our study, the WWW was used as the source for gathering useful information to restore incomplete relational tables. We mined the web using search engines, selected the relevant documents via Naive Bayes text classifier, extracted the useful knowledge via WHISK, and finally filled in the missing information via uncertainty processing. The filled-in tables were, in turn, used to simulate two Bayesian network models of the unknown system under study.

The method is proposed to solve many realistic problems, but several interesting questions need to be addressed before it can be effectively used. The most obvious questions relate to the computational complexity, reliability and scalability of the method. The text classifier used in Step C1 is a supervised learning method and thus takes some manual effort of labeling training documents. An alternative would be unsupervised document categorization (i.e., document clustering). Currently work is in progress to investigate the potential of existing clustering techniques. Reliability depends on the confidence one can place on the filled-in numbers generated by this method. That is, any possible confliction between the extracted information and the available data need to be further examined. Scalability refers to the range of missing data. We are particularly interested in the case when a large percentage of data is missing, i.e., when the problem domain is data poor.

Acknowledgements

Work reported in this paper is supported in part by the AFOSR grant FA9559- -4-1-0159.

References

[1] http://www.heartinfo.org, http://www.americanheart.org, http://www.heartcenteronline.com, http://www.heartsavers.org, http://www.healthandage.com, http://www.nhlbi.nih.gov, http://heartdisease.about.com.

[2] C. Blake and C. Merz. UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/mlrepository.html, 1998.

[3] M. Craven, D. DiPasquo, D. Freitag, A. K. McCallum, T. M. Mitchell, K. Nigam, and S. Slattery. Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence*, 118(1/2):69–113, 2000.

[4] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statistical Society*, pages 1–38, 1977.

[5] M. J. Druzdzel and L. C. van der Gaag. Elicitation of probabilities for belief networks: Combining qualitative and quantitative information. *Eleventh Annual Conference on Uncertainty in Artificial Intelligence*, 22:141–148, 1995.

[6] N. Friedman. Learning belief networks in the presence of missing values and hidden variables. *Fourteenth International Conference on Machine Learning*.

[7] N. Friedman. The bayesian structural em algorithm. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 647–654, 1998.

[8] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, 1997.

[9] D. Heckerman, D. Geiger, and D. M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

[10] C. C. T. Kwok, O. Etzioni, and D. S. Weld. Scaling question answering to the web. pages 150–161, 2001.

[11] S. K. S. Lam and M. T. Özsu. Querying web data - the webqa approach. *Third International Conference on Web Information Systems Engineering*, pages 139–148, 2002.

[12] McCallum and A. Kachites. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www-2.cs.cmu.edu/ mccallum/bow/rainbow/, 1996.

[13] T. M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

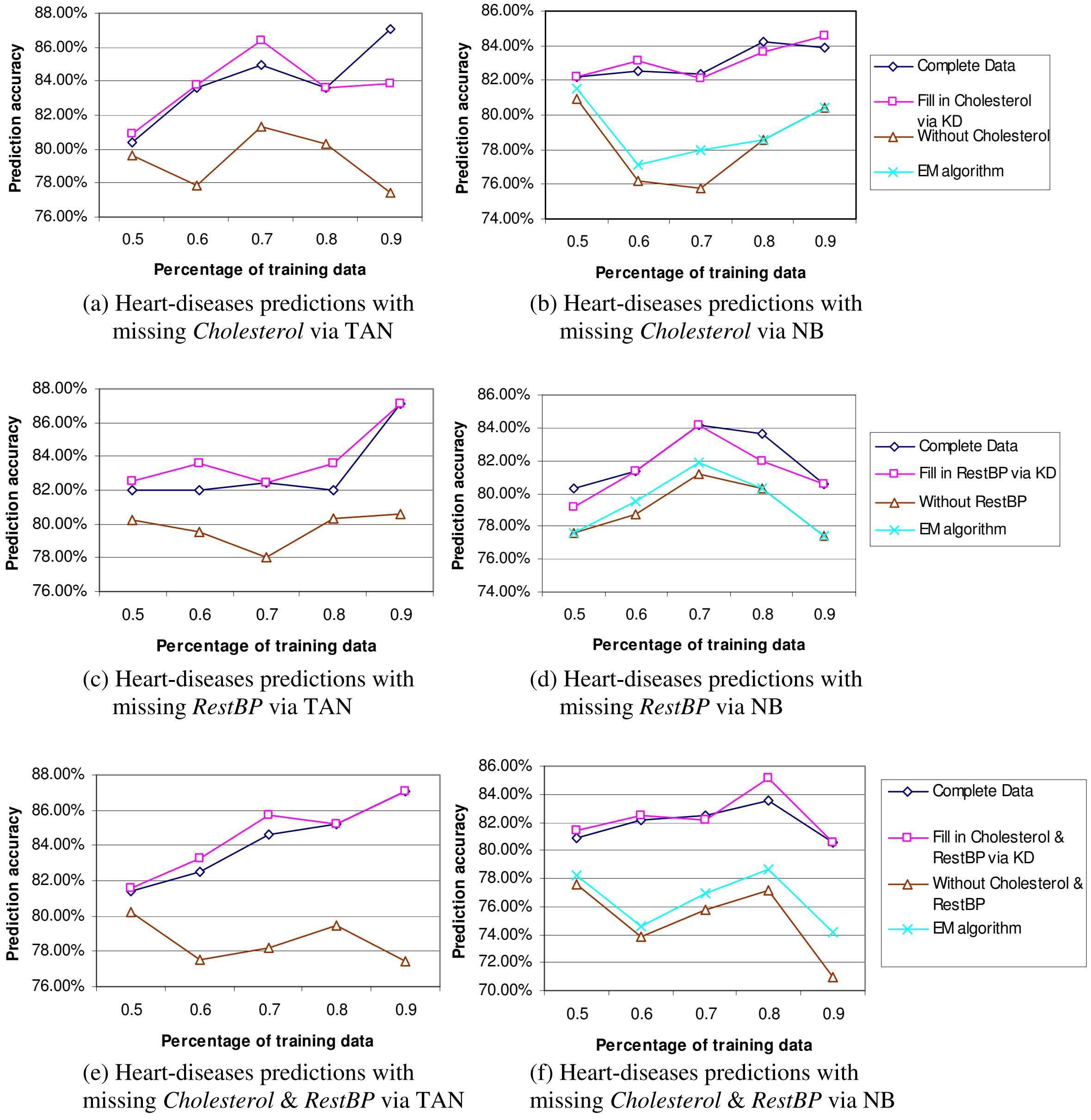
[14] J. W. Myers, K. B. Laskey, and K. A. DeJong. Learning bayesian networks from incomplete data using evolutionary algorithms. 1:458–465, 1999.

[15] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1988.

[16] S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1-3):233–272, 1999.

[17] D. Spiegelhalter and S. Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20:579–605, 1990.

[18] V. R. Vemuri and N. Tang. Web-based knowledge acquisition to solve inverse problems arising in health-care management. In M. S. Takumi Ichimura and K. Yoshida, editors, *Knowledge Based Intelligent System for Health Care*, pages 207–237. CRC Press, 2004.



(a) Heart-diseases predictions with missing *Cholesterol* via TAN

(b) Heart-diseases predictions with missing *Cholesterol* via NB

(c) Heart-diseases predictions with missing *RestBP* via TAN

(d) Heart-diseases predictions with missing *RestBP* via NB

(e) Heart-diseases predictions with missing *Cholesterol* & *RestBP* via TAN

(f) Heart-diseases predictions with missing *Cholesterol* & *RestBP* via NB

Figure 6. Heart-disease prediction accuracies with missing *Cholesterol* (a, b), missing *RestBP* (c, d) or missing *Cholesterol* & *RestBP* (e, f)