CHAPTER 17

Non-Traditional Applications of Data Mining

Andrew Kusiak
andrew-kusiak@uiowa.edu
Intelligent Systems Laboratory, Department of Industrial Engineering,
4312 Seamans Center, The University of Iowa, Iowa City, IA 52242 – 1527

ABSTRACT

Machine learning offers algorithms for extraction of knowledge in an understandable form based on historical data. It is viewed as a key tool in development of autonomous systems. This chapter shows that learning algorithms can be used for novel problem solving in engineering design and manufacturing. Decomposition is a key problem in the latter two application areas. A data mining approach is used for matrix decomposition for the case with unknown and known decisions associated with each object in the matrix. Of particular interest to control of manufacturing processes is the case when decisions, e.g., product quality, are ill-defined. Data mining is a viable tool for solving problems with ill-defined outcomes.

A. Kusiak, Non-Traditional Applications of Data Mining, in D. Braha (Ed.), *Data Mining for Design and Manufacturing*, Kluwer, Boston, MA, 2001, pp. 401-416.

INTRODUCTION

Data mining techniques and tools explore patterns and associations in databases. As a new approach, it awaits applications in diverse areas. This chapter discusses a new class of potential applications of data mining that have not been discussed in the literature, such as:

- Matrix decomposition with learning algorithms; Machine learning algorithms produce interesting matrix structures.
- Learning from data sets with ill-defined decisions; Decisions assigned to objects in a training data set may be ill-defined, e.g., the object may be classified as category B rather than C for the set {A, B, C} of decision values.
- Using learning from data sets with ill-defined outcomes to enhance decision-making accuracy; Approximate rules identify objects of a training set with conflicting outcomes which are assigned new decision values thus improving classification accuracy of decion rules.

Learning (classification) systems fall into five general categories:

A. Classical statistical methods (e.g., linear, quadratic, and logistic discriminant analyses) (see Michie *et al.* 1994).
B. Modern statistical techniques (e.g., projection pursuit classification, density estimation, *k*-nearest neighbor, casual networks, Bayes theorem [Domingos and Pazzani 1996]).
C. Neural networks (e.g., backpropagation, Kohonen, linear vector quantifiers, and radial function networks) [see Michie *et al.* 1994]).
D. Decision tree algorithms (e.g., ID3 [Quinlan 1986], CN2 [Clark 1989], C4.5 [Quinlan 1993], T2 [Auer *et al.* 1995], Lazy decision trees [Friedman *et al.* 1996], OODG [Kohavi 1995], OC1 [Aha 1992], AC, BayTree, CAL5, CART, ID5R, IDL, TDIDT, and PROSM [all discussed in Michie *et al.* 1994]).
E. Decision rule algorithms (e.g., AQ15 [Michalski *et al.* 1986], LERS [Grzymala-Busse 1997, and numerous other algorithms based on the rough set theory [Pawlak 1991]).

In this chapter the algorithms of the last category are used for rule extraction from data sets.

A. Kusiak, Non-Traditional Applications of Data Mining, in D. Braha (Ed.), *Data Mining for Design and Manufacturing*, Kluwer, Boston, MA, 2001, pp. 401-416.

MATRIX DECOMPOSITION

Learning algorithms can be used to explore clusters of objects and features in data sets. This task has been traditionally performed with cluster analysis. Clustering algorithms tend to use all features while creating clusters, e.g., based on distances between objects (Kusiak 2000). Machine learning algorithms usually use a few features at a time.

Clustering algorithms support unsupervised learning, which implies that the objects in a data set are not assigned decision values. On the other hand, machine learning algorithms follow the principle of supervised learning and therefore decision values have to be specified.

In this chapter two ways of dealing with decision values in matrices are considered:

- Dummy object approach; All objects in a matrix (data set) are assigned identical decision values and a dummy object (row) is added to the matrix with a different value of the decision. The feature values of the dummy object should differentiate it from all other objects in the matrix.
- Filling undesirable entry approach; To increase discernity among objects of a sparse data set, random values are assigned to the blank entries.

The following types of decomposition are considered:

- Mutually separable matrix
- Non-decomposable matrix
- Matrix with conflicting outcomes

Mutually Separable Matrix

Consider the data in Table 1, which could represent a machine–part incidence matrix in the group technology problem. Matrix decomposition has applications beyond group technology and therefore the rows of the matrix in Table 1 are called objects and the columns are referred to as features. The entry ‘1’ in the object-feature matrix indicates that the feature is present and ‘blank’ denotes absence of the corresponding feature.

The matrix in Table 1 is sparse and blank entries have to be addressed before a learning algorithm is used. Data mining offers numerous algorithms for filling missing data. In this case it was assumed that a blank implies that a corresponding feature is not present rather than it is missing.

A. Kusiak, Non-Traditional Applications of Data Mining, in D. Braha (Ed.), *Data Mining for Design and Manufacturing*, Kluwer, Boston, MA, 2001, pp. 401-416.

Table 1. Object-feature incidence matrix

	F1	F2	F3	F4	F5	F6
1		1		1		
2			1			
3	1				1	
4			1			1
5		1		1		
6	1				1	
7					1	

As it is done in the group technology matrix, the blanks in Table 1 are replaced with zeros. In addition, dummy object 8 with decision value Two is introduced as shown in Table 2.

Table 2. Matrix with dummy object 8

	F1	F2	F3	F4	F5	F6	D
1	0	1	0	1	0	0	One
2	0	0	1	0	0	0	One
3	1	0	0	0	1	0	One
4	0	0	1	0	0	1	One
5	0	1	0	1	0	0	One
6	1	0	0	0	1	0	One
7	0	0	0	0	1	0	One
8	1	1	1	1	1	1	Two

A machine learning algorithm applied to the data set in Table 2 has produced the decision rules in Figure 1.

Rule 1. `(F6 = 0) THEN (D = One); [6, 85.71%, 100.00%][6, 0][1, 2, 3, 5, 6, 7]`
Rule 2. `(F5 = 0) THEN (D = One); [4, 57.14%, 100.00%][1, 2, 4, 5]`
Rule 3. `(F1 = 1) AND (F6 = 1) THEN (D = Two); [1, 100.00%, 100.00%][8]`

Figure 1. Decision rules extracted from the data set in Table 2

These decision rules in Figure 1 are presented in the following format: `(Condition) THEN (Outcome);` [Rule support, Relative rule strength, Discrimination level] [Objects represented by the rule] (For the definitions of these terms see Stefanowski 1998).

For example, Rule 3 `(F1 = 1) AND (F6 = 1) THEN (D = Two); [1, 100.00%, 100.00%] [8]`
reads

IF (The value of feature `F1` equals `1`) AND (The value of `F6` equals `1`) THEN (The decision `D` is `Two`); [This rule represents one object; In this case the one object makes up `100.00%` of all objects in the training data set with the decision `D` = `Two`; The two objects match the conditions and decision of this rule with the accuracy of `100%`] [The object represented by this rule is `8`].

Each of the three rules in Figure 1 group the objects as shown in Table 1. The matrix corresponding to Rule 1 is shown in Table 3. The cluster of objects {1, 2, 3, 5, 6, 7} shown in Table 3 has been created based on the value of feature F6 and is not interesting.

Table 3. Clusters corresponding to Rule 1 of Figure 1

	F1	F2	F3	F4	F5	**F6**
1		1		1		
2			1			
3	1				1	
5		1		1		
6	1				1	
7					1	
4			1			1

Rule 2 of Figure 1 has resulted in two clusters shown in Table 4.

The matrices in Tables 3 and 4 indicate that the clusters may be formed based on ‘0’ (previously ‘blank’) or ‘1’ entries as opposed to some clustering algorithms that create clusters based on the entries ‘1’ only. A way to ‘force’ the rules to select features with entries ‘1’ is to assign random numbers to all ‘blank’ entries.

Table 4. Clusters corresponding to Rule 2 of Figure 1

	F1	F2	F3	F4	**F5**	F6
1		1		1		
2			1			
4			1			1
5		1		1		
3	1				1	
6	1				1	
7					1	

The '0' entries of the incidence matrix in Table 3 have been filled with random integer numbers in the interval [3, 9] as shown in Table 5.

Table 5. The data of Table 3 with entries '0' replaced by random numbers

	F1	F2	F3	F4	F5	F6	D
1	4	1	3	1	8	5	One
2	3	0	1	2	2	4	One
3	1	0	4	5	1	2	One
4	6	0	1	8	3	1	One
5	7	1	5	1	4	3	One
6	1	0	6	9	1	8	One
7	8	0	7	4	1	5	One
8	1	1	1	1	1	1	Two

The rule set extracted from the data in Table 5 is shown in Figure 2.

Rule 4. (F2 = 0) THEN (D = One); [5, 71.43%, 100.00%][2, 3, 4, 6, 7]
Rule 5. (F1 = 4) THEN (D = One); [1, 14.29%, 100.00%][1]
Rule 6. (F1 = 7) THEN (D = One); [1, 14.29%, 100.00%][5]
Rule 7. (F1 = 1) AND (F6 = 1) THEN (D = Two); [1, 100.00%, 100.00%][8]

Figure 2. Rules extracted from the data in Table 5

The clusters corresponding to Rules 4 – 6 of Figure 2 are shown in Tables 6 and 7.

A. Kusiak, Non-Traditional Applications of Data Mining, in D. Braha (Ed.), *Data Mining for Design and Manufacturing*, Kluwer, Boston, MA, 2001, pp. 401-416.

Table 6. Clusters corresponding to Rule 4 of Figure 2

	F1	**F2**	F3	F4	F5	F6
1		1		1		
5		1		1		
2			1			
3	1				1	
4			1			1
6	1				1	
7					1	

Table 7. Clusters corresponding to Rules 5 and 6 of Figure 2

	F1	F5	F2	F4	F3	F6
3	1	1				
7		1				
6	1	1				
1	4		1	1		
5	7		1	1		
4					1	1
2					1	

Non-Decomposable Matrix

Most of the time a matrix does not decompose into mutually exclusive submatrices. Consider the matrix in Table 8 and its transformed form in Table 9 with '0s' replacing 'blanks', dummy row 9, and decision values {One, Two}.

Table 8. Matrix with eight objects

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
1		1	1	1				1		
2			1				1			
3	1				1					1
4					1					1
5		1		1						
6			1			1	1		1	
7					1					
8	1					1	1		1	

A. Kusiak, Non-Traditional Applications of Data Mining, in D. Braha (Ed.), *Data Mining for Design and Manufacturing*, Kluwer, Boston, MA, 2001, pp. 401-416.

Table 9. Expanded Table 8

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	D
1	0	1	1	1	0	0	0	1	0	0	One
2	0	0	1	0	0	0	1	0	0	0	One
3	1	0	0	0	1	0	0	0	0	1	One
4	0	0	0	0	1	0	0	0	0	1	One
5	0	1	0	1	0	0	0	0	0	0	One
6	0	0	1	0	0	1	1	0	1	0	One
7	0	0	0	0	1	0	0	0	0	0	One
8	1	0	0	0	0	1	1	0	1	0	One
9	1	1	1	1	1	1	1	1	1	1	Two

Rule 8. `(F8 = 0) THEN (D = One); [7, 87.50%, 100.00%][2, 3, 4, 5, 6, 7, 8]`
Rule 9. `(F5 = 0) THEN (D = One); [5, 62.50%, 100.00%][1, 2, 5, 6, 8]`
Rule 10. `(F1 = 1) AND (F8 = 1) THEN (D = Two); [1, 100.00%, 100.00%][0, 1][9}]`

Figure 3. Rules from the data in Table 9

The clusters corresponding to Rules 8 and 9 of Figure 3 are shown in Tables 10 and 11.

Table 10. Clusters associated with Rule 8 of Figure 3

	F5	F10	F1	F4	F2	**F8**	F3	F6	F7	F9
3	1	1	1							
7	1									
4	1	1								
1				1	1	1	1			
5				1	1					
6							1	1	1	1
2							1		1	
8			1					1	1	1

Table 11. Clusters associated with Rule 9 of Figure 3

	F5	F10	F1	F4	F2	F8	F3	F6	F7	F9
3	1	1	1							
7	1									
4	1	1								
1				1	1	1	1			
5				1	1					
6							1	1	1	1
2							1		1	
8			1					1	1	1

Clustering matrices based on the rules extracted with machine learning algorithms proves to be useful. The fact that machine learning algorithms work on different principles than traditional clustering algorithms may provide interesting insights into structure of matrices, especially when considered in a broad context of autonomous systems.

The next section will show that machine learning algorithms can be applied to structure matrices containing objects with conflicting (ill-defined) outcomes. In some applications, e.g., in group technology, the conflicting outcome approach is equivalent to grouping with alternative process plans (Kusiak 2000).

Matrix with Ill-Defined Outcomes

In the previous section, the exact rules were considered for matrix decomposition. The concept of approximate rules will be applied to discover objects with alternative outcomes. The approximate rules 5 involve two objects 4 and 9 as well as 5 and 10 with identical features and different outcomes.

Consider the data set in Table 12 and the corresponding rule set in Figure 4. As the set in Table 12 contains two objects with conflicting decisions, the rule set in Figure 4 contains approximate rules.

A. Kusiak, Non-Traditional Applications of Data Mining, in D. Braha (Ed.), *Data Mining for Design and Manufacturing*, Kluwer, Boston, MA, 2001, pp. 401-416.

Table 12. Matrix containing two objects 4 and 5 with conflicting decisions

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	D
1	0	1	1	1	0	0	0	1	0	0	Two
2	0	0	1	0	0	0	1	0	0	0	Three
3	1	0	0	0	1	0	0	0	0	1	One
4	0	0	0	0	1	0	0	0	0	1	One
5	0	1	0	1	0	0	0	0	0	0	Two
6	0	0	1	0	0	1	1	0	1	0	Three
7	0	0	0	0	1	0	0	0	0	0	One
8	1	0	0	0	0	1	1	0	1	0	Three
9	0	0	0	0	1	0	0	0	0	1	Two
10	0	1	0	1	0	0	0	0	0	0	Three

The exact and approximate rules extracted from the data in Table 12 are shown in Figure 4.

Exact rules

Rule 1. `(F1 = 1) AND (F9 = 0) THEN (D = One); [1, 33.33%, 100.00%][3]`
Rule 2. `(F4 = 0) AND (F7 = 0) AND (F10 = 0) THEN (D = One); [1, 33.33%, 100.00%][7]`
Rule 3. `(F8 = 1) THEN (D = Two); [1, 33.33%, 100.00%][1]`
Rule 4. `(F7 = 1) THEN (D = Three); [3, 75.00%, 100.00%[2, 6, 8]`

Approximate rules

Rule 5. `(F1 = 0) AND (F10 = 1) THEN (D = One) OR (D = Two); [2, 100.00%, 100.00%][4, 9]`
Rule 6. `(F3 = 0) AND (F4 = 1) THEN (D = Two) OR (D = Three); [2, 100.00%, 100.00%][5, 10]`

Figure 4. Exact and approximate rules for the data in Table 12

The exact rules of Figure 4 formed the clusters illustrated in Table 13 while the approximate rules discovered that objects (4 and 9) and (5 and 10) have conflicting outcomes.

Table 13. Clusters corresponding to the exact rules of Figure 4

	F5	F10	F1	F4	F2	F8	F3	F6	F7	F9
3	1	1	1							
7	1									
1				1	1	1	1			
6							1	1	1	1
2							1		1	
8			1					1	1	1
4	1	1								
5				1	1					

CLASSIFICATION ACCURACY ENHANCEMENT

The property of approximate rules discussed in the previous section will be applied to an industrial data set with ill-defined outcomes. It will be shown that approximate rules will identify objects in the training set that have been assigned incorrect outcomes. Rules are extracted from the same data set for two different feature sets.

Feature Set 1

Case 1: One approximate rule involving two conflicting objects

The quality of predictions with the rules extracted from a data set is usually evaluated by a cross-validation scheme (Stone 1974). The k-fold (here $k = 10$) cross-validation scheme is often recommended. In this scheme, a training data set is partitioned into $k = 10$ folds (subsets) and one fold of objects is removed from the training data set and the rules are extracted from the remaining $k - 1 = 9$ folds.

Table 14. Absolute classification accuracy

	N	Z	P	None
N	3	2	11	2
Z	3	11	12	1
P	2	3	30	6

The diagonal numbers in Table 14 represent the number of outcomes `D = Y, Z, P` that have been correctly predicted. In this case 3 of the 17 (= 3 + 2 +

11 + 2) decisions `D = N` have been correctly predicted along with 11 decisions `D = Z,` and 30 decisions `D = P`. The numbers off the diagonal indicate the incorrectly predicted decisions. For the decision `D = N` (row N in Table 14) 2 objects were incorrectly classified as `Z`, 11 objects as `P`, and two objects were placed in the 'None' category.
Table 15 reports the percentage of objects (known in data mining as classification accuracy) that have been classified correctly, incorrectly, or fall into the 'None' category for the three decision values `D = N, Z, P`.

Table 15. Classification accuracy

	Correct	Incorrect	None
N	12.50%	57.50%	10.00%
Z	50.83%	44.17%	5.00%
P	79.17%	10.69%	10.12%
Av	51.39%	38.33%	10.28%

Note that the numbers in each row of Table 15 do not necessarily add up to 100%.

Case 2: The outcome of one of the two conflicting object of Case 1 changed from `D = P` to `D = Z`

Table 16. Absolute classification accuracy

	N	Z	P	None
N	9	1	7	1
Z	3	11	10	4
P	2	3	33	2

Table 17. Classification accuracy

	Correct	Incorrect	None
N	36.37%	40.83%	2.50%
Z	50.00%	32.50%	17.50%
P	86.48%	9.60%	3.92%
Av	61.67%	30.00%	8.33%

Case 3: The outcomes of the two conflicting objects of Case 1 changed from `D = P` to `D = Z` and from `D = N` to `D = Z`.

Table 18. Absolute classification accuracy

	N	Z	P	None
N	3	3	8	3
Z	4	7	15	3
P	2	4	27	7

Table 19. Classification accuracy

	Correct	Incorrect	None
N	12.50%	47.50%	20.00%
Z	25.83%	56.67%	17.50%
P	73.38%	15.33%	11.29%
Av	43.06%	42.36%	14.58%

Case 4: The two conflicting objects of Case 1 removed from the training set

Table 20. Absolute classification accuracy

	N	Z	P	None
N	4	5	4	4
Z	1	7	10	9
P	1	5	24	10

Table 21. Classification accuracy

	Correct	Incorrect	None
N	21.67%	43.33%	15.00%
Z	25.00%	35.50%	39.50%
P	63.69%	12.62%	23.69%
Av	41.39%	31.25%	27.36%

Before the results of Cases 1 through 4 will be summarized classification accuracy for Feature Set 2 is computed. This data set contains features that differ from the ones in Feature Set 1.

Feature Set 2

Case 5: Training data set producing seven approximate rules

The support of each approximate rule varied from one to four objects.

Table 22. Absolute classification accuracy

	N	Z	P	None
N	4	1	10	3
Z	3	8	6	0
P	3	2	32	4

Table 23. Classification accuracy

	Correct	Incorrect	None
N	15.83%	49.17%	15.00%
Z	42.50%	57.50%	0.00%
P	84.45%	9.60%	5.95%
Av	51.39%	40.56%	8.05%

The outcomes of the 15 objects included in the approximate rules were arbitrarily modified `D = P` to `D = Z` and from `D = N` to `D = Z`.

Case 6: Training data set with 15 modified outcomes

Table 24. Absolute classification accuracy

	N	Z	P	None
N	1	5	6	2
Z	3	16	10	7
P	2	9	14	11

Table 25. Classification accuracy

	Correct	Incorrect	None
N	5.00%	65.00%	10.00%
Z	46.26%	27.90%	25.83%
P	45.83%	28.00%	26.17%
Av	36.39%	40.56%	23.05%

Rather than changing values of 15 objects, their removal from the training set could be considered. This action would certainly improve classification accuracy. However, the generality of the rules might be sacrificed.

Table 26 summarizes the correctly predicted outcomes of the six cases for two deferent feature sets.

Table 26. Summary of correctly predicted outcomes for cases 1 – 4 and 5 – 6

	Feature Set 1				Feature Set 2	
	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
N	3	9	3	4	4	1
Z	11	11	7	7	8	16
P	30	33	27	24	32	14

The most desirable values of absolute accuracy in Table 26 are highlighted. Case 2 of Feature Set 1 has produced the most desirable values of classification accuracy with 53 of 86 correctly predicted outcomes `D`. Case 6 with modified decision values for 15 objects resulted in improvement in the `D = Z` category.
It should be stressed that the modifications of decision values for all six cases were rather random and no attempt was made to identify assignments maximizing classification accuracy.

The average classification accuracy in Table 27 reinforces the above observations.

Table 27. Summary of average classification accuracy for cases 1 – 4 and 5 – 6

	Feature Set 1				Feature Set 2	
	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Correct	51.39%	61.67%	43.06%	41.39%	51.39%	36.39%
Incorrect	38.33%	30.00%	42.36%	31.25%	40.56%	40.56%
None	10.28%	8.33%	14.58%	27.36%	8.05%	23.05%

The results presented in Tables 28 and 29 present interesting insights into classification accuracy. The feature set of each training data set used in Cases 1 – 6 was expanded to the same base feature set. The results of the $k = 10$ fold cross-validation are reported in Tables 28 and 29.

Table 28. Summary of absolute classification accuracy of the data set with expanded features

	Case B1	Case B2	Case B3	Case B4	Case B5	Case B6
N	2	3	5	6	2	2
Z	9	7	11	10	9	15
P	29	29	19	25	29	14

Table 29. Summary of average classification accuracy of the data set with expanded features

	Feature Set 1				Feature Set 2	
	Case B1	Case B2	Case B3	Case B4	Case B5	Case B6
Correct	46.25%	45.28%	40.97%	48.61%	46.25%	36.25%
Incorrect	45.97%	35.28%	31.53%	31.25%	45.97%	31.39%
None	7.78%	19.44%	27.50%	20.14%	7.78%	32.36%

The highlighted cells in the two tables indicate the most preferred accuracy values. The computational results summarized in Tables 26 – 29 prove that changing values of ill-defined outcomes impacts classification accuracy. The classification accuracy can be controlled by changing values of objects with conflicting outcomes, eliminating some of these objects from the training data set, and using decision making rules extracted from different data sets, i.e., the rules corresponding the categories highlighted in Tables 26 – 29.

CONCLUSIONS

The interest in data mining is growing and its full potential is awaiting realization. While some applications of data mining have been discussed in the literature, many are still to come. In this chapter it was shown that machine learning algorithms could be applied to problems traditionally solved with operations research, mathematical programming, or cluster analysis tools. Machine learning may provide unknown insights into problem structure and discover other interesting properties. Besides discovering patterns, learning algorithms can be used to modify data sets in order to enhance decision making accuracy.

REFERENCES

Auer, P., R. Holte and W. Maass, “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, 1995.

Clark, P. and R. Boswell, "The CN2 induction algorithm," Machine Learning, 3(4), pp. 261-283, 1989.

Domingos, P. and M. Pazzani, "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, 1996.

Friedman, J., Y. Yun and R. Kohavi, "Lazy decision trees," Proceedings of the Thirteenth National Conference on Artificial Intelligence, AAAI Press and MIT Press, 1996.

Grzymala-Busse, J., "A new version of the rule induction system LERS," Fundamenta Informaticae, Vol. 31, pp. 27-39, 1997.

Kohavi, R., "Wrappers for Performance Enhancement and Oblivious Decision Graphs," Ph.D. Thesis, Computer Science Department, Stanford University, Stanford, CA, 1995.

Kusiak, A., "Computational Intelligence in Design and Manufacturing," John Wiley, New York, 2000.

Kusiak, A., "Engineering Design: Products, Processes, and Systems," Academic Press, San Diego, 1999.

Kusiak, A., "Decomposition in data mining; An industrial case study," IEEE Transactions on Electronics Packaging Manufacturing, 23(4), pp. 345-353, 2000a.

Kusiak, A., J.A. Kern, K.H. Kernstine, and T.L. Tseng, "Autonomous decision-making: A data mining approach," IEEE Transactions on Information Technology in Biomedicine, 4(4), pp. 274-284, 2000.

Michalski, R.S., I. Mozetic, J. Hong, and N. Lavrac, "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, 1986.

Michie, D., D.J. Spiegelhalter, and C.C. Taylor, "Machine Learning, Neural, and Statistical Classification," Ellis Horwood, New York, 1994.

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

Quinlan, J.R., "C4.5: Programs for Machine Learning," Morgan Kaufmann, Los Altos, CA, 1993.

Quinlan, J.R., "Induction of decision trees," Machine Learning, 1(1), pp. 81-106, 1986.

Stefanowski, J., "On rough set approaches to induction of decision rules," in L. Polkowski and A. Skowron (Eds), Rough Sets in Knowledge Discovery 1: Methodology and Applications, Springer-Verlag, New York, pp. 501-529, 1998.

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

A. Kusiak, Non-Traditional Applications of Data Mining, in D. Braha (Ed.), *Data Mining for Design and Manufacturing*, Kluwer, Boston, MA, 2001, pp. 401-416.