International Journal of Operations Research

International Journal of Operations Research Vol. 2, No. 2, 48–57 (2005)

Data Farming: A Primer

Andrew Kusiak[*]

Intelligent Systems Laboratory, Mechanical and Industrial Engineering, 2139 Seamans Center, The University of Iowa, Iowa City, Iowa 52242-1527

Abstract—A typical data mining project uses data collected for various purposes, ranging from routinely gathered data, to process improvement projects, and to data required for archival purposes. In some cases, the set of considered features might be large (a wide data set) and sufficient for extraction of knowledge. In other cases the data set might be narrow and insufficient to extract meaningful knowledge or the data may not even exist. Mining wide data sets has received attention in the literature, and many models and algorithms for feature selection have been developed for wide data sets. Determining features for which data should be collected in the absence of an existing data set or when a data set is partially available has not been sufficiently addressed in the literature. Yet, this issue is of paramount importance as the interest in data mining is growing. The methods and process for the definition of the most appropriate features for data collection, data transformation, data quality assessment, and data analysis are referred to as data farming. This paper outlines the elements of a data farming discipline.

Keywords—Data farming, Data mining, Feature definition, Feature functions, New features

1. INTRODUCTION

Data farming is concerned with methods and processes used to define the most appropriate features for data collection, data transformation, data quality assessment, and data analysis. The experience indicates that the magnitude of a data farming effort often outweighs the data mining task, especially in an industrial setting. This might be due to the fact that the industrial data is often collected for reasons other than decision-making. This data may involve a wide range of attributes (features) that go beyond traditional models. The lack of analysis tools, the limited awareness of data mining and data farming tools, and the cost reduction initiatives have contributed to scaling down some data collection efforts. Data farming mitigates this "loss" of information by enhancing the data on hand and determining the most relevant data that need to be collected.

Many data mining projects are based on data sets collected for various purposes, ranging from routinely collected data to process improvement projects and data required for archival purposes. In some cases, the set of considered features might be large (a wide data set) and sufficient for extraction of knowledge. In other cases the data set might be narrow and insufficient to extract meaningful knowledge, or the data may not even exist.

The mining of wide data sets has received the most attention in the literature. Numerous feature selection models and algorithms have been developed for such data sets. The feature selection methods can be divided into two classes:

- Open-loop methods. These methods are also called filter, preset bias, and front-end methods (Cios et al., 1998) Features are selected based on between-class separabilty criteria, e.g., covariance defined for different classes (Duda and Hart, 1973, and Fukunaga, 1990).
- Closed-loop methods. These methods are also referred to as wrapper, performance bias, and classifier feedback methods (John et al., 1994). Features are selected based on the performance criteria, e.g., classification accuracy.

Examples of methods for feature selection include the principle component analysis (Duda and Hart, 1973) and a branch-and-bound algorithm (Fukunaga, 1990). The feature selection problem is computationally complex as the total number of subsets for a set with n features is $2n$, and the number of subsets with m features is $n!/(n-m)!m!$ (Cios et al., 1998).

Determining the most appropriate features for which data should be collected in the absence of a data set or its partial availability (a narrow set) has not been sufficiently addressed in the literature. Yet, this issue is of paramount importance as the interest in data mining is growing. Feature selection and data farming cover the opposite ends of the data spectrum. The former deals with a redundant number of features and the latter begins with a potentially empty set of features that gradually leads to a set of features satisfying the selected performance criteria. Feature extraction supports a push approach to data mining as the selected features determine the quality of the extracted knowledge. On the other hand, data farming pulls the data necessary for knowledge extraction.

One of the goals of data farming is to define metrics capturing the quality of the data in terms of the performance criteria, e.g., the prediction accuracy. Some of these metrics are listed next.

Section 2 of this paper outlines elements of data farming methods. The data farming process is discussed in

[*] Corresponding author's email: andrew-kusiak@uiowa.edu

Section 3. The case study of Section 4 illustrates the benefits of data farming. Conclusions are drawn in Section 5.

2. DATA FARMING METHODS

The most important criteria of data farming are to define features that:

- Maximize performance measures (e.g., prediction accuracy, knowledge utility), and
- Minimize the data collection cost.

The two criteria directly translate into cost savings and other tangible or non-tangible benefits.

The basic methods of data farming are categorized as follows:

- Feature evaluation
- Data transformation
- Knowledge transformation
- Outcome definition
- Feature definition

Each of these data farming methods is discussed next.

2.1 Feature evaluation

The appropriateness of data sets for knowledge extraction 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, and so on.
- Other metrics such as percentage of missing values, data error, discretization parameters, and so on.

Cross-validation (Stone, 1974) is a widely used indirect method for feature evaluation. However, this method is computationally expensive, as it requires multiple uses of learning and decision-making algorithms (Vafaie and De Jong, 1998).

The feature evaluation metrics are of interest to data farming as they assess the quality of the data set without knowledge extraction, which is computationally complex. An ideal direct feature evaluation method should be able to determine whether a given data set will satisfy the required classification accuracy or any other performance measure without the repetitive knowledge extraction process.

2.2 Data Transformation

Data sets can be mined in their raw collected form or they can be transformed. The following transformation methods can be used:

- Filling in missing values
- Discretization
- Feature content modification (generalization, specialization)
- Feature transformation
- Data evolution

The data engineering methods are illustrated next. The first three data engineering methods have received some coverage in the data mining literature.

2.2.1 Filling in missing values

Examples of methods and algorithms for filling in missing values include:

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

Other methods for handling missing values are surveyed in Han and Kamber (2001).

2.2.2 Discretization

The most widely referenced discretization methods (also referred to as binning methods) are as follows:

- Equal width interval. The range of observed values is divided into k intervals of equal size. This method is vulnerable to outliers that may dramatically skew the value range, e.g., accidental typo of one value may significantly change the range.
- Equal frequency interval. The continuous values are grouped into k intervals with each interval containing m/k (possibly duplicated) adjacent values, where m is the number of examples. This method may lead to the inclusion of the same values in adjacent intervals. Both methods, the latter and the former fall into the category of unsupervised discretization methods as they do not consider decision values (Dugherty et al., 1995).
- Clustering. The intervals are created by clustering the examples (Tou and Gonzalez, 1974).
- Recursive minimal entropy. The intervals are established by considering the class information entropy (Carlett, 1991; Fayyad and Irani, 1993).
- Recursive minimal Gini index. Similar to the entropy, the Gini index characterizes the impurity of an arbitrary collection of examples (Breiman et al., 1984).
- Recursive minimal deviance: The deviance measure aims at selecting the best binary split (Venables and Ripley, 1998).

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

2.2.3 Feature content modification

The feature content modification method is illustrated with the data set in Figure 1, which is representative of

numerous data sets considered in medical and industrial applications.

Example 1

Consider the data set with five features and the decision shown in Figure 1.

Number	Index	Color	Material	Time	Temperature	Decision
1	TN-01	Blue	C-O-01	12	289.5	Good
2	NM-02	Red	C-R-30	78	333	Bad
3	NM-05	Orange	C-R-12	123	228	Bad
4	TN-04	Orange	C-O-02	15	321.7	Good
5	TN-14	Red	C-O-03	45	423	Good
6	NM-03	Red	C-R-11	77	630	Bad

Figure 1. A data set with five features**.**

A set of decision rules extracted from the data set in Figure 1 is shown in Figure 2.

Rule 1. IF (Index = TN-01) THEN (Quality = Good);
[1, 33.33%, 100.00%][1]
Rule 2. IF (Index = TN-04) THEN (Quality = Good);
[1, 33.33%, 100.00%][4]
Rule 3. IF (Index = TN-14) THEN (Quality = Good);
[1, 33.33%, 100.00%][5]
Rule 4. IF (Index = NM-02) THEN (Quality = Bad);
[1, 33.33%, 100.00%][2]
Rule 5. IF (Index = NM-05) THEN (Quality = Bad);
[1, 33.33%, 100.00%][3]
Rule 6. IF (Index = NM-03) THEN (Quality = Bad);
[1, 33.33%, 100.00%][6]

Figure 2. Rule set obtained from the data set in Figure 1.

The decision rules in Figure 2 are presented in the following format:

IF (Condition) THEN (Outcome); [Rule support, Relative rule strength, Confidence] [Objects represented by the rule].
The metrics characterizing each rule are defined next:

- *Rule support* is the number of all objects in the data set that share the property described by the conditions of the rule;
- *Rule strength* is the number of all objects in the data set that have the property described by the conditions and the action of the rule;
- *Relative rule strength* is the ratio of the rule strength and the number of all objects in a given class;
- *Confidence* is the ratio of the rule strength and the rule support.

The support of each rule in Figure 2 is only 1. These rules can be easily generalized by modifying the content of the feature "Index" in Figure 1 from TN-xx to TN and NM-xx to NM (see Figure 3).

Number	Index	Color	Material	Time	Temperature	Decision
1	TN	Blue	C-O-01	12	289.5	Good
2	NM	Red	C-R-30	78	333	Bad
3	NM	Orange	C-R-12	123	228	Bad
4	TN	Orange	C-O-02	15	321.7	Good
5	TN	Red	C-O-03	45	423	Good
6	NM	Red	C-R-11	77	630	Bad

Figure 3. Modified data set with five features.

The rules in Figure 4 have been extracted from the modified data set in Figure 3.

Rule 1:

IF (Index = TN) THEN (Quality = Good);

[3, 100.00%, 100.00%][1, 4, 5]

Rule 2:

IF (Index = NM) THEN (Quality = Bad);

[3, 100.00%, 100.00%][2, 3, 6]

Figure 4. Two rules generated from data set of Figure 3.

The feature generalization method is of interest to mining temporal data sets as the value of generalized features tend to be time invariant. The one-out-of n (n = 5) cross-validation scheme has been applied to the data sets in Figures 1 and 3. The results of cross-validation are presented in Figure 5.

As it is visible in Figures 5(c) and (d) the average classification accuracy for the data set in Figure 1 is 0% while for the modified data set of Figure 3 is 100%.

2.2.4 Feature transformation

Constructive induction is a process of describing objects for improved classification (Wnek and Michalski, 1994; 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).

In this paper, the data transformation aspect of constructive induction will be emphasized in order to improve usability, transparency, and the decision-making accuracy of the extracted rules.

While traditional data mining concentrates on establishing associations among feature values, temporal data farming is to determine the nature of feature behavior in time. In some cases, the temporal behavior of a singular feature might be difficult to capture and may not be appropriate for making predictions. Rather than

concentrating on individual features, the data mining approach presented in this paper advocates capturing relationships among feature functions.

(a)

	Good	Bad	None
Good	0	2	1
Bad	2	0	1

(b)

	Good	Bad	None
Good	3	0	0
Bad	0	3	0

(c)

	Correct	Incorrect	None
Good	0%	66.67%	33.33%
Bad	0%	66.67%	33.33%
Average	0%	66.67%	33.33%

(d)

	Correct	Incorrect	None
Good	100%	0%	0%
Bad	100%	0%	0%
Average	100%	0%	0%

Figure 5. Cross-validation results: (a) Confusion matrix for the data set in Figure 1, (b) Confusion matrix for the modified data set of Figure 3 cross-validation results: (c) Classification accuracy for the data set in Figure 1, (d) Classification accuracy for the data set of Figure 3.

Most data mining algorithms establish associations among individual feature values. The approach proposed in this paper captures relationships among features in the form of feature functions. Examples of feature functions include Kusiak (2001):

- 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 the sequence F2_F9_F4.
- Arithmetic expression of features Fi, Fj, ..., Fn, where the <arithmetic operator> = {+, −, /, ×, $\sqrt{\ }$, n, |}, e.g., F3 – 4.5 × F8, |F3 – 4.2 × F8|, (.7 × F23 – F4)/(2.1 × F52 + .2 × F8). Note that the inequality relation Fi ≥ Fj is equivalent to the ratio Fi/Fj ≥ 1.

A rule involving two feature functions, a sequence 5_7_2 (a set of features F2_F4_F9), and an inequality relation is shown next.

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

The feature transformation method is illustrated using Example 2. In this example the term classification quality will be used. Classification quality of a feature is a measure used in rough set theory to expresses the degree of association between the feature values and the outcome. It can be loosely defined as the number of objects with non-conflicting values to the total number of object in the data set. For a formal definition of the classification quality the reader may refer to Pawlak (1991).

Example 2

Consider the "as-is" data set in Figure 6.

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

Figure 6. A data set with four features.

The classification quality (CQ) of each feature in Figure 6 is as follows: CQ(F1) = 1/5 = 0.2, CQ(F2) = 2/5 = 0.4, CQ(F3) = 0/5 = 0, CQ(F4) = 3/5 = 0.6.

The data set in Figure 6 has been transformed in the data set of Figure 7, where two features F2, F4 have been replaced with the feature sequence F2_F4 denoted for short as F2_4.

No.	F1	F2_4	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

Figure 7. Transformed data set of Figure 6.

The classification quality of the feature sequence F2_4 has the value CQ(F2_4) = .6, which is higher than that of the individual features F2 and F4. The one-out-of n (n = 5) cross-validation scheme has been applied to the rules generated from the data sets in Figures 6 and 7. The cross-validation results of the original data set (Figure 6) and the transformed data set (Figure 7) are presented in Figure 8. The average classification accuracy has increased from 20% for the rules extracted from the data set in Figure 6 to 60% for the transformed data in Figure 7.

(a)

	Correct	Incorrect	None
Average	20%	60%	0%

(b)

	Correct	Incorrect	None
Average	60%	40%	0%

Figure 8. Cross-validation results: (a) Average classification accuracy for the data set in Figure 6, (b) Average classification accuracy for the transformed data set of Figure 7.

Example 2 illustrates one of many feature transformation methods involving sequences (sets) of features. The need for more elaborate feature transformations discussed earlier in this section leads to the evolutionary computation methods. Both, the feature transformation method and the previously discussed feature content modification method can be applied simultaneously. Moreover, numerous data farming methods can be combined for the same application.

2.2.5 Evolution

In a typical data mining process the knowledge is extracted from the historical data. The values of each feature can be described with various statistics, e.g., the probability density function as symbolically illustrated in Figure 9. The relationships between the features (columns F1–F4) themselves and the decision D can be also characterized by appropriate metrics, e.g., a correlation coefficient.

Rather than extracting knowledge from the original data a derived data set could be used. The latter data set could be created by using the statistical and other properties of the original data set. Changing the parameter values of these measures would evolve the source data and the extracted knowledge.

2.3 Knowledge transformation

Structuring knowledge may result in the discovery of patterns useful in understanding the knowledge content and may lead to its generalization. The need for knowledge structuring is supported by the notion of cognitive maps and mental models discussed in Caroll and Olson (1987) and Wickens et al. (1998). Structured decision rules are easier to evaluate by a user.

As an alternative to evolving the source data, the knowledge extracted from such data could be evolved.

One of the main reasons for extracting knowledge from data sets is decision-making – an area that has not received sufficient attention in the literature in the context of data mining. Most decision-making algorithms are rather simplistic and are usually based on partial or full matching schemes (Kusiak et al., 2000). Many users have difficulty accepting decision rules that are non-intuitive and algorithms making decisions based on non-transparent matching. Here we address a gap in the presentation of knowledge for effective decision-making.

The rule-structuring concept illustrated in Example 3 generates knowledge in a form that meets user expectations.

F1	F2	F3	F4	D
0.150	66	2	2	0
0.175	78	2	2	1
0.182	78	1	0	0
0.249	68	1	1	2
0.174	54	2	2	0

Figure 9. Data set and the corresponding statistical distributions.

Example 3

Consider the eight rules R1 – R8 represented as the rule-feature matrix in Figure 10.

F1_F6_F7	F2	F3	F4	F5	D	Rule	Algorithm
B_C_D	a				Low	R1	A1
C_ F		<4			Medium	R8	A1
			>2		Medium	R5	A1
			(2, 9]		Medium	R2	A2
		(2, 6]		=<8	High	R3	A2
			<2	(2, 5]	Low	R7	A3
E_ F_G	b				Low	R4	A3
		>=2		[1, 3]	High	R6	A3

Figure 10. Rule-feature matrix with eight rules.

Three different learning algorithms A1 – A3 were used to extract the eight decision rules R1 – R8 from a data set. To simplify our considerations the information pertinent to each rule such as support, classification quality, and so on has not been included. The first row (beside the header) in Figure 10 reads as follows: IF (F1_ F6_ F7 = B_ C_ D) AND (F2 = a) THEN (D = Low). The last entry of this row indicates that this rule has been derived with algorithm A1.

Though the rule set in Figure 10 is small, its analysis is not simple. Transforming the matrix in Figure 10 into the structured matrix in Figure 11 significantly improves interpretation and understanding of this rule set. Solving the model (1) – (5), presented later in this section, for the data in Figure 10 has resulted in the matrix of Figure 11. Two rules, R7 and R8, have been removed from the structured matrix as they are dissimilar to the rules R1

through R6.

F3	F5	F2	F1_F6_F7	F4	D	Rule	Algorithm
(2, 6]	=<8				High	R3	A2
>=2	[1, 3]				High	R6	A3
		a	B_C_D		Low	R1	A1
		b	E_ F_G		Low	R4	A3
				>2	Medium	R2	A2
				(2, 9]	Medium	R5	A1

Figure 11. Structured rule-feature matrix.

The content of the matrix in Figure 11 is structured and it allow drawing numerous conclusions, for example:

- The decisions D = High, Medium, and Low are totally separated by features, i.e., decision D = High is made based on values of two features F3 and F5 that do not appear in the other two decisions.
- The rules R3 and R6 are good candidates for the following generalization IF (F3 $\geq$ 2) AND (F5 $\leq$ 8) THEN (D = High).
- The decision D = Low can be reached in alternative ways, using the feature values F2 = a, or F1_F6_F7 = B_C_D, or F2 = b, or F1_F6_F7 = E_F_G.
- Rule R2 is more general than rule R5.

Example 3 illustrates only a few of the users' requirements that can be incorporated in the rule-structuring algorithm, such as:

- Classification accuracy. The knowledge included in the structured matrix is cross validated and tested to ensure the required level of classification accuracy.
- Matrix structure. To help the user better understand the rule-feature matrix, different structures may be considered, e.g., a block-diagonal (see Figure 11), a block-diagonal matrix with overlapping features, the block-diagonal matrix with overlapping rules, a triangular (for dependency analysis among rules), an L-shape matrix, a T-shape matrix, etc.
- Differentiation of decisions on features. This occurs when each decision value is associated with an independent subset of features.
- Differentiation of decisions on feature values. This occurs when any two decision values are discernable on a unique subset of feature values.
- Inclusion of user preferences. To increase confidence in the rules, a user may wish to have her/his feature preferences included in the selected rules, to exclude some features, to establish a lower bound on the number of features, and so on.
- Contrasting positive rules against negative ones.

The learning classifier systems (e.g., Wilson, 1995; Kovacs, 2001) and other learning concepts such as decision tree and decision rule algorithms are perceived as different. The former is based on concepts from evolutionary biology and the latter draws from information theory and mathematical logic. It appears that the two classes of algorithms share more commonality than indicated in the current literature. This unifying view results from the fact that the "machine learning school" assumes that the learning data set remains static. Filling in missing data, discretization, and feature content modification are the only three methods of data transformation. The two data transformation methods of data engineering discussed in this paper (i.e., feature transformation and data evolution) involve the evolutionary computation concepts. For example, a typical learning algorithm produces a decision rule as follows:

IF (F3 = 7) AND (F5 $\in$ [7.1, 12.4]) AND (F6 = 4)
THEN (D = No)

Each term in the above rule is concerned with a single feature.

The data and knowledge transformation concepts advocated in this paper lead to richer decision rules that may contain relationships between feature functions, in particular the feature sequence Seq illustrated by the following rule:

IF (F3 < F4) AND (F5/F8 $\geq$ 3) AND (Seq = f7_f9_f11)
THEN (D = No)

The computational experience presented in Cattral et al. (2001) indicates that the classification accuracy of the decision rules involving relationships between features exceeds those of the traditional decision rules.

To generate these robust and high quality results, the learning algorithms may remain essentially unchanged or in some cases require only minor modifications.

A user is interested in viewing the context of the knowledge used for decision-making from different perspectives, even if a decision is reached autonomously. The variability among knowledge viewing preferences grows with the number of users. Potential knowledge visualization patterns include a decision table, a decision rule, a decision tree, a graph, a bar chart, a pie chart, and a data cube that can be expressed with a Voronoi diagram, a Gabriel graph, Delaunay's approach, a relative neighborhood graph, a minimum spanning tree (Preparata and Shamos, 1985). Note that one pattern, e.g., a decision tree, can be transformed into another pattern, e.g., a decision table. The decision table provides a multitude of knowledge visualization patterns (views) such as:

- Rule – feature view (see Figure 11).
- Rule – feature function view.
- Object – feature view.
- Cluster of object – feature view.
- Cluster of object – group of features view.
- Rule – rule view.
- Chain of rules view (for multistage decision processes).
- Rule performance metric (e.g., rule strength, confidence, discrimination) view.

The rules themselves can be illustrated graphically. One possible representation of a cluster of two rules is shown in Figure 12.

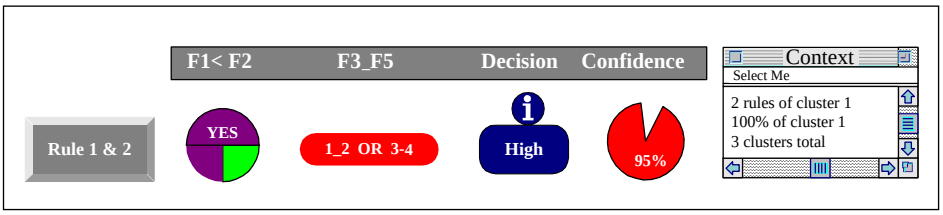

Figure 12. Visual representation of a cluster of two rules.

The above views call for knowledge structuring to be accomplished by solving various models. One of such models, the generalized p–median model, is illustrated next (Kusiak, 2002).

Define:

n = the total number of decision rules
m = the number of features (used to compute the distance d_{ij})
l = the number of rule categories
F_k = the set of decision rules from source k, k = 1, ..., l, where $|\cap F_k| = n$
p = the minimum number of rule clusters k
q_k = the minimum number of rules to be selected for rule category k
d_{ij} = the distance between rules i and j
c_j = the performance index of rule j
α, β = constants used in the objective function
x_{ij} = 1, if rules i and j are selected, otherwise $x_{ij} = 0$
x_j = 1, if rules j is selected, otherwise $x_j = 0$

The objective of the generalized p–median model is to minimize the total weighted distance between the rules and the rule performance index. The two constants α and β are used as the weights.

$$Min \quad \alpha \sum_i \sum_j d_{ij} x_{ij} + \beta \sum_j c_j x_{jj} \tag{1}$$

$$s.t. \quad \sum_{i \in F_k} \sum_j x_{ij} \geq q_k \quad \text{for all } k = 1,...,l,\ j = 1,...,n \tag{2}$$

$$\sum_j x_{jj} \geq p \quad \text{for all } j = 1,...,n \tag{3}$$

$$x_{ij} \leq x_{jj} \quad \text{for all } i = 1,...,n \ \ j = 1,...,n \tag{4}$$

$$x_{ij} = 0,\ 1 \quad \text{for all } i = 1,...,n \ \ j = 1,...,n \tag{5}$$

Constraint (2) ensures that for each rule category at least q_k rules are selected. Constraint (3) imposes a lower bound on the number of rule clusters. Constraint (4) ensures that a pair of rules, i and j, can be selected only when the corresponding cluster is formed. Constraint (5) imposes the integrality of the decision variable.

The input to the p–median model is a set of rules of different categories. For example, a rule category can be based on the learning algorithm type, decision type, rule type (positive, negative, etc.), feature transformation method, and so on.

Solving the generalized p–median model for the data in Figure 10 has resulted in the structured matrix in Figure 11. The p–median model has been solved with the LINDO software (LINDO, 2003).

2.4 Outcome definition

Some outcomes may be either not defined or assigned in error, e.g., misclassified by one or two classes. For unsupervised learning (not defined outcomes), clustering methods can be applied to define and validate the outcome values. For cases with the outcomes assigned in error, unsupervised and supervised learning may be warranted. An outcome definition method is illustrated in Example 4.

Example 4

Consider the data in Figure 13 with five features (e.g., maximum torque, temperature, and number of hours and corresponding rules (Figure 14) derived using a rough set algorithm).

No.	F1	F2	F3	F4	F5	D
1	0	0	1	0	2	0
2	2	1	1	0	2	1
3	0	0	0	0	1	0
4	1	0	1	1	0	1
5	0	0	0	1	3	0

Figure 13. A data set with five features.

Rule 1:
IF (F1 = 0) THEN (D = 0);
[3, 100.00%, 100.00%][1, 3, 5]

Rule 2:
IF (F1 ∈ {1, 2}) THEN (D = 1);
[2, 100.00%, 100.00%][2, 4]

Figure 14. Rules from the data set of Figure 13.

Assume that some values of the outcome D in Figure 13 were assigned in error, e.g., the fault type was improperly coded. The analysis of the data in Figure 13 and other background information has lead to changing the value of the decision for object 2 from D = 1 to D = 2. The rules extracted from the data set with the modified outcome are shown in Figure 15.

Rule 3:
IF (F1 = 0) THEN (D = 0);
[3, 100.00%, 100.00%][1, 3, 5]
Rule 4:
IF (F1 = 1) THEN (D = 1);
[1, 100.00%, 100.00%][4]
Rule 5:
IF (F1 = 2) THEN (D = 2);
[1, 100.00%, 100.00%][2]

Figure 15. Rules extracted from the transformed data set of Figure 13.

The one-out-*n* (*n* = 5 objects) cross-validation scheme was applied to the data set in Figure 13 and its transformed form. The results of cross-validation are shown in Figure 16.

(a)

	Correct	Incorrect	None
Average	40%	0%	60%

(b)

	Correct	Incorrect	None
Average	60%	0%	40%

Figure 16. Cross-validation results: (a) average classification accuracy for the data set in Figure 13; (b) average classification accuracy of the data set with modified outcome.

Figure 16, parts (a) and (b), shows that the data set with modified outcome provided better classification accuracy than the source data set of Figure 13.

2.5 Feature definition

The previously discussed data farming methods enhance the data and knowledge of an existing set of features. The feature definition approach is concerned with the definition of new features for which the data must be collected. In this setting, the maximization of the performance of the extracted knowledge and the minimization of the data collection cost are extremely important.

Methods of defining candidate features include:
- Parameters and variables of equations
- Object and environment descriptions
- Design of experiments

The ultimate goal of data farming is to extract useful knowledge that represents associations among features and decisions. Science offers expressions (e.g., reliability formula) that might be helpful in the definition of new features. In some cases it may be useful to map the associations as a feature map that is similar to a process model created via any formal methodology, e.g., the process methodology presented in (Kruchten et al., 2000).

The newly defined features should allow for building strong associations with the outcome. The data mining experience with numerous data sets indicates that the most promising feature types include:
- Chemistry based features (e.g., calcium content)
- Biology based features (e.g., genetics)
- Time and frequency (e.g., number of years in use, number of tasks performed)
- Control parameters (e.g., temperature)

3. THE DATA FARMING PROCESS

Data mining applications call for the definition of appropriate features and data collection at minimal cost. The data farming process includes the following steps:

Step 1. Setting a data farming goal.
Step 2. Definition of candidate features and dependency analysis (discussed later in this section).
Step 3. Selection and application of suitable data farming methods.
Step 4. Data mining process.
Step 5. Evaluation of the data farming goal.

These steps can be implemented sequentially or in parallel.

Step 2 above has not been discussed thus far. It involves determining a candidate set of features and the identification of the dependencies between them. These features may be targeted for data collection. Dependencies among features, though not absolutely essential in data farming, may be important for understanding the data set. Some of the methods for feature dependency analysis are discussed in the following paragraphs.

Numerous methods and tools have been developed for the analysis of systems. The primary methods that can be used for feature dependency recording and analysis are as follows:
- Feature map. For example, a graph showing relationships between features that may take different forms, e.g., a fish bone diagram used in statistics and a link analysis graph (Barry and Linoff, 1997).
- Structure breakdown methods for a problem, a process, or a product. For example, a diagram visualizing hierarchical representation of the studied phenomenon (a problem, a process, or a product).
- Process models. Numerous process models and tools developed in the context of process engineering can be used to analyze dependencies among features (see

Kusiak (1999) for review of process modeling tools and methods).

- Semantic networks. Though primarily used to represent knowledge, semantic networks can be applied to represent concepts, including the relationships between features.

In addition to the above tools, methodologies and software used to model information systems or processes can be used to analyze dependencies among features, e.g., Yourdon, Gane-Sarson, Express-G, and Rumbaugh diagrams.

The dependency among features can be analyzed in two modes:

- Forward mode (feature – decision direction)
- Reverse mode (decision – feature direction)

Most often the two modes are combined while analyzing features.

The type of data farming method used depends on the purpose of data mining. The main purposes of data mining are as follows:

- Gaining insights into the problem studied
- Learning
- Decision-making with the discovered knowledge

While the first purpose may be accomplished with a rather small data set, the last two call for sufficiently wide and long data sets that are typically obtained at a higher cost.

4. A CASE STUDY

Some of the data farming concepts discussed in this paper have been applied to an equipment diagnosis application. A database with the maintenance records in excess of 1 GB was available for analysis. The data collected on more than 300 features was distributed over 26 different Microsoft Access tables. The number of objects in each table varied from twenty to about 100,000. The goal of the study was to predict the duration MAINT_TIME of a service action performed on different types of equipment. The analysis of the data with dependency diagrams has revealed that most features were irrelevant to the study as the data was collected over many years to meet operational requirements imposed over time.

For the purpose of prediction of MAINT_TIME a data set with 644 objects was extracted by merging three data tables with maintenance related features. The length of the shortest data set dictated the number of objects. Some objects were missing most feature values and therefore the number of objects was reduced to 599 and 17 features (15 categorical and integer, and two continuous) and the decision MAINT_TIME which was continuous. The data set was discretized with three intervals resulting in the following classes (1, 2, and 3) for the decision MAINT_TIME:

MAINT_TIME: (< 0.25) ~ 1, [0.25, 0.35) ~ 2, [> 0.35) ~ 3,

where (< 0.25) ~ 0 means that the maintenance time of less than .25 [hour] was labeled as category 0, the maintenance time in the interval [0.25, 0.35) was labeled as category 1, and the maintenance time greater than 0.35 [hour] was labeled as category 2.

Different learning algorithms have been applied to extract rules and the number of rules was generally large. A rough set algorithm (Pawlak 1982, 1991) produced some of the most interesting rules (102 exact and 19 approximate rules).

The $k = 10$ fold validation with the rough set algorithm has produced the results in Figure 17, which are encouraging considering the nature of the data considered in this study.

	Correct	Incorrect	None
Average	68.45%	31.38%	0.17%

Figure 17. Average classification accuracy for the 599-object data set.

Further analysis of the 599-object data set has revealed that some maintenance actions involved multiple elementary actions. The results in Figure 17 include both types of actions. The MAINT_TIME for multiple maintenance actions in the 599-object data set were aggregated thus resulting in 525 objects.

The 525-object data set was discretized with the previously used scheme and produced the cross validation results in Figure 18. These results are not substantially different from the results in Figure 17.

	Correct	Incorrect	None
Average	69.19%	30.61%	0.19%

Figure 18. Average classification accuracy for the 525-object data set.

Creating sequences of two variables at a time produced some of the most interesting results. Two such results are illustrated in Figure 19 for the following MAINT_TIME discretization scheme:

MAINT_TIME: (<0.25) ~ 0, [0.25, 4.75) ~ 1, [>4.75) ~ 2

	Correct	Incorrect	None
Average	79.52%	20.48%	0.00%

Figure 19. Average classification accuracy for the 525-object data set with the feature sequence.

Please also notice that the results in Figure 19 indicate that the average classification accuracy is much better than that in Figure 18.

The main goal of this case study was to prove that the maintenance data collected during regular operations contained some useful patterns that could be used to predict values of parameters, including the maintenance time. Some of the data farming methods applied in this study enhanced the value of this data set. Extensions of the classification quality measure and some statistical metrics will be used to define more relevant features for which the data should be collected.

5. CONCLUSION

The purpose of knowledge discovery is to gain insights into the problem studied, or using the discovered knowledge for decision-making. These objectives can be realized, if proper data is collected. The appropriateness of data and the data collection cost are the goals of data farming that, among others, offers tools for the definition of appropriate features for which the data is to be collected at an acceptable cost. The data farming methods presented in this paper are intended to enhance the data collection process, add value to the collected data, and define new features for which the data should be collected. The data farming concepts presented in this paper were illustrated with numerical examples and a case study.

REFERENCES

1. Barry, M.J.A. and G. Linoff (1997). *Data Mining Techniques: For Marketing, Sales, and Customer Support*. John Wiley, New York.
2. Bloedorn, E. and R.S. Michalski (1998). Data-driven constructive induction. *IEEE Intelligent Systems*, 13(2): 30-37.
3. Breiman, L., J.H. Friedman, R.A. Olshen, and P.J. Stone (1984). *Classification and Regression Trees*. Wadworth International Group, Belmont, CA.
4. Carlett, J. (1991). *Megainduction: Machine Learning on Very Large Databases*. Ph.D. Thesis, Department of Computer Science, University of Sydney, Australia.
5. 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.
6. 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.
7. Cios, K., W. Pedrycz, and R. Swiniarski (1998). *Data Mining: Methods for Knowledge Discovery*. Kluwer, Boston, MA.
8. 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.
9. Duda, R.O. and P.E. Hart (1973). *Pattern Recognition and Scene Analysis*. John Wiley, New York.
10. 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.
11. Fukunaga, K. (1990). *Introduction to Statistical Pattern Analysis*. Academic Press, San Diego, CA.
12. Han, J. and M. Kamber (2001). *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Diego, CA.
13. John, G., R. Kohavi, and K. Pfleger (1994). Irrelevant features and the subset selection problem, *Proceedings of the 11th International Conference on Machine Learning, ICLM'94*. Morgan Kaufmann, San Diego, CA, pp. 121-127.
14. Kruchten, P. (2000). *The Rational Unified Process: An Introduction*. Addison-Wesley, New York.
15. Kovacs, T. (2001). What should a classifier system learn. *Proceedings of the 2001 Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, pp. 775-782.
16. Kusiak, A. (1999). *Engineering Design: Products, Processes, and Systems*. Academic Press, San Diego, CA.
17. Kusiak, A. (2000). Decomposition in data mining: an industrial case study. *IEEE Transactions on Electronics Packaging Manufacturing*, 23(4): 345-353.
18. 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*, 4(4): 274-284.
19. Kusiak, A. (2001). Feature transformation methods in data mining. *IEEE Transactions on Electronics Packaging Manufacturing*, 24(3): 214-221.
20. Kusiak, A. (2002). A Data mining approach for generation of control signatures. *ASME Transactions*: *Journal of Manufacturing Science and Engineering*, 24(4): 923-926.
21. LINDO (2003). http://www.lindo.com (Accessed June 5, 2003).
22. Pawlak Z. (1982). Rough sets. *International Journal of Information and Computer Science*, 11(5): 341-356.
23. Pawlak, Z. (1991). *Rough Sets: Theoretical Aspects of Reasoning About Data*. Kluwer, Boston, MA.
24. Preparata, F.P. and Shamos, M.I. (1985). *Pattern Recognition and Scene Analysis*. Springer-Verlag, New York.
25. Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning*, 1(1): 81-106.
26. Ragel, A. and B. Cremilleux (1998). Treatment of missing values for association rules. *Proceedings of the Second Pacific Asia Conference*, PAKDD'98, Melbourne, Australia.
27. Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, 36: 111-147.
28. Slowinski, R. (1993). Rough set learning of preferential attitude in multi-criteria decision making, in Komorowski, J. and Ras, Z. (Eds), *Methodologies for Intelligent Systems*, Springer-Verlag, Berlin, Germany, pp. 642-651.
29. Tou, J.T. and R.C. Gonzalez (1974). P*attern Recognition Principles*, Addison Wesley, New York.
30. Vafaie, H. and K. De Jong (1998). Feature space transformation using genetic algorithms. *IEEE Intelligent Systems*, 13(2): 57-65.
31. Venables, W.N. and B.D. Ripley (1998). *Modern Statistics with S-PLUS*. Springer-Verlag, New York.
32. Wickens, G., S.E. Gordon, and Y. Liu (1998). *An Introduction to Human Factors Engineering*. Harper Collins, New York.
33. Wilson, S.W. (1995). Classifier fitness based on accuracy. *Evolutionary Computation*, 3(2): 149-175.
34. Wnek, J. and R.S. Michalski (1994). Hypothesis-driven constructive induction in AQ17-HCI: A method and experiments. *Machine Learning*, 14(2): 139-168.
35. Yang, J. and V. Honavar (1998). Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems*, 13(2): 44-49.