Data Mining and Decision Making

Andrew Kusiak
Department of Mechanical and Industrial Engineering, The University of Iowa, Iowa City, Iowa 52242 -1527, USA
andrew-kusiak@uiowa.edu, http://www.icaen.uiowa.edu/~ankusiak

ABSTRACT

Models and algorithms for effective decision-making in a data-driven environment are discussed. To enhance the quality of the extracted knowledge and decision-making, the data sets are transformed, the knowledge is extracted with multiple algorithms, the impact of the decisions on the modeled process is simulated, and the parameters optimizing process performance are recommended.
The applications discussed in this paper differ from most data mining tasks, where the extracted knowledge is used to assign decision values to new objects that have not been included in the training data. For example, in a typical data mining application the equipment fault is recognized based on the failure symptoms. In this paper, a subset of rules is selected from the extracted knowledge to meet the established decision-making criteria. The parameter values represented by the conditions of this set of rules are called a decision signature. A model and algorithms for the selection of the desired parameters (decision signatures) will be developed. The parameters of this model are updated using a framework provided by the learning classifier systems.

Keywords: data mining, temporal data, feature transformation, data transformation, decision making, decision signatures.

1. INTRODUCTION

The problems considered in this paper differ from most data mining tasks where knowledge is extracted and used to assign decision values to the new objects that have not been included in the training data. For example, the equipment fault is recognized (i.e., the value of the fault number is assigned) based on the failure symptoms. There are many applications discussed in this paper, where a subset of rules, in particular a single rule, is selected from the extracted knowledge. The parameter values corresponding to the conditions of the rules in this subset are called a decision signature. The decision signature is used to control the process under consideration.

One of the questions posed in this research is how to construct the most promising decision signatures for large-scale rule bases. The construction of such decision signatures becomes a challenge due to the temporal nature of the processes from which the data sets considered in this research have been collected.

The ideas discussed in this paper are structured in six phases: Phase 1: Data transformation; Phase 2: Rule extraction with alternative algorithms; Phase 3: Decision signature selection; Phase 4: Decision signature validation; Phase 5: A return to the data transformation phase; Phase 6: The acceptance of the decision signature (see Fig. 1).

2. PROBLEM DESCRIPTION

The complexity of decision-making in manufacturing, business, and medical applications is rapidly increasing, as the world is becoming data-driven. To cope with this increasing complexity in a changing environment, new modeling and computing paradigms are needed. The salient features of the new modeling and computing paradigm are:

- Adaptability of decision-making models to the evolving decision environment.

A. Kusiak, Data Mining and Decision Making, in B.V. Dasarathy (Ed.), *Proceedings of the SPIE Conference on Data Mining and Knowledge Discovery: Theory, Tools, and Technology IV*, Vol. 4730, SPIE, Orlando, FL, April 2002, pp. 155-165.

- Ability to handle changing qualitative and quantitative data.
- Short decision response time.
- Large and overlapping problem domains.
- Interpretability of the results.
- Process rather than problem orientation.

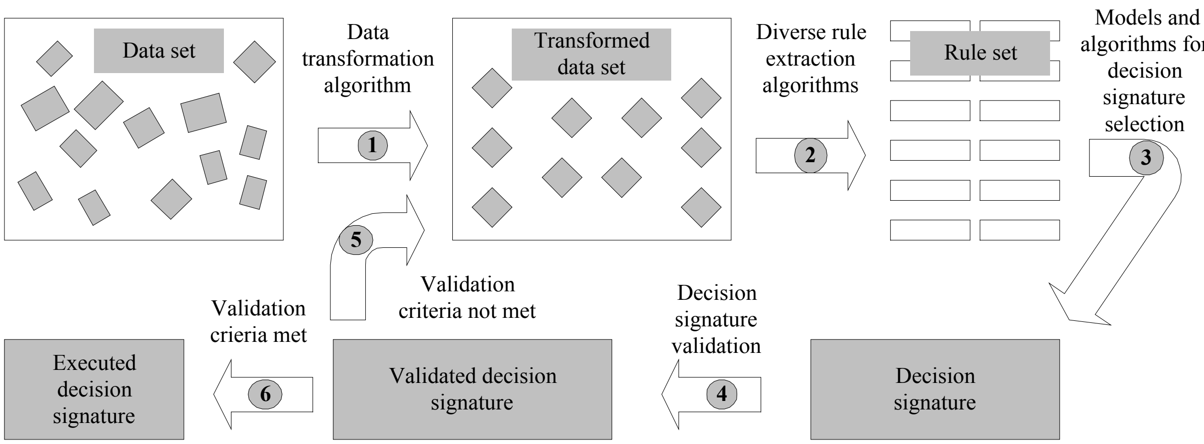

Fig. 1. Structure of the proposed framework.

The models and algorithms discussed in this paper are to meet these challenges. In response to the most recent advances in genetics, where the volume of data is growing at an unprecedented rate, new data and knowledge paradigms are introduced. The data for temporal processes to be considered in this paper has been collected with various acquisition systems. It will be transformed for knowledge extraction and decision-making. The usable form of knowledge extracted from the transformed data is analogous to customized models.

The mining of temporal data sets is to determine the nature of the relationships among features rather than analyzing the values of singular features. In some cases, the temporal behavior of an individual feature may be difficult to capture and might not be reliable for making predictions. Rather than concentrating on the individual feature values, the approach presented in this paper advocates capturing relationships among multiple features in the form of feature functions defined in Section 3.2.2.

2.1 Potential Applications

This approach discussed in this paper applies to temporal processes. Most, if not all, processes are to some degree temporal, however, for various reasons they are often considered as time invariant. The processes of focus to this paper are continuous, however, the data has been collected at discrete intervals.

Aluminum processing plants are interested in production of high quality aluminum products. Though numerous neural controllers are involved in the production process, the values of dozens of parameters are manually set. The decision signatures to be generated by the algorithms proposed in this research will provide the values for these parameter settings.

Semiconductor manufacturers are interested in improving the quality of wafers. For thus far unknown reasons under seemingly similar manufacturing conditions, some wafers attain perfect quality and others are not acceptable. Even within individual plants product quality varies. The concept of decision signatures is to determine the ranges of control parameters leading to the production of wafers of the desired quality (Kusiak 2001a).

Electronic assembly processes face quality problems with printed circuit boards where assemblies rather than components fail the quality test for unknown reasons. The management is not satisfied with the current process control or other tools that provide solutions for a “population” of products rather than an “individual” product. The managers would like to predict circumstances under which an individual product (object) might fail, and thereby prevent this failure. The ideas proposed in this research will utilize the rules extracted from a population of objects to generate knowledge, which will determine conditions preventing the production of faulty products (Kusiak and Kurasek 2001).

DNA manufacturing is the most recent addition to the topics of interest to the approach discussed in this paper. The DNA manufacturing process has evolved from a biological laboratory to an industrial-scale process and it involves many unknowns. Finding ways of improving product quality by generating decision signatures that would determine process control parameters are of interest to DNA manufacturers.

Medical applications call for analysis of vast amount of data and recommend medical actions, e.g., for critically ill infants after open-heart surgery on a minute-to-minute basis. Even the most skilled health professionals have difficulty understanding the relationships between dozens of parameters and translating these relationships into consistent treatment (Kusiak *et al.* 2001).

This subset of features can be obtained from different data sets and by different algorithms. Such a subset of features and their values constitute a decision signature. The decision signature for a test engineer becomes a fault signature; for a physician, a disease signature, and so on. The proposed research is designed to identify decision signatures and use them to solve production quality problems, to find the best interventions for patients suffering from the diseases, and to contribute to finding numerous other answers. If there were just a single decision signature for an application, there would be little room for research. The fact is that decision signatures are highly individualized, and therefore the problem of generating and using these signatures for decision-making needs research attention. In many applications, decisions have to be reached quickly, and therefore the decision signatures can be selected and evaluated before their implementation. Often, the decision signatures have to be evaluated using multiple criteria, e.g., utility and accuracy.

2.2 Data Transformation

Constructive induction is a process of describing objects for improved classification (Wnek and Michalski 1994, and Bloedorn and Michalski 1998). New features are built from the existing ones, and some features (attributes) of objects are modified or deleted. It should be noted that the deletion of features is related to the feature selection problem (Yang and Honavar 1998), which is not addressed in this paper.

In this paper the data transformation aspect of constructive induction is emphasized in order to improve usability, transparency, and the decision-making accuracy of the extracted rules. A novel data transformation scheme is developed together with an algorithm for forming feature functions (see Section 3.2.3).

Data sets can be mined in their raw form (as collected), or they can be transformed. The following three data transformation approaches are widely used in data mining:

- Filling in missing values. For example, the most common value method replaces the missing values with the most frequent values. The data set decomposition method partitions the data set into subsets without missing values that are in turn used for mining (Ragel and Cremilleux 1998).
- Discretization. For example, the equal frequency interval method groups continuous values into k intervals, where each interval contains m/k (possibly duplicated) adjacent values, where m is the number of examples (Dugherty *et al.* 1995). The recursive minimal entropy algorithm establishes intervals by considering the class information entropy (Fayyad and Irani 1993).
- Feature content modification. Feature generalization and specialization methods are discussed in Han and Kamber (2001).

For each of the above three approaches, numerous methods and algorithms have been developed, for example:

- The removal of examples with missing values.
- The most common value method. The missing values are replaced with the most frequent values.
- Data set decomposition. The data set is partitioned into subsets without missing values that are in turn used for mining (Ragel and Cremilleux 1998, Kusiak 2000).

Other data transformation methods are discussed in Cios *et al.* (1998) and Han and Kamber (2001).

In this research, new feature transformation functions and an algorithm for forming these functions are proposed (see Section 3.2).

The data transformation approach has received rather limited attention in the data mining literature under the umbrella of constructive induction (Bloedorn and Michalski 1998). The data transformation approaches proposed in this paper are novel. The data is transformed based on the metrics characterizing the data itself (see Section 3.2.1), rather than based on the result of the costly evaluation of the outcome of the data mining process, see Vafaie and De Jong (1998). Extensive validation is needed to assess the quality of models and algorithms proposed in this paper. Once developed, the models and algorithms should significantly reduce the knowledge validation effort.

2.3 Decision Signature Generation

The topic of decision signature generation has not been sufficiently researched in the data mining literature. Most data mining projects have naturally concentrated on the knowledge discovery process. Decision signatures make novel use of knowledge for setting parameter values aiming at optimizing a process performance. Many processes for which the decision signatures are generated are temporal, therefore the decision signatures are likely to be dynamically generated over the process life cycle. The latter implies that the algorithms for a generation of decision signatures have to be efficient.

This paper offers models and algorithms for the selection of decision signatures. These models will incorporate different objective functions (e.g., related to predictive accuracy and knowledge utility) and constraints. The constraints will consider user preferences (e.g., the maximum length of a decision signature). Optimal and heuristic algorithms will assure scalability of the proposed approach.

3. THE RESEARCH APPROACH

3.1 Rule Extraction

This paper emphasizes decision-making rather than the development of new learning algorithms discussed in the data mining literature and reviewed in Section 3.1. In fact, some of the criticism of the rule extraction approaches might have been due to the rather simplistic matching algorithms used for testing the quality of the extracted knowledge. This paper will demonstrate that high quality decisions can be produced using even “imperfect” data sets. A new approach for mining transformed rather than an "as-is" data set is proposed. One of these transformations involves grouping features into sequences that significantly increase the classification accuracy of the extracted knowledge. This data transformation approach has been tested on independent benchmark sets and has provided consistent improvements of classification accuracy (Kusiak 2001). The data transformation methods are presented in Section 3.2, and their relationship with decision signatures are discussed in Section 3.3.

3.2 Data Transformation

The research on the evaluation of data sets is scarce. In the next section potential metrics for feature evaluation are discussed.

3.2.1 Metrics for Evaluation of Data Sets

Data sets can be directly evaluated by the following metrics:

- Upper and lower approximation measures (defined in Pawlak 1991).
- Classification quality (Pawlak 1991).
- Entropy measure (Quinlan 1986).
- Gini index (Breiman *et al.* 1984).
- Correlation, distribution type, etc.
- Other metrics such as percentage of missing values, data error, discretization parameters, etc.

Classification accuracy of the knowledge extracted from a data set is the most widely used indirect method of feature evaluation (see Kusiak 2001). However, this method is computationally expensive as demonstrated in Vafaie and De Jong (1998).

The metrics for evaluation of data sets will be an important component of the algorithm for forming feature functions that is discussed in Section 3.2.3.

3.2.2 Feature Transformation Functions

Most data mining algorithms establish associations among individual feature values. The approach proposed in this paper captures relationships among feature functions. Examples of feature functions to be considered in this paper include:

- Logic expression of features `Fi, Fj, ...., Fn`, where the <logic operator> = {`AND, OR, NOT, EXOR`}. Note that an ordered set of features linked by the `AND` operator becomes a sequence, e.g., the expression `F2 AND F9 AND F4` is denoted as `F2_F9_F4`.
- Arithmetic expression of features `Fi, Fj, ...., Fn`, where the <arithmetic operator> = {+, -, /, ×, √, ⁿ, | |}, e.g., `F3 - 4.5×F8`, `|F3 - 4.2×F8|`, $(.7\times F2^3 - F4)/(2.1\times F5^2 + .2\times F8)$. Note that the inequality relation `Fi` ≥ `Fj` can be handled by the ratio `Fi/Fj` ≥ `1`.

A rule involving two feature functions, a sequence `5_7_2`, and an inequality relation is shown next.

```
IF (F2_F4_F9 = 5_7_2) AND (F3 < F7) THEN D = Hot
```

The impact of a simple feature transformation method on the classification accuracy is presented in Example 1.

Example 1

Consider the "as-is" data set in Fig. 2 with four features F1 – F4, the decision D, and five objects.

No.	F1	F2	F3	F4	D
1	0	1	0	2	0
2	1	1	0	2	2
3	0	0	0	0	0
4	0	1	1	1	1
5	0	0	1	3	0

Fig. 2. Data set with four features.

No.	F1	F2_F4	F3	D
1	0	1_2	0	0
2	1	1_2	0	2
3	0	0_1	0	0
4	0	1_0	1	1
5	0	0_3	1	0

Fig. 3. Transformed data set of Fig. 2.

Classification quality of a feature set can be expressed as the percentage of all objects in a data set that can be unambiguously associated with the decision value based on this feature set (Pawlak 1991). The classification quality of the features in Fig. 2 is as follows: CQ(F1) = .2, CQ(F2) = .4, CQ(F3) = 0, CQ(F4) = .6. The data set in Fig. 2 was transformed in the data set of Fig. 3, where the two features `F2`, `F4` were replaced with the feature sequence `F2_F4`.

The classification quality of the feature sequence `F2_F4` has the value CQ(F2_F4) = .6, which is higher than that of individual features `F2` and `F4`. The one-out-of *n* (*n* = 5) cross-validation scheme (Stone 1974) has been applied to the rules generated from the data sets in Figs 2 and 3. The results of cross-validation are presented in Fig. 4. The average classification accuracy has increased from 20% for the rules extracted from the data set in Fig. 2 to 60% for the transformed data in Fig. 3.

Example 1 illustrates one of many possible feature functions. Feature sequences with a larger number of features have been successfully tested on a large-scale data set involving equipment maintenance (Kusiak 2001a). The introduction of sequences in the maintenance data set has improved classification accuracy by 21%. Our own computational experience and the results published in the literature (Cattral *et al.* 2001, Vafaie and De Jong 1998, and Lesh *et al.* 2000) indicate that the classification accuracy of the decision rules involving relationships between feature functions may exceed those of the traditional decision rules. In addition, forming feature functions according to user preferences may enhance the transparency of decision-making.

(a)

	Correct	Incorrect	None
Average	20%	60%	20%

(b)

	Correct	Incorrect	None
Average	60%	20%	20%

Fig. 4. Cross-validation results: (a) classification accuracy for the data set in Fig. 2, (b) classification accuracy for the transformed data set of Fig. 3.

The need for more elaborate feature functions leads to the algorithm proposed in the next section.

3.2.3 Algorithm for Forming Feature Functions

Feature functions can be created one at a time or as a group (a bundle), e.g., the sequence `F1_F7_F9` and expression $F1^2/(F8 - 2.3 \times F10)$ make a bundle with two functions. The reasons for creating feature bundles are numerous, e.g., feature dependency, common equations, and user preferences. The listed in Section 3.2.1, metrics for feature evaluation, are used to construct an algorithm for forming feature functions. The algorithm produces feature functions similar to the ones illustrated in Section 3.2.2 for optimization of classification accuracy or any other selected performance measure. Depending on the size of the feature-formation problem, the algorithm can be run as an optimal branch-and-bound algorithm or as a branching heuristic (see Kusiak 2000a). To ensure full scalability of the algorithm, incorporation of all constraints, and to prevent overfitting, the feature-forming algorithm is implemented as an evolutionary computation algorithm (Koza 1992, Bäck 1996). The implementation of the evolutionary feature-forming algorithm involves the shell developed by Kovacs (2001).

3.3 Decision Signature Generation

A decision signature contains a preferred set of feature values leading to optimization of a selected performance measure, e.g., process efficiency. Examples of decision signatures for different applications include:

- ❑ Control signature. A set of feature values or ranges of their values leading to an expected outcome.
- ❑ Plan of medical interventions. A set of medical interventions for treatment of the underlying medical problem.
- ❑ Phenotype obtained from a genotype, which was determined by the genetic programming algorithm optimizing a selected performance measure (e.g., classification accuracy).

The concept of the decision signature applied to one of the industrial data sets considered in this paper is discussed next. Decision signatures are built based on the knowledge extracted from data sets. For example, process control engineers are interested in understanding processes at a level beyond analog and digital controllers, which is wider in scope of control than that of individual controllers. The range of data available for knowledge extraction is the only factor that may limit the scope of control with a data mining approach.

The selection of a decision signature is illustrated in Example 2.

Example 2

Consider a process with two sequential stages, where the first stage is realized in two scenarios. In this case a scenario represents a distinct way of operating and controlling the process due to different properties of the material.

Two data sets for Stage 1 and one for Stage 2 are available, each with three to four features and five to seven objects (see Fig. 5). Stage 1 involves two operational scenarios.

Different learning algorithms have been used to extract decision rules. Three subsets of decision rules have been selected, one for each data set. The selected rules are shown below.

Stage 1: The decision rules selected in Scenario 1 are:

Rule R1. `IF (F1 = 2) THEN (D = DD); [2][4, 5]`
Rule R2. `IF (F3 ∈ [3, 4)) THEN (D = DD); [2] [4, 5]`
Rule R3. `IF (F3 ∈ [4, 5]) THEN (D = DD); [1] [3].`

Stage 1

Scenario 1

No.	F1	F2	F3	D1
1	4	a	14.1	BC
2	4	b	16.2	BC
3	3	b	4.5	DD
4	2	c	3.4	DD
5	2	a	3.3	DD
6	3	b	8.1	EE
7	1	c	7.9	EE

Scenario 2

No.	F4	F5	F6	D1
8	3	7	xc	BA
9	1	8	xc	DE
10	3	12	xa	DD
11	1	12	xa	DD
12	3	8	xd	ED

Stage 2

No.	F8=D1	F9	F10	F11	D
13	BA	4	44.6	ba	Low
14	DE	4	33.7	de	Low
15	ED	3	15.3	ed	Low
16	BC	3	45.2	bc	High
17	DD	1	102.3	aa	High
18	DD	4	100.1	fa	High
19	EE	1	11.9	ee	High

Fig. 5. Data sets for a two-stage process.

These decision rules are presented in the following format:

`IF(Condition) THEN (Outcome);` [Rule support], [Objects represented by this rule]

Each of the two decision rules R1 and R2 is preferred over rule R3 for three reasons:
- The support of the rules R1 and R2 is greater than the support of R3.
- Rules R1 and R2 are supported by the same set of objects {4, 5}.
- The decision `DD` predicted by the simultaneous use of the rules R1 and R2 is more robust that the decision determined by rule R3 due to its association with two independent features.

Stage 1: The decision rules selected in Scenario 2 are:
Rule R4. `IF (F5 = 12) THEN (D = DD); [2] [10, 11]`
Rule R5. `IF (F6 = xa) THEN (D = DD); [2] [10, 11].`

Stage 2: The decision rules selected are:
Rule R6. `IF (F8 = DD) THEN (D = High); [2] [17, 18]`
Rule R7. `IF (F10 ∈ [99, 104]) THEN (D = High); [2] [17, 18].`

The seven selected decision rules have the strongest support of all the rules extracted from the three training data sets in Fig. 5. The decision signature in Fig. 6 has been generated based on these rules.

Note that across two scenarios the four parameters (features) `F2`, `F4`, `F9`, and `F11` are not used to control the process considered in this example.

For temporal processes, the decision signatures are dynamic and are recomputed in order to improve the process outcome. The advantage of decision signatures is that a user can be presented with a set of parameter values (decision signatures) and the corresponding performance metrics before they get implemented. Examples of performance metrics include a confidence interval and a utility of the decision signature. The latter can be measured with various criteria to be defined in this paper, e.g., relevance to the underlying process science.

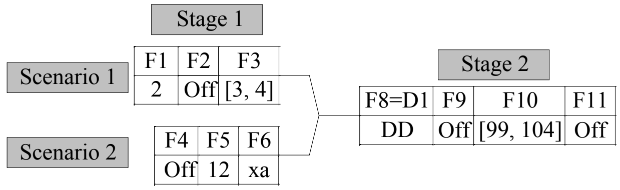

Fig. 6. Decision signature leading to D = High.

Both deterministic and probabilistic metrics will be investigated in this paper and attached to decision signatures. In some applications, decision signatures may be considered as the best practices.

The rule-structuring concept together with the decision signature generation will provide a viable approach for decision-making in a data intensive environment.

3.3.1 Model for Selection of Decision Signatures

The decision signature in Fig. 6 was easy to construct due to the small number of rules and features. The cases involving large numbers of rules and features call for a formal approach for the selection of decision signatures. Besides handling large-scale problems, a model for the selection of decision signatures will allow for the incorporation of various objectives and constraints.

The formulation (1) – (5) presented below illustrates the nature of the decision signature selection model proposed in this paper.

Notation:

m = the number of rules
n = the number of features (this value is used in computing the distance measure d_{ij})
q = the number of decision classes
S_k = the set of rules of class k
c_i = the rule i performance metric (e.g., rule support)
d_{ij} = the distance between rules i and j
x_i = 1, if rule i is selected, otherwise $x_i = 0$
y_{ij} = 1, if rules i and j are selected, otherwise $y_{ij} = 0$

$$\text{Max } \tfrac{1}{2} \sum_i \sum_j d_{ij} y_{ij} + \sum_i c_i x_i \qquad (1)$$

$$\sum_{i \in S_k} x_i = 1 \qquad \text{for } k = 1, \ldots, q \qquad (2)$$

$$x_i \le y_{ij} \qquad \text{for } i, j = 1, \ldots, m \qquad (3)$$

$$x_i = 0, 1 \qquad \text{for } i = 1, \ldots, m \qquad (4)$$

$$y_{ij} = 0, 1 \qquad \text{for } i, j = 1, \ldots, m \qquad (5)$$

The objective function (1) maximizes the total distance between rules and the total rule performance metric. Some domain experts interested in a decision signature that would be most dissimilar from all other signatures have suggested this form of an objective function. Constraint (2) makes sure that for each class exactly one rule is selected. Constraint (3) imposes consistency of the decision variables x_i and y_{ij}. The non-negativity of variables x_i and y_{ij} is ensured by constraints (4) and (5).

Due to the temporal nature of the data sets considered in this paper, each decision signature might be valid for only a certain period of time. In other words, the parameters recommended by a decision signature may or may not lead to the desired outcome. This is most likely caused by the insufficient representation of the process with the collected data. It is also true that certain decision signatures may perform better than others. The latter justifies the need for modeling the decision signature selection problem in way that it could be solved dynamically. To justify the best match of the model with the decision-making problem some of the parameters of this model should be dynamically updated.

One way to make the model (1) – (5) and its generalizations dynamic would be by introducing two parameters α and β in the objective function (1), i.e., $\alpha\Sigma\Sigma d_{ij}y_{ij} + \beta\Sigma c_i x_i$. The two parameters would have to be periodically updated each time the model (1) – (5) would have to be solved.

The proposed modeling and solution approach fits the framework of learning classifier systems, e.g., XCS (Wilson 1995). The integer programming model will add structure to the XCS framework by generating an action set based on the solution of the signature selection model. The reward values of the XCS algorithm could correspond to the parameters α and β, or any other parameters to be included in the model to be developed in this paper. The value of reward parameters could be dynamically computed based on the number of times of the decision signature has been successful, rule support, statistical properties of the features, etc. The function used to update the reward parameters would be developed in this research. To facilitate efficient implementation of the XCS system, the shell developed by Kovacs (2001) has ben used.

3.3.2 The Algorithm for Selection of Decision Signatures

Due to high computational complexity of the decision signature selection problem, two classes of algorithms are proposed for solving large-scale models. Small size models can be solved with standard optimization algorithms. The first algorithm for large-scale models is a constructive algorithm that quickly generates a feasible solution and then improves it in subsequent iterations. Algorithms taking advantage of the special structure of the decision signature selection problem were explored. The second algorithm is based on the concepts from evolutionary computation (Fonseca and Fleming 1995). The decision signature selection algorithm addresses a wide range of biological operators such as recombination, mutation, and dominance and their impact on evolutionary efficiency. Stable genotype combinations in a given fitness landscape are considered. The need for an evolutionary computation approach is due to the changing nature of the extracted knowledge. The following aspects of the evolutionary computation are emphasized in the research reported in this paper:

- Solution representation. An AND/OR graph appears to be a suitable representation for solutions of the rule-structuring problem due to its ability to represent alternatives.
- Fitness functions. The research challenge is to define a fitness function that would be effective without the need for a genotype-phenotype transformation that is complex. The effectiveness of various similarity measures (e.g., see Kusiak 2000) defined on chromosomes will be tested.
- Specialized genetic operators: The nature of the rule-structuring problem and the proposed solution representation method imply that special-purpose genetic operators need to be defined. These operators will accommodate alternative genotypes.
- Adaptive representation schemes and the use of introns (defined in Benzhaf *et al.* 1998) and recognized in the Genome Project Report (February 2001) as potentially important in evolution.
- Solution-structure modification schemes.
- Selection of algorithm parameters, such as initialization method, selection method, probabilities, and stopping criteria.

Besides the hard (formally defined constraints), the signature selection algorithm considers soft constraints leading to the following knowledge properties:

- Knowledge diversity. The knowledge to be used for structuring will be diverse due to the data transformation schemes applied before knowledge extraction and the use of algorithms of different types for knowledge extraction.
- Knowledge interestingness. According to Han and Kamber (2001) knowledge is interesting if it is unexpected (surprising to the user) and actionable (the user can do something with it).
- Knowledge comprehensibility. A user should able to comprehend and have confidence in the extracted knowledge regardless of his/her background.

4. CONCLUSION

The volume of information available for decision-making is growing at an unprecedented rate. Models, algorithms, and tools offered by decision-making theory, operations research, mathematical programming, and other disciplines have met some of these growing needs for data processing. The expanding scope of decision-making problems calls for a new class of computational tools. Data mining satisfies some of these needs by offering capabilities to process data of different types (e.g., qualitative and quantitative) and originating at different sources. None of the traditional approaches is able to cover such a wide spectrum of data diversity.

The ideas discussed in this paper are unique, as the accuracy associated with the decision signatures is evaluated prior to making predictions. Some decisions may be highly accuracy, while others may entail a certain degree of probability. This ability to assess *a priori* the accuracy of a decision makes the proposed ideas of interest to technology, business, and medical applications.

Besides classification accuracy, the utility of decision signatures was evaluated based on the specific criteria to be defined in this paper, for example, feature diversity, current practice, and knowledge comprehensibility. The evaluation was done following the group assessment and decision-making approaches developed by the human factors community. A group decision-making algorithm was used to rank order the decision signatures and update the utility index. Examples of group decision-making algorithms with various preference functions are discussed in Kusiak (1999).

REFERENCES

Bäck, T. (1996), *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York.

Benzhaf, W., P. Nordin, R.E. Keller, and F.D. Francone (1998), *Genetic Programming; An Introduction*, Morgan Kaufmann, San Francisco, CA.

Bloedorn, E. and R.S. Michalski (1998), Data-driven constructive induction, *IEEE Intelligent Systems*, Vol. 13, No. 2, pp. 30-37.

Breiman, L., J.H. Friedman, R.A. Olshen, and P.J. Stone (1984), *Classification and Regression Trees*, Wadworth International Group, Belmont, CA.

Cattral, R., F. Oppacher, and D. Deugo (2001), Supervised and unsupervised data mining with an evolutionary algorithm, *Proceedings of the 2001 Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, pp. 767-776.

Cios, K., W. Pedrycz, and R. Swiniarski (1998), *Data Mining: Methods for Knowledge Discovery*, Kluwer, Boston, MA.

Dugherty, D., R. Kohavi, and M. Sahami (1995), Supervised and unsupervised discretization of continuous features, *Proceedings of the 12th International Machine Learning Conference*, pp. 194-202.

Fayyad, U.M. and K.B. Irani (1993), Multi-interval discretization of continuously-valued attributes for classification learning, *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pp. 1022-1027.

Fonseca, C.M. and P.J. Fleming (1995), An overview of evolutionary algorithms in multiobjective optimization, *Evolutionary Computation*, Vol. 3, No. 1, pp. 1-16.

Han J. and M. Kamber (2000), *Data Mining: Concepts and Techniques*, Academic Press, San Diego, CA.

Kovacs, T. (2001), What should a classifier system learn?, *Proceedings of the 2001 Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, pp. 775-782.

Koza, J. (1992), *Genetic Programming*, MIT Press, Cambridge, Mass.

Kusiak, A. (1999), *Engineering Design: Products, Processes, and Systems*, Academic Press, San Diego, CA.

Kusiak, A. (2000), Decomposition in data mining: An industrial case study, *IEEE Transactions on Electronics Packaging Manufacturing*, Vol. 23, No. 4, pp. 345-353.

Kusiak, A. (2000a), *Computational Intelligence in Design and Manufacturing*, John Wiley, New York.

Kusiak, A. (2001), Feature transformation methods in data mining, *IEEE Transactions on Electronics Packaging Manufacturing*, Vol. 24, No. 3, 2001, pp. 214-221.

Kusiak, A. (2001a), Rough set theory: a data mining tool for semiconductor manufacturing, *IEEE Transactions on Electronics Packaging Manufacturing*, Vol. 24, No. 1, pp. 44-50.

Kusiak, A. and C. Kurasek (2001), Data mining analysis of printed-circuit board defects, *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 2, pp. 191-196.

Kusiak, A., I.H. Law, and M.D. Dick (2001), The G-algorithm for extraction of robust decision rules: Children's postoperative intra-atrial arrhythmia case study, *IEEE Transactions on Information Technology in Biomedicine*, Vol. 5, No. 3, pp. 225-235.

Lesh, N., M.J. Zaki, and M. Ogihara (2000), Scalable feature mining for sequential data, *IEEE Intelligent Systems*, Vol. 15, No. 2, pp. 48-55.

Pawlak, Z. (1991), *Rough Sets: Theoretical Aspects of Reasoning About Data*, Kluwer, Boston, MA.

Quinlan, J.R. (1986), Induction of decision trees, *Machine Learning,* Vol. 1, No 1, pp. 81-106.

Ragel, A. and B. Cremilleux (1998), Treatment of missing values for association rules, *Proceedings of the Second Pacific Asia Conference, PAKDD'98*, Melbourne, Australia.

Stone, M. (1974), Cross-validatory choice and assessment of statistical predictions, *Journal of the Royal Statistical Society*, Vol. 36, pp.111-147.

Vafaie, H. and K. De Jong (1998), Feature space transformation using genetic algorithms, *IEEE Intelligent Systems*, Vol. 13, No. 2, pp. 57-65.

Wilson, S.W. (1995), Classifier fitness based on accuracy, *Evolutionary Computation*, Vol. 3, No. 2, pp. 149-175.

Wnek, J. and R.S. Michalski (1994), Hypothesis-driven constructive induction in AQ17-HCI: A method and experiments, *Machine Learning*, Vol. 14, No, 2, pp. 139-168.

Yang, J. and V. Honavar (1998), Feature subset selection using a genetic algorithm, *IEEE Intelligent Systems*, Vol. 13, No. 2, pp. 44-49.