

Data Analysis: Models and Algorithms

Andrew Kusiak
Intelligent Systems Laboratory
4312 Seamans Center
The University of Iowa
Iowa City, Iowa 52242 – 1527

andrew-kusiak@uiowa.edu
<http://www.icaen.uiowa.edu/~ankusiak>

Abstract

Data analysis has many facets, ranging from statistics to engineering. In this paper basic models and algorithms for data analysis are discussed. Novel uses of cluster analysis, precedence analysis, and data mining methods are emphasized. The software for the cluster analysis algorithm and the triangularization is presented.

Keywords: Cluster analysis, data mining, precedence analysis, data classification

1. Introduction

The growing volume of data has sparked renewed interest in data analysis. The two most important problems in data analysis are:

- Classification, and
- Pattern analysis

The basic goal of classification is to provide insights into a data set by its simplification, i.e., grouping data objects into categories. Cluster analysis provides models and algorithms for classification.

The most useful methods for analysis of patterns in data are offered by data mining, in particular machine learning algorithms.

Algorithms for clustering and data mining are emphasized in this paper.

2. Cluster Analysis

Cluster analysis algorithms form groups of objects that share common properties. The early cluster analysis algorithms are the leader algorithm, the k-means algorithm, ISODATA, and the quick partition algorithm (Anderberg 1973). Cluster analysis algorithms may or not require specification of the number of objects in a class or the number of classes. Therefore cluster analysis falls into the category of unsupervised classification tools. For review of most recent cluster analysis models and algorithms see Kusiak (2000).

The artificial intelligence community has studied conceptual clustering (Michalski 1983) as well as other methodologies with a statistical flavor. The basic idea behind conceptual clustering is that instead of considering the similarity between objects, conceptual cohesiveness among the objects is considered as a criterion for classification. Conceptual clustering techniques are context based and arrange objects hierarchically (Michalski 1983).

Autoclass is a known Bayesian classifier proposed by Cheeseman et al. (1988). Their strategy involved making simplifying assumptions about the classification model. Rather than searching

the entire hypothesis space and considering all states of the world, they focused on a limited number of possible states thereby reducing the number of possibilities to be analyzed. In the case of real value attributes, the assumption is that data is distributed according to the normal probability distribution. A multinomial distribution is assumed for the discrete attributes. Autoclass uses the Expectation Maximization (EM) algorithm (Dempster 1977), to estimate the class parameters that maximize the posterior probability of the parameters for a given number of classes.

One of the most efficient cluster analysis algorithms is the cluster identification algorithm presented in Kusiak (2000). The use of the modified version of this algorithm, called the extended cluster identification algorithm, is demonstrated next.

Consider that matrix in Figure 1 with four rows, denoted as R1-R4, and six columns, denoted as C1-C6. An '*' in Figure 1 indicates that a column (attribute) applies to the corresponding row (object), e.g., object R1 has two attributes C2 and C4.

	C1	C2	C3	C4	C5	C6
R1		*		*		
R2			*			*
R3	*	*				
R4		*		*	*	

Figure 1. Matrix with four objects (R1-R4) and six attributes (C1-C6)

The extended cluster identification algorithm (Kusiak 2000) transforms the matrix in Figure 1 in the matrix in Figure 2.

	C1	C2	C4	C5	C3	C6
R1		*	*			
R3	*	*				
R4		*	*	*		
R2					*	*

Figure 2. The transformed matrix from Figure 1

The structure of the matrix in Figure 2 has been further improved in Figure 3 by removing columns (attributes) C1 and C5 from the clustering process. The two columns are placed in Figure 3 next to the diagonally structured matrix with two clusters of objects {R1, R3, R4} and {R2}, and two clusters of attributes {C2, C4} and {C3, C6}.

	C2	C4	C3	C6	C1	C5
R1	x	x				
R3	x				x	
R4	x	x				x
R2			x	x		

Figure 3. The transformed matrix from Figure 1 with separated columns C1 and C2

The cluster analysis that was demonstrated in Figures 1 through 3 did not involve precedence relationships among objects or attributes. To analyze such precedences another algorithm needs to be applied, e.g., the triangularization algorithm described in Kusiak (1999). This algorithm is available at <http://www.icaen.uiowa.edu/~ankusiak/process-model.html>.

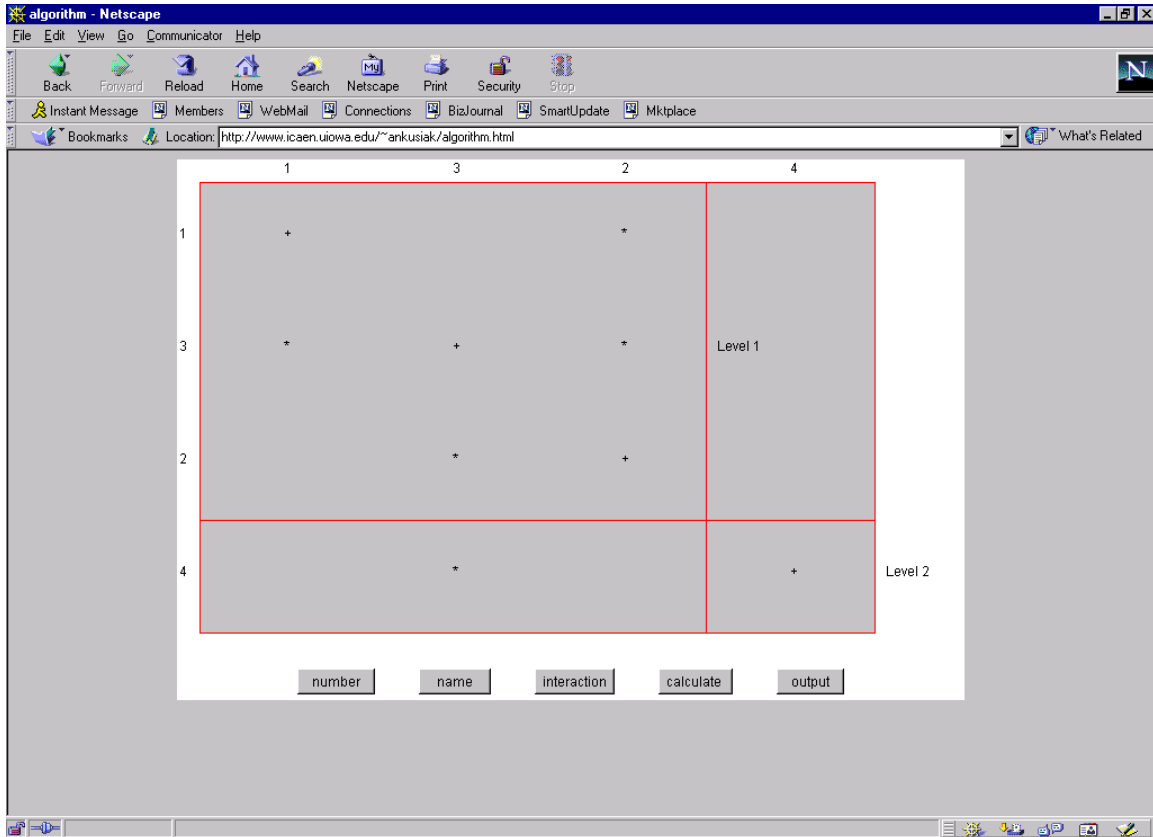


Figure 4. Four objects organized by the triangularization algorithm

Each ‘*’ in Figure 4 denotes a precedence relationship between the corresponding column (object) and row (object). The result in Figure 4 indicates that there are two cycles among the four objects. The cycles are symbolized by the two ‘*’ above the matrix diagonal.

3. Competitive Learning

The basic idea behind competitive learning or ‘winner takes all’ learning is similar to the natural selection principle proposed by Darwin and attributed to human and animal evolution. In a ‘winner takes all’ model, neurons compete in response to a stimulus and the best responding neuron wins the competition and rewards itself by increasing the strength of its active synapses. Variations of the basic competitive learning model include Kohonen's self-organizing maps and Adaptive Resonance Theory. Kohonen's self-organizing maps are topographic maps autonomously organized by a cyclic process of comparing input patterns to the weight vector stored in each output node. Grossberg's Adaptive Resonance Theory (ART) addresses the problem of instability of learned patterns by adding extensive feedback connections.

ART1 - ART3, ART Map, ART Star and Fuzzy ART are some of the variations on the Adaptive Resonance architecture (Grossberg and Carpenter 1991). ART1 classifies binary input patterns in an unsupervised environment whereas ART2 works in the real pattern space. ART Map and ART Star modify the basic ART architecture for training in a supervised environment.

4. Inductive Learning and Data Mining

Machine learning and data mining are used to identify patterns in data sets. Langley and Simon (1995) grouped machine learning research into the following categories:

- a) Neural networks
- b) Genetic algorithms
- c) Case-based learning
- d) Rule induction
- e) Analytical learning

Some of the best-known learning algorithms in the last three categories are listed next.

ID3: Induction Decision Tree is a supervised learning algorithm developed by Quinlan (1986).

AQ15: Inductive learning system generates decision rules, where the conditions are logical formulas (Michalski et al. 1986). Domain knowledge is used to generate new attributes that are not present in the input data.

Naïve-Bayes: A simple induction algorithm that computes conditional probabilities of the classes. Given the instance, it selects the class with the highest posterior probability (Domingos and Pazzani 1996).

OODG: Oblivious read-Once Decision Graph induction algorithm for building oblivious decision graphs, using a bottom-up approach (Kohavi 1995).

Lazy decision trees: An algorithm for building the best decision tree for every test instance developed by Friedman et al. (1996).

C4.5: The decision-tree induction algorithm by Quinlan (1993).

CN2: The direct rule induction algorithm by Clark (1989). This algorithm combines the best features of both ID3 and AQ15, where it uses pruning techniques similar to the techniques employed in ID3 and similar to the decision rules used in AQ15.

IB: The set of instance based learning algorithms by Aha (1992).

OCI: The Oblique decision-tree algorithm by Murthy et al. (1994).

T2: The two-level error-minimizing decision tree by Auer et al. (1995). It minimizes the number of errors and discretizes continuous attributes.

Other major developments in learning and data mining are summarized in the edited volumes by Lin and Ceccerone (2000), Carbonell (1990), Michalski et al. (1998) and the book by Mitchell (1997). For a survey of important applications of machine learning see Langley and Simon (1995) and Kusiak et al. (2000).

5. Rule Structuring

Rule structuring is to enhance the decision-making capability of the knowledge generated with learning algorithms. The need for knowledge structuring is supported by the notion of cognitive maps and mental models discussed in Carroll and Olson (1987) and Wickens et al. (1998). By structuring decision rules a human dimension will be incorporated into the knowledge extracted from data. The idea of structured knowledge is introduced by two examples of simplified decision tables presented in Figure 5. In the table of Figure 5(a), each of the four decisions A – D is made based on the same number of features. A learning algorithm has derived each of the four decision rules based on 1000 examples. There is no exception to these rules, which creates an ideal decision-making setting that could be easily automated. The decision-maker matches the features of a new decision case with the features in the decision table and assigns the new case a decision equal to one of the four decision rules represented in the table. For example, a new case with the feature values F1 = yes, F2 = 1, F3 = 1.9 would be assigned decision B by rule 2 of Figure 5(a). Note that in this table, the decisions are differentiated based on the feature values, not the features.

In the decision table in Figure 5(b) the decisions A – D are differentiated on features. Each of the four decisions is made based on the values of three to four different features. The feature sets associated with each of the four rules and decisions are mutually exclusive.

a)	F1	F2	F3	Decision	Support	b)	F1	F7	F10	F4	F6	F9	F3	F5	F8	F12	F2	F6	F11	Decision	Support
Rule 1	yes	1	[8.1-9.9]	A	1000 examples	Rule 1	ax	2	[7.7-9.1]											A	1000 examples
Rule 2	yes	1	[1.7-2.1]	B	1000 examples	Rule 2				[1.3-1.7]	bb	4								B	1000 examples
Rule 3	no	2	[2.2-4.9]	C	1000 examples	Rule 3							di	7	yes	no				C	1000 examples
Rule 4	yes	2	[5.0-8.0]	D	1000 examples	Rule 4											no	4	[52-56]	D	1000 examples

Figure 5. Examples of simple decision tables: a) single feature table, b) multi-feature table

Analyses of many engineering and medical data sets indicate that in many cases decision tables have distinct structures. Exploring different structures of tables will be helpful in decision making by:

- ❑ Decision process becoming transparent to the user and computing environment
- ❑ Supporting data evolution
- ❑ Increasing decision accuracy
- ❑ Exposing missing feature, which would be helpful in planning future data acquisition and research directions

The decision table in Figure 6 illustrates the case where decisions are differentiated based on features and their values.

	Decision	Support
Rule 1	ax 2 [7.7-9.1] >7, 3, no	A 100 examples
Rule 2	[1.3-1.7] bb 2	B 37 examples
Rule 3	bz 2 di 7 yes no [<7]	C 81 examples
Rule 4	1 no 4 [52-56]	D 45 examples

Figure 6. Example of a decision table with differentiation based on features and their values

The decision table structure and the decision differentiation methods are determined by factors such as:

- ❑ Type of a learning algorithm
- ❑ Rule selection criteria
- ❑ Constraints and objective functions imposed on a decision table structure

The decomposition problem considered in this paper is NP-hard, therefore heuristic rather than optimal algorithms are likely to be developed. Note that a learning algorithm may produce more than one rule for a decision. The fact that more than one learning algorithm will be used at a time will result in multiple rules per decision. This will complicate the unstructured decision table and it will increase computational complexity of the table-structuring problem.

The structured decision tables have multiple applications. First, they can be used to generate backbones of visualization environments (e.g., virtual reality), and may have a profound impact

on the quality and transparency of decision making. Second, the decision signatures derived from these tables are useful in surfing through the growing volume of information. Third, the tables support the knowledge discovery process by fusing information from diverse sources.

6. Data Engineering

The term data engineering introduced in this paper is analogous to the term genetic engineering and it has some relationship with genetic programming (Koza 1992) and evolutionary computation in general (Bentley 1999). Similarly to genetic engineering that may use simple methods such as selective breeding to complicated ones such as gene cloning, data engineering methods will vary in type and scope. The data engineering methods can improve the quality of decisions generated based on the knowledge stored in decision tables. The improvements will be accomplished by a better handling of exceptions. Three examples of data engineering actions performed on the entries of a decision table are discussed next:

- Modifying feature values
- Modifying feature sets
- De-coupling compound decisions

These actions aim at improvement of the generalization capability of the rules extracted in the learning phase of data mining. The first two actions sound intuitive, however, the data engineering methods leading to accomplishing them are not. In support of these actions, a range of methods needs to be derived. At the lowest level of data granularity genetically inspired evolutionary approaches will be developed. At the higher level, application of the design of experiments (DOE) approach appears to be promising. The biological concepts combined with DOE theory will be applied to develop data engineering methods through experimentation with industrial and medical data sets. The third action requires more elaboration. Rules extracted from a data set may include a multitude of decisions rather than a singular decision. This is due to the fact that training sets may be ‘tainted’ with vectors of feature values that are assigned compound decisions. Such training sets may include a mixture of data with singular and compound decisions. The data engineering approach identifies features and their values with a single decision, thus ‘breaking down’ the compound decisions into unique decisions as illustrated in Figure 7.

The analysis of the matrix in Figure 7(a) revealed that the decision labeled as C could be de-coupled into decision A and decision B in Figure 7(b).

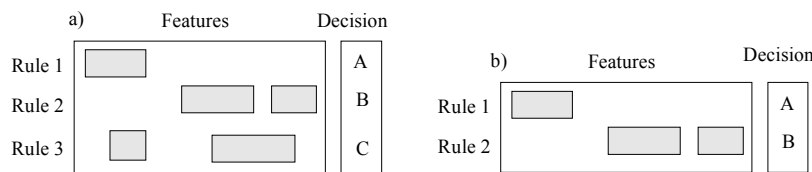


Figure 7. Decision matrix: a) compound decision C, b) decision C de-coupled in A and B

The concepts from genetics, decomposition, design of experiments and statistical process control will be used to develop data engineering methods. The flavor of one of the data engineering methods based on statistical process control (SPC) is illustrated next.

SPC Data Engineering Method

The decision making approach proposed Kusiak et al. (2000) generates accurate decisions for high percentage of cases with unknown outcomes, rather than making decisions for all cases with error. This approach is ‘individual object’ oriented in contrast to ‘population-based’ approaches

followed by neural networks, regression analysis, and quality control. The latter three approaches are concerned with describing the entire population of data (Besterfield 1994). Statistical control methods are useful in analyzing outcome trends and improving processes. The proposed data engineering methods will extend and complement the SPC approach by providing a justification of the decisions made. The essence of the relationship between statistical control and data mining is captured in Figure 8.

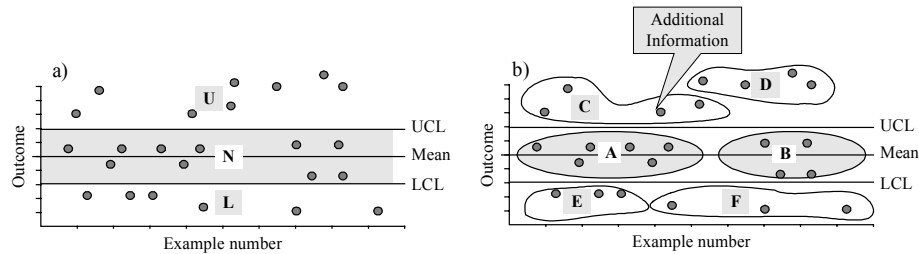


Figure 8. Data interpretation: a) control chart, b) data mining

The process control chart in Figure 8(a) divides the outcome population into three regions, U (upper), N (nominal), and L (lower). The information analyzed with the SPC approach is typically numerical and no direct relationship is captured between these three regions and the features. The rules extracted from an SPC data set decompose the outcomes into numerous regions, for example A – F in Figure 8(b), other than the three regions in Figure 8(a). The information used to generate these partitions may come from different sources and may be qualitative or quantitative.

7. Conclusion

In this paper, basic models and algorithms for data analysis were discussed. Novel uses of methods based on cluster analysis, precedence analysis, and data mining were discussed. The software for the extended cluster identification algorithm and the triangularization was presented.

Acknowledgement

The author expresses appreciation to the University of Iowa students, Z. Ren (Electrical and Computer Engineering) for coding the extended cluster identification algorithm, and D. Wang (Industrial Engineering) for the triangularization algorithm code.

References

- Aha, D.W. (1992), Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms, *International Journal of Man-Machine Studies*, Vol. 36, No. 2, pp. 267-287.
- Anderberg, M.R. (1973), *Cluster Analysis for Applications*, Academic Press, New York.
- Auer, P., R. Holte and W. Maass (1995), Theory and application of agnostic PAC-learning with small decision trees, in A. Prieditis and S. Russell, Eds, *ECML-95: Proceedings of 8th European Conference on Machine Learning*, Springer Verlag, New York.
- Bentley, P.J., Ed. (1999), *Evolutionary Design by Computers*, Morgan Kaufmann, San Francisco, CA.

- Besterfield, D.H. (1994), *Quality Control*, Prentice Hall, Englewood Cliffs, N.J.
- Carbonell, J.G., Ed (1990), *Machine Learning: Paradigms and Methods*, MIT Press, Cambridge, MA.
- Caroll, J.M. and J. Olson (1987), *Mental Models in Human-Computer Interaction: Research Issues About the User of Software Knows*, National Academy Press, Washington, DC.
- Clark, P. and R. Boswell (1989), The CN2 induction algorithm, *Machine Learning*, Vol. 3, No. 4, pp. 261-283.
- Domingos, P. and M. Pazzani (1996), Beyond independence: conditions for the optimality of the simple Bayesian classifier, *Machine Learning: Proceedings of the Thirteenth International Conference*, Morgan Kaufmann, Los Altos, CA, pp. 105-112.
- Cheeseman, P., J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman (1988). Autoclass: A Bayesian classification system. In M. B. Morgan, Ed., *Proceedings of Fifth International Conference on Machine Learning*, Morgan Kaufmann, San Mateo, CA, pp. 54-64.
- Dempster, A.P. (1977), Maximum likelihood from incomplete data via the EM algorithm, *Royal Journal of Statistical Society, Series B*, Vol. 39, pp. 1-38.
- Friedman, J., Y. Yun, and R. Kohavi (1996), Lazy decision trees, *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, AAAI Press and MIT Press.
- Grossberg, S. and G. Carpenter (1991), *Pattern Recognition by Self-Organizing Neural Networks*, MIT Press, Cambridge, MA.
- Kohavi, R. (1995), Wrappers for Performance Enhancement and Oblivious Decision Graphs, Ph.D. Thesis, Computer Science Department, Stanford University, Stanford, CA.
- 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), *Computational Intelligence in Design and Manufacturing*, John Wiley, New York.
- Kusiak, A., J.A. Kern, K.H. Kernstine, and T.L. Tseng (2000), Autonomous Decision-Making: A Data Mining Approach, *IEEE Transactions on Information Technology in Biomedicine*, Vol. 4, No. 4, pp. 274-284.
- Landis S.A., T. Murray, S. Bolden, and P.A. Wingo (1999), *Cancer Journal for Clinicians*, Vol. 49, pp. 8-31.
- Lin, T.Y. and N. Ceccerone, Eds. (2000), *Rough Sets and Data Mining*, Kluwer, Boston, MA.
- Michalski, R.S. (1983). A theory and methodology of inductive learning. In R.S. Michalski, J.G. Carbonell, and T. M. Mitchell, Eds, *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufmann, Los Altos, CA.
- Michalski, R.S., I. Bratko, and M. Kubat, Eds (1998), *Machine Learning and Data Mining*, John Wiley, New York.
- Michalski, R.S., I. Mozetic, J. Hong, and N. Lavrac (1986), The multi-purpose incremental learning system AQ15 and its testing application to three medical domains, *Proceedings of the 5th National Conference on Artificial Intelligence*, AAAI Press, Palo Alto, CA, pp. 1041-1045.
- Mitchel, T. (1997), *Machine Learning*, MacGraw Hill, New York.
- Murthy, S.K. and S. Salzberg (1994), A system for the induction of oblique decision trees, *Journal of Artificial Intelligence Research*, Vol. 2, No. 1, pp. 1-33.
- Pawlak Z. (1982), Rough sets, *International Journal of Information and Computer Science*, Vol. 11, No. 5, pp. 341-356.
- Quinlan, J.R. (1993), *C4.5: Programs for Machine Learning*, Morgan Kaufmann, Los Altos, CA.
- Quinlan, J.R. (1986), Induction of decision trees, *Machine Learning*, Vol. 1, No 1, pp. 81-106.

Wickens, G., S.E. Gordon, and Y. Liu (1998), *An Introduction to Human Factors Engineering*, Harper Collins, New York, N.Y.

Proceedings of the SPIE Conference on Intelligent Systems and Advanced Manufacturing, P.E. Orban and G.K. Knopf (Eds), SPIE, Vol. 4191, Boston, MA, November 200, pp. 1-9.