In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '98):
Intelligent Systems for Man in a Cyberworld, pp. 2882-2887, San Diego, CA. October 11-14, IEEE Computer Society Press, 1998

A Methodology and Life Cycle Model for Data Mining and Knowledge Discovery in Precision Agriculture

Seok Won Lee
Learning Agents Laboratory
Dept. of Computer Science
George Mason University
Fairfax, VA 22030-4444 USA

Larry Kerschberg
Center for Information Systems Integration and Evolution
Dept. of Information and Software Systems Engineering
George Mason University
Fairfax, VA 22030-4444 USA

ABSTRACT

This paper presents a methodology for data mining and knowledge discovery in large, distributed and heterogeneous databases. In order to obtain potentially interesting patterns, relationships, and rules from such large and heterogeneous data collections, it is essential that a methodology be developed to take advantage of the suite of existing methods and tools available for data mining and knowledge discovery in databases (KDD). One of the most important methodologies is an integration of diverse learning strategies that cooperatively performs a variety of discovery techniques that achieves high quality knowledge. KDLC is an extended study of AqBC [8] which is a multistrategy knowledge discovery approach that combines supervised inductive rule learning and unsupervised Bayesian classification via constructive induction mechanism. A case study dealing with "crop yields" for a farm in the state of Idaho is presented and preliminary results are visualized by using ArcView GIS system. The significance of the multistrategy knowledge discovery process and visualization process in analyzing the classifications and learned rules has been empirically verified in KDLC.

1. INTRODUCTION

This paper presents a methodology for data mining and knowledge discovery in large, distributed and heterogeneous databases. The data collections may consist of maps, imagery, and sensor data, and in situ measurements that must be integrated to present a composite "picture" of the information to be analyzed. In order to obtain potentially interesting patterns, relationships, and rules from such large and heterogeneous data collections, it is essential that a methodology be developed to take advantage of the suite of existing methods and tools available for data mining and knowledge discovery in databases (KDD) [2].

We develop a methodology and an associated *Knowledge Discovery Life Cycle* model, called the *KDLC*. The KDLC consists of well-defined activities that guide and assist the user throughout the KDD process. The learning processes of the KDLC are a combination of unsupervised and supervised learning. A key contribution of this approach is the integration of these two types of learning into a multistrategy KDD approach, combining unsupervised Bayesian classification with supervised inductive learning. The particular approach begins by analyzing large data sets using an unsupervised Bayesian classification system, AutoClass. AutoClass discovers interesting taxonomic classes from databases, and these can be represented as new attributes in an expanded representation space via Constructive Induction mechanism. The robust inductive learning system, AQ15c, can then be used to "learn" useful concepts, relationships, and rules that characterize knowledge in the data space.

A case study dealing with "crop yields" for a farm in the state of Idaho is presented and preliminary results of the data mining and knowledge discovery process using the KDLC are presented. ArcView GIS system allowed us to visualize the knowledge from KDLC onto the geographic regions of the Idaho property. The significance of the knowledge visualization process in analyzing the classifications and learned rules has been empirically verified in KDLC.

2. KNOWLEDGE DISCOVERY IN DATABASES PROCESS MODEL

This section provides an introduction to the methods, processes, and methodologies being developed. In particular, we believe that a Knowledge Discovery Life Cycle is needed to allow domain experts and knowledge engineers to work together to harvest truly breakthrough knowledge for large collections of data.

2.1. Knowledge Discovery Life Cycle Model

The developed methodology for data mining and knowledge discovery, called the Knowledge Discovery Life Cycle (KDLC), is a closed-loop iterative process model consisting of six major activities as shown in Figure 1. These activities are explained in section 2.2. Note that each activity may access multiple types of data and knowledge stored in a locally-curated Information Repository, distributed heterogeneous on-line databases, data warehouses, and external sources that may provide in situ measurements, etc.

2.2. KDLC Activities

This section illustrates a description of the various activities that comprise the KDLC. Each activity is important to the success of a data mining and knowledge discovery session.

- **Plan for Learning**: In order to actually discover knowledge, one must plan a set of experiments and formulate a set of hypotheses. The data must also be prepared for the data mining and discovery process. This involves the cleansing of data, ensuring the quality of the data, integrating data from multiple sources, etc.

- **Generate and Test Hypothesis**: This activity involves exploratory analysis, concept formulation, pattern

definition, and template specification via user queries. This allows hypotheses to be tested by actually analyzing the data.

- **Discover Knowledge**: In this phase learning algorithms are selected and tailored for the learning task. These are used to discover knowledge and to transform it into human-understandable structures such as rules and decision trees.

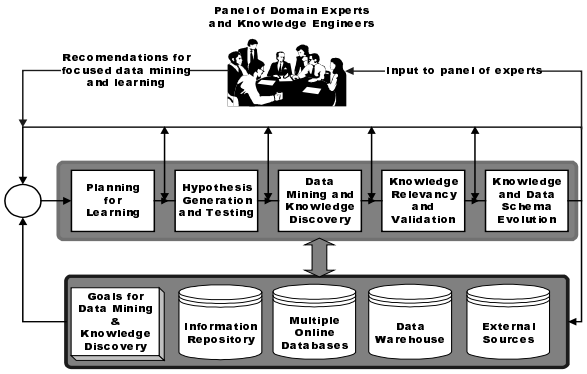


Figure 1. The Knowledge Discovery Life Cycle Model

- **Determine Knowledge Relevancy**: Here the discovered knowledge is assessed in terms of its relevancy, coverage of test cases, and usefulness for the problem at hand. Knowledge visualization, explanation, and validation are performed in this phase.

- **Evolve Knowledge/Data**: Once discovered knowledge is accepted into the Information Repository, it will have possible impacts on the current database schema, on the information contained in the data warehouse, and on the knowledge base. This activity is concerned with the evolution of the combined data/knowledge bases. Also, the knowledge lineage, or derivation of the knowledge, is important because the knowledge was "discovered" from specific versions of data, domain knowledge, learning algorithms, etc. This meta-information must also be stored and maintained in the information repository.

- **Critique by a Panel of Experts**: In order to assess the true meaning of a relevance of discovered patterns and rules, these must be examined by a panel of experts who confer with the Knowledge Engineers performing the actual experiments and running the learning and classification algorithms. Without this *ongoing feedback from experts*, the engineers will not have enough information to obtain truly breakthrough and valuable knowledge.

3. MULTISTRATEGY APPROACHES TO LEARNING AND DISCOVERY

The following sections present each of learning and knowledge discovery techniques used as parts of the learning and knowledge discovery process in KDLC.

3.1. AutoClass: An Unsupervised Bayesian Classifier

AutoClass is an unsupervised Bayesian classification system that looks for a maximum posterior probability classification [1]. The system infers classes based on Bayesian statistics, deriving a belief network via probability theory.
The idea of accumulating and applying auxiliary evidence here can be mapped into the constructive induction mechanism that employs a new attribute, which summarizes the data patterns. The new attribute's degree of belief is very high because it is generated from the best model of Bayesian classification. Therefore, this new attribute can potentially reorganize and improve the knowledge representation space. The theory of how Bayesian learning is applied in AutoClass, summarized from [3], is described below.
Let E denote a set of *evidence* and H a set of possible *hypotheses* that can conceptualize a set of combinations in E. Assume that the sets of possible evidence E and possible hypotheses H are mutually exclusive power sets. $P(ab|cd)$ represents a real number probability describing a *degree of belief in the conjunction of propositions a* and *b* conditioned on the assumption that the given propositions *c* and *d* are true. Let $\pi(H|E)$ denote a *posterior probability* describing a belief in H after observing evidence E. Let $L(E|H)$ denote a *likelihood* containing a theory of how likely it would be to see each possible evidence combination E in each possible set of hypotheses H. Beliefs are non-negative ($0 \le P(a|b) \le 1$) and normalized ($\sum_H \pi(H) = 1$ and $\sum_E L(E|H) = 1$). The combination of likelihood and priors produces a *joint probability* $J(EH) \equiv L(E|H)\pi(H)$ of both E and H. Bayes's rule shows how beliefs should change when evidence is obtained by normalizing the joint probability.

$$\pi(H|E) = \frac{J(EH)}{\sum_H J(EH)} = \frac{L(E|H)\pi(H)}{\sum_H L(E|H)\pi(H)}$$

A possible set of hypotheses H with an associated likelihood function in any given situation indicates what evidence we expect to observe when the chosen set of hypotheses is true. A set of prior probabilities corresponding to this set of hypotheses should be obtained with a set of evidence. Bayes's rule then specifies the appropriate posterior beliefs about the hypothesis. These posterior probabilities and a *utility over hypotheses*, *U(H)*, which describes preferences for each individual hypothesis, can thus be combined to obtain the maximum expected utility:

$$EU(A) = \sum_H U(H)\pi(H|EA)$$

AutoClass discovers various kinds of knowledge including classifications and causal mechanisms between cases.

3.2. AQ15c Inductive Learning System

AQ15c is a C-language reimplementation of AQ15 with significant performance improvement and dynamic allocation of representation spaces. The AQ family of inductive learning programs implements the STAR method of inductive learning. AQ15c learns decision rules for a given set of decision classes from examples. When learning rules, AQ15c uses 1) background knowledge in the form of rules (input hypotheses),

2) the definition of descriptors and their types and 3) a rule preference criterion that evaluates competing candidate hypotheses. Each training example characterizes an object, and its class label specifies the correct decision associated with that object. The generated decision rules are expressed as symbolic descriptions involving relations between objects' attribute values. The program performs a heuristic search through a space of logical expressions, until it finds a decision rule that best satisfies the preference criterion while covering all positive examples, but no negative examples. Each decision rule is described by one or more conditions, all of which must be met for the rule to apply. A *condition* is a relational statement. A *rule* is a conjunction of conditions. A *hypothesis* is a disjunction of rules that together describe a concept. A hypothesis is satisfied if any of its rules are satisfied, while a rule is satisfied if all of its conditions are satisfied. A condition is satisfied if the term takes one of the values in the reference.

3.3. A Multistrategy Knowledge Discovery Approach

This section motivates the methodology and approach used in this study. The concept of constructive induction is introduced to show that a transformation of the problem's representation space may lead to more efficient and effective data mining and KDD results. The KDLC uses a multistrategy knowledge discovery approach that combines unsupervised Bayesian classification with supervised inductive learning. Figure 2 shows the general structure of the KDLC approach. KDLC can be applied to two different goals. First, AutoClass provides classifications that help the user generate "expert knowledge" for a potential target concept when a data set without predefined classifications is given to the supervised inductive learning system. The inductive learning system AQ15c learns the concept descriptions of these classifications (step 3 in Fig. 2). Second, when the data set is already divided into classes, KDLC repeatedly searches for the best classifications using AutoClass, then uses the best one for creating and modifying more suitable and meaningful knowledge representation spaces. Figure 3 shows the constructive induction process in which a new attribute is created for training and testing. In the first phase, a new attribute, in which the values are class labels generated by AutoClass, is added to the data table under the name of *CA* (Step 1 & 2 in Fig. 2). Note that one of the attributes provided to AutoClass is the original target concept *C* (Fig. 3-a). However, *C* is not included when we are dealing with the testing data set (Fig. 3-b) since the system is not supposed to know the given class. The modified data table generated by this phase, which now includes *CA*, is used by AQ15c to learn the new concept description for *C*. In other words, the concept description of the original classification is learned from the modified knowledge representation space.

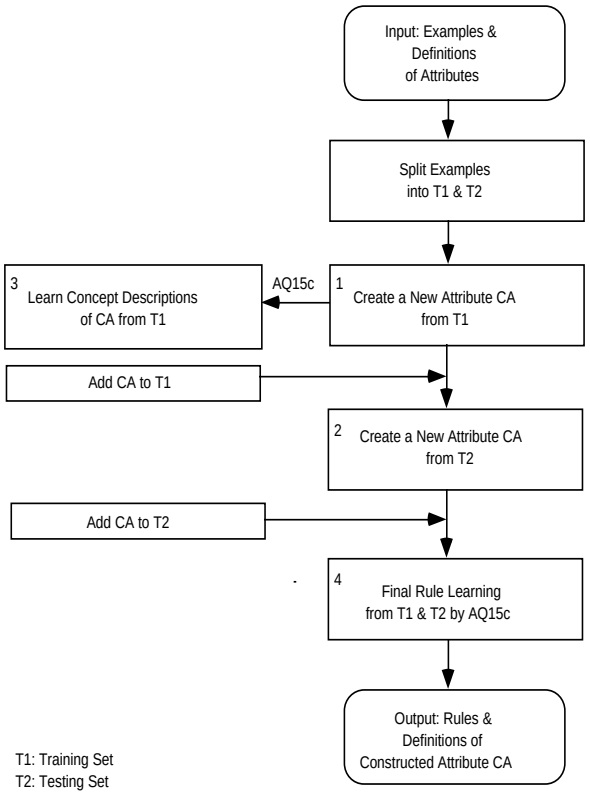
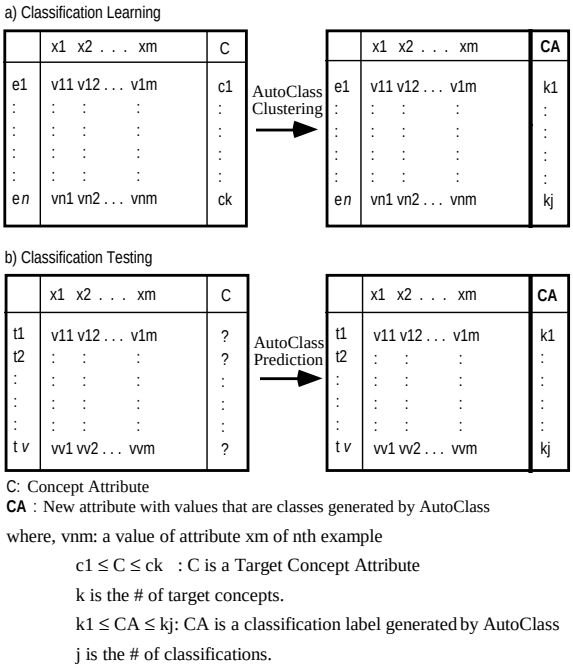


Figure 2. General Structure of KDLC Multistrategy Data Mining and Knowledge Discovery Approach



C: Concept Attribute
CA : New attribute with values that are classes generated by AutoClass

where, vnm: a value of attribute xm of nth example

$c1 \le C \le ck$: C is a Target Concept Attribute

k is the # of target concepts.

$k1 \le CA \le kj$: CA is a classification label generated by AutoClass

j is the # of classifications.

Figure 3. Creating a New Attribute for a) Training and b) Testing Data Set using AutoClass.

The only change is the augmentation of the table with the new attribute. It is at this time that a separate AQ15c run generates the concept descriptions for the classes represented by *CA*.

In the second phase, the testing data set, an unseen data set with the original concept attribute *C* containing the value "?" (don't care) for all examples, is input to the AutoClass classifier trained in the first phase (see Figure 3b). By not

providing the original classifications, the original properties of the testing data set can be preserved and AutoClass's ability to correctly classify previously unseen examples can be verified. The technique of constructing classification labels is based on constructive induction methodology [13].

The improvement of the knowledge representation space can potentially allow us to address large, complicated data sets not recognized well by a supervised inductive learner alone, by having an unsupervised Bayesian classifier as an oracle. Then the robust supervised inductive learner learns the target concept descriptions, providing clear, understandable rules describing these classes. Thus, the system takes advantage of the strengths of its component subsystems.

4. KDLC DATA MINING AND KNOWLEDGE DISCOVERY CASE STUDY

The following sections present the preliminary results achieved by applying KDLC to Idaho precision agriculture data.

4.1. Idaho Precision Agriculture Data Description

As described in Table 1, the data contains a total of 500 examples with 62 attributes. All of the attributes have "continuous" numeric values except the "yield" attribute. Basic interpretations of each attribute are as follows:

- CA*X*: Amount of Calcium in soil. *X* takes values 4, 7, and 10 meaning the soil was sampled in April, July and October.
- CEC, CU, EL, FE, K, MG, MN, N, NA, OM, ZN, ORN, P, PH, S, SA: soil nutrient values as a floating point and were sampled from the small square of soil in the same way. Most of these are chemical names (eg. FE = Iron, CU = Copper, K = Potassium), others are nutrients (eg. EL = excess lime, OM = organic matter, ON = organic nitrogen, SA = salts).
- ELEV: elevation
- Yield: amount of crop produced by that square of ground. Since we did not receive enough guidance from the domain experts on how to scale this class attribute, we scaled it by standard deviations from 0 to 6.

The Range denotes the interval within which attribute values fall, and the Lvl (Levels) column denotes the number of distinct values found for a given attribute.

4.2. Experimental Results

In our experiment, the unsupervised Bayesian classification system, AutoClass discovered 10 classes from the Idaho data (500 data samples for 1996). The supervised inductive rule learning system, AQ15 was then used to learn rule descriptions of each class.

We note that the combination of AutoClass unsupervised learning coupled with supervised learning using AQ15 provided some very interesting results. AutoClass found 10 classes of interest and then AQ was able to find succinct rules to characterize each cluster in terms of important variables associated with farm soil nutrients' measurements.

class_0-outhypo
1…[ca_4=5.67..7.38] [el_10=0..0.13] [mn_4=21.73..31.5]
[mn_10=11.33..41.77] [n_4=6.85..13.28] [n_10=3..9.37]
[p_7=17.58..26.99] [p_10=18.48..25.99] [ph_10=5.4..6.54]
[s_7=1..24.73] (t:119, u:90)
2…[ca_4=6.61..7.6] [n_7=13.43..20.03] (t:32, u:3)
class_1-outhypo
1…[cec_10=8.26..10.09] [k_7=130.54..175.23]
[n_4=6.85..12.39] [n_10=6.78..26.32] (t:77, u:77)
class_2-outhypo
1…[cu_7=0.93..1.07] [k_10=75.11..190.12]
[mn_4=21.12..39.96] [n_4=12.39..21] (t:76, u:76)
class_3-outhypo
1…[ca_10=4.66..5.4] [cec_4=9.07..9.54] [zn_4=1.41..1.6]
(t:45, u:45)
class_4-outhypo
1…[ca_10=4.67..5.3] [cec_10=7.5..8.73] [om_7=2.19..2.32]
[p_4=19.31..21.62] (t:42, u:42)
class_5-outhypo
1…[ca_7=3.5..4.75] [ca_10=5.87..7.49] [cec_4=9.71..10.17]
(t:33, u:33)
class_6-outhypo
1…[ca_7=4.67..5.87] [k_7=104.23..132.17]
[mn_10=15.94..22.38] [n_4=6.85..11.94] (t:33, u:33)
class_7-outhypo
1…[ca_4=4.9..6.13] [fe_7=65.7..76.15] (t:27, u:27)
class_8-outhypo
1…[ca_10=5.28..5.95] [mn_4=8..21.12] (t:24, u:24)
class_9-outhypo
1…[ca_4=5.26..6.59] [cu_7=0.92..0.97] [el_10=0.004..0.6]
[zn_4=0.7..1.42] (t:21, u:21)

Figure 4. Learned Rules and Classes

For example, *Class 0* was characterized by two rules, while each of the other nine classes was characterized by exactly one rule. We note also that the total (t) number of instances and the unique (u) number of instances covered by a rule are identical for classes 1 through 9. In an experiment involving the classes discovered in Figure 4, we notice that each class has varying values for the six values of the YIELD attribute. The rules for YIELD classes 0 – 6, using constructive induction (the class number 0 - 9 is used as a variable) on AutoClass clusters, generated some interesting explanations (the actual rules learned are not shown here but brief explanations are following). For example, the rules for *yield_6-outhypo* captured the Classes from 0 to 5, which cover most of the highest yield regions. One of its rules contained CA_7 and EL_10 attributes from Class 6 and Class 9 whose values overlap with other Classes from 0 to 5. In other words, this rule indicates that the Class 6 and Class 9 also cover the parts of the highest yield regions. However, the yield_6-outhypo rules completely exclude Class 7 and Class 8 without having any common attributes used to characterize the rules. It is also very interesting to note that the attributes CEC_4, CU_4, FE_10, MN_7, S_4, and ELEV are never used to depict the yield_6 region, which is scattered over the original yield map. Based on this analysis for each of rules learned, we have noticed that the rules are described with only those attributes that can discriminate the each of yield classes. It is also noticeable that the complexes in yield rules including class labels consisted of *sets of disjoint attributes* that employed the background theories of AutoClass classification using the most influential set of attributes and AQ's discriminant inductive rule generation.

Table 1. Description of Attribute Values for Idaho Data

#	Name	Range	Lvl	#	Name	Range	Lvl	#	Name	Range	Lvl
1	ca_4	4.9 - 7.6	183	22	mn_4	8 - 39.96	408	43	s_4	3 - 15.99	353
2	ca_7	3.5 - 9.04	196	23	mn_7	4.59 - 35.23	382	44	s_7	1 - 39.99	446
3	ca_10	4.5 - 7.49	175	24	mn_10	5.21 - 41.77	401	45	s_10	4 - 11.99	330
4	cec_4	8.6 - 11.3	187	25	n_4	6.85 - 21	314	46	sa_4	0.5 - 0.8	31
5	cec_7	7.6 - 10.9	158	26	n_7	7.01 - 25.99	392	47	sa_7	0 - 1.1	101
6	cec_10	7.5 - 10.09	189	27	n_10	3 - 34.93	334	48	sa_10	0.5 - 1.1	57
7	cu_4	0.8 - 1.1	31	28	na_4	0.1 - 0.2	11	49	zn_4	0.7 - 2.3	120
8	cu_7	0.9 - 1.1	21	29	na_7	0.2 - 0.3	11	50	zn_7	1.3 - 2.1	73
9	cu_10	0.5 - 1.39	76	30	na_10	0.2	1	51	zn_10	0.7 - 1.69	81
10	el_4	0	1	31	om_4	1.75 - 2.35	51	52	elev	1384 - 1653	48
11	el_7	0 - 0.3	38	32	om_7	1.85 - 2.62	62	53	mnt_4	30.97	1
12	el_10	0 - 0.6	48	33	om_10	1.8 - 2.59	75	54	mxt_4	51.776	1
13	fe_4	30.3 - 74.13	426	34	orn_4	40 - 44.99	216	55	pp_4	0.0441	1
14	fe_7	13.83 - 76.15	457	35	orn_7	40 - 48.45	301	56	mnt_7	47.6283871	1
15	fe_10	17.06 - 74.16	458	36	orn_10	40 - 49.96	300	57	mxt_7	80.69806452	1
16	k_4	125.08 - 374.55	473	37	p_4	14.02 - 23.97	334	58	pp_7	0.025403226	1
17	k_7	95.03 - 294.67	467	38	p_7	15 - 26.99	353	59	mnt_10	26.53323	1
18	k_10	75.11 - 404.74	477	39	p_10	16 - 25.99	325	60	mxt_10	53.53	1
19	mg_4	1.3 - 2.09	60	40	ph_4	5.5 - 6.3	64	61	pp_10	0.0312	1
20	mg_7	1.3 - 2	64	41	ph_7	5 - 7.77	152	62	yield	0 - 6	7
21	mg_10	1.2 - 1.8	55	42	ph_10	5.4 - 7.49	139				

4.3. Graphical Depiction of Learning Experiments

In this section we present the visualizations obtained from the ArcView System showing the geographic regions of the Idaho property from which the data sets were obtained. The images correspond to the classes discovered by the AutoClass program. The complexes (conditions) of the AQ rules characterizing each class were used to query the ArcView database of the entire Idaho data set, and the various classes were plotted on the map. We note that the classes 0, 1 and 2 when plotted over the background constitute contiguous regions on the farm, and each class has been characterized by relatively simple rules as shown in Figure 4. We note that these rules include measurements of variables from the months of April, July and October. Here it would be appropriate to call in the "panel of experts" to provide expert advice on how best to pursue the inductive learning part of the KDLC. Clearly, the visualization of knowledge is crucial to the KDLC and the use of the ArcView GIS has allowed us to visualize the classes. The following Figures 5 - 7 show screen shots for various classes discovered by the AutoClass Program. Only Classes 0, 1, and 2 from Figure 4 have been presented due to the lack of space.

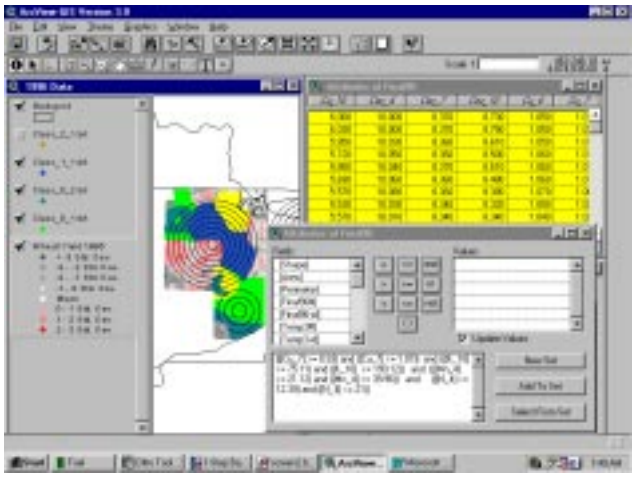


Figure 5. The ArcView GIS Query Form

5. CONCLUSIONS

This paper has presented both a methodology and an associated knowledge discovery life cycle model, called the *KDLC*, for data mining and knowledge discovery in large, multiple and heterogeneous databases. A key contribution of this approach is the integration of these two types of learning into a multistrategy KDD approach, combining *unsupervised learning*, such as Bayesian or statistical classification, with *supervised inductive learning.*

The particular approach begins by analyzing large data sets using an unsupervised Bayesian classification system, *AutoClass*. AutoClass discovers interesting taxonomies from databases, and these taxonomic class representations can be represented as new attributes in an expanded representation space via constructive induction. The inductive learning system, *AQ15c*, can then be used to "learn" useful concepts, relationships, and rules that characterize knowledge in the data space. We have also performed experiments with statistical classification K-means Clustering and Ward Clustering [12], but lack of time did not permit a thorough analysis of the results. A case study dealing with "crop yields" for a farm in the state of Idaho is presented and preliminary results of the KDD process with the KDLC are discussed.

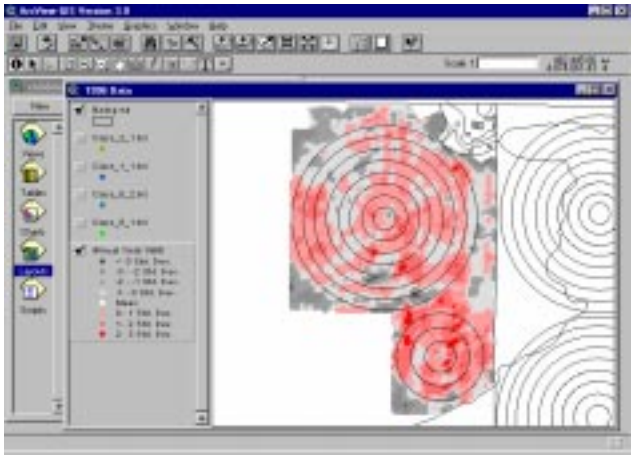


Figure 6. Background and Yield Map

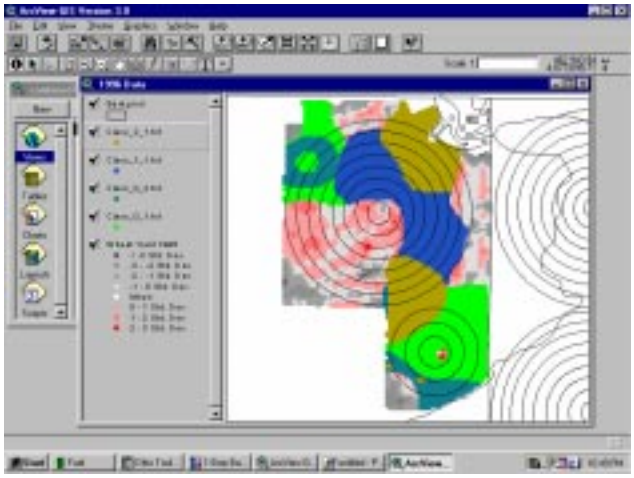


Figure 7. Classes 0, 1, and 2, Yield and Background

Our experiments have shown that a multistrategy KDD approach combining unsupervised and supervised learning could yield interesting results. In particular, we have shown that knowledge visualization is indeed crucial to helping both users and knowledge engineers to understand and analyze the rules and concepts that have been discovered. Additionally, the KDLC envisions a "panel of experts" to comment of discovered knowledge and to provide advice for future learning experiments. We believe these techniques, when combined with the Knowledge Discovery Life Cycle, can form the basis of a robust set of tools for decision support, data mining, knowledge discovery, knowledge visualization, and knowledge/data evolution. There is a need to couple the learning approaches with advanced database technology to handle very large databases [14]. Future research should also focus on the use of intelligent software [6] for the negotiation, retrieval, and data mining from high-quality, reliable information sources.

6. ACKNOWLEDGEMENTS

The authors thank Linda Tischer for visualizing the experimental results using ArcView GIS system. The first author also greatly appreciates the support of a George Mason University Doctoral Fellowship.

7. REFERENCES

[1] Cheeseman, P., and Stutz, J. "Bayesian Classification (AutoClass): Theory and Results" In *Advances in Knowledge Discovery and Data Mining*. Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R, eds.: AAAI Press, Menlo Park. 1996

[2] Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. Eds. *Advances in Knowledge Discovery and Data Mining.*: The AAAI Press, Menlo Park. 1996.

[3] Hanson, R., Stutz, J., and Cheeseman, P. "Bayesian Classification Theory", Technical Report FIA-90-12-7-01, NASA Ames Research Center, Artificial Intelligence Branch. 1991.

[4] Heckerman, D. "Probabilistic interpretations for Mycin's certainty factors". In *Readings in Uncertain Reasoning*. 298-312. Shafer, G., and Pearl, J. eds., Morgan Kaufmann, San Mateo. 1990.

[5] Kerschberg, L. "Knowledge Rovers: Cooperative Intelligent Agent Support for Enterprise Information Architectures" *Cooperative Information Agents*, P. Kandzia and M. Klusch, Springer-Verlag. **1202:** 79-100. 1997.

[6] Kerschberg, L. "The Role of Intelligent Software Agents in Advanced Information Systems" *Advances in Databases*. C. Small, P. Douglas, R. Johnson, P. King and N. Martin. London, Springer-Verlag. **1271:** 1-22. 1997.

[7] Kohonen, T. *Self Organizing Maps*. Springer-Verlag, Heidelberg, 1995.

[8] Lee, S. W., Fischthal, S. and Wnek, J. "A Multistrategy Learning Approach to Flexible Knowledge Organization and Discovery", In *Proceedings of AAAI-97*, Workshop on AI and Knowledge Management, pp. 15-24. Providence, Rhode Island, AAAI Press, Menlo Park, CA. 1997.

[9] Michalski, R. S. "Pattern Recognition as Knowledge-Guided Computer Induction" *Technical Report No. 927*, Department of Computer Science, University of Illinois, Urbana-Champaign. 1978.

[10] Michalski, R. S. "A Theory and Methodology of Inductive Learning" *Artificial Intelligence*. **20:**111-116. 1983.

[11] Michalski, R. S., Kerschberg, L. et al. "Mining for Knowledge in Databases: The INLEN Architecture, Initial Implementation and First Results." *Journal of Intelligent Information Systems* **1**(1): 85-113. 1992.

[12] Sharma, S. *Applied Multivariate Techniques*.: John Wiley, New York. 1996.

[13] Wnek, J., and Michalski, R. S. "Hypothesis-driven Constructive Induction in AQ17-HCI: A method and experiments" *Machine Learning*, **14**(2):139-168. 1994.

[14] Yoon, J. P., and L. Kerschberg "A Framework for Knowledge Discovery and Evolution in Databases." *IEEE Transactions on Knowledge and Data Engineering*. 1993.