AI Magazine Volume 21 Number 3 (2000) (© AAAI)



# Building Intelligent Learning Database Systems

*Xindong Wu*

■ Induction and deduction are two opposite operations in data-mining applications. Induction extracts knowledge in the form of, say, rules or decision trees from existing data, and deduction applies induction results to interpret new data. An intelligent learning database (ILDB) system integrates machine-learning techniques with database and knowledge base technology. It starts with existing database technology and performs both induction and deduction. The integration of database technology, induction (from machine learning), and deduction (from knowledge-based systems) plays a key role in the construction of ILDB systems, as does the design of efficient induction and deduction algorithms. This article presents a system structure for ILDB systems and discusses practical issues for ILDB applications, such as instance selection and structured induction.

Over the past 30 years, database research has developed technologies that are now widely used in almost every computing and scientific field. However, many new advanced applications, including computer-aided design (CAD) and computer-aided manufacturing (CAM), have revealed that traditional database management systems (DBMSs) are inadequate, especially in the following cases:

Conventional database technology has laid particular stress on dealing with large amounts of persistent and highly structured data efficiently and using transactions for concurrency control and recovery. For some applications, such as CAD and CAM where the data schemata need to vary frequently, new data models are needed.

In some applications, such as geographic data and image data, the semantic relationships among data (such as the variations and developments of real-world entities in function, performance, structure, and status, with time and external variables' variations) need to be represented as well as the data itself. Conventional data models in database technology cannot support any representation facility for complex semantic information.

Traditional database technology can only support facilities for processing data. Along with the developments of other subjects, such as decision science and AI, more and more applications need facilities for supporting both data management and knowledge management (such as rules for automatic data inferring and management of integrity constraints between data).

To widen the applicability of database technology to these new kinds of application, object-oriented approaches are currently popular in processing structurally complex objects, and deductive databases or logic databases have been expected to support a solution to those applications where both knowledge and data models are needed. However, the knowledge bases (which contain deductive rules and/or semantic information such as the conceptual hierarchy among data) in existing deductive database systems can only be built up manually with known technology. Automatic knowledge acquisition or learning from databases directly in deductive database systems has become a central and difficult problem in deductive database systems research.

Existing work relevant to knowledge acquisition falls into the following four categories: (1) adding an induction engine to an existing database system in an ad hoc way to implement rule induction from (or data mining in) databases, (2) designing a specific engine to learn from a domain-specific data set, (3) building ontologies and knowledge bases for expert systems, and (4) designing various learning algorithms that have no direct connections with existing database technology. However, when we integrate machine-learning techniques into database systems to implement data mining or knowledge acquisition from databases, we face many problems, such as (1) efficient learning algorithms because

*Because database technology has found wide applications in various fields, it will surely generate significant effect on machine-learning research if we can couple them well. Therefore, research on knowledge acquisition from databases can be viewed as an important frontier for both database and machine-learning technology*

realistic databases are typically large and noisy and (2) expressive representations for both data (for example, tuples in relational databases, which represent instances of a problem domain) and knowledge (for example, rules in a rule-based system, which can be used to solve users' problems in the domain, and the semantic information contained in the relational schemata).

Meanwhile, although some commercial successes have been found in existing learning systems, there are limitations on current machine-learning techniques for both research and applications. The limited industrial support to the machine-learning community is an example of such limitations. Because database technology has found wide applications in various fields, it will surely generate significant effect on machine-learning research if we can couple them well. Therefore, research on knowledge acquisition from databases can be viewed as an important frontier for both database and machine-learning technology (Wu 1993c).

This article discusses issues in integrating machine learning with database and knowledge base technology to construct intelligent learning database (ILDB) systems. In the following section, I outline the system structure of an ILDB system that I have developed. In Induction from Databases and Deduction of Induction Results, I survey existing induction and deduction techniques, respectively, for ILDB systems. I follow these sections with a discussion of some essential problems in practical systems construction.

## System Structure

An ILDB system (Wu 1995) supports database and knowledge base management functions as well as learning facilities. It provides mechanisms for (1) preparing and translating standard (for example, relational) database information into a form suitable for use by its induction engines, (2) using induction techniques to extract knowledge from databases, and (3) interpreting the knowledge produced to solve users' problems. With an ILDB system, one can, for example, produce 100 to 200 conjunctive rules for 50 diseases from 2 million medical cases of the 50 diseases. Then, the ILDB system can use the rules in two different ways: (1) keep these rules instead of the original cases because the original cases might take a large space and (2) use these rules to diagnose new cases.

Figure 1 shows the system structure of my KESHELL2 system (Wu 1995). In the diagram, *Monitor* is a man-machine interface that exchanges information with users in the form of pull-down menus. *KBMS* and *DBMS* are facilities to support knowledge base and database management functions. *DB* and *KB* denote databases and knowledge bases, respectively, and *OS* indicates operating system facilities. The knowledge-acquisition engine, *K.A. Engine,* implements induction from databases. *I/D Engine* is an inference and deduction engine. *Utility* contains a set of common procedures that are shared by K.A. Engine, KBMS, and DBMS. *Access Storage Interface* is composed of the basic knowledge and data operators.

### KBMS

The KBMS module supports facilities for interactively building, adapting, and displaying knowledge bases; checking for semantic inconsistencies; sorting knowledge bases to implement efficient chaining (Wu 1993a); and editing knowledge base files.

### DBMS

The DBMS module is based on a commercial relational DBMS. Users can do conventional database operations by simply calling the commercial system. However, a new function, List a Relation, is developed here to translate relational files into the Prolog-based representation (Wu 1993b) suitable for use by the K.A. Engine. This representation relates to the problem of expressive representations for both data and knowledge. It binds actual relational data and the data schema together in an explicit way and can represent semantic information (such as logic implication and constraints between attributes or fields in entities) as well as all the information that can be represented in the widely adopted entity-relationship (E-R) model.

### K.A. Engine

There are two submodules in the K.A. Engine: (1) *semantic information,* which generates semantic networks from relational database schemata in an interactive manner, and (2) *rule induction,* which constructs decision trees and generates rules.

### I/D Engine

The I/D Engine tests the knowledge produced by the K.A. Engine on new data sets and interprets it to solve users' problems.

## Induction from Databases

Among the functions shown in figure 1, the K.A. Engine is the central module in an ILDB system. The K.A. Engine relates to the design of

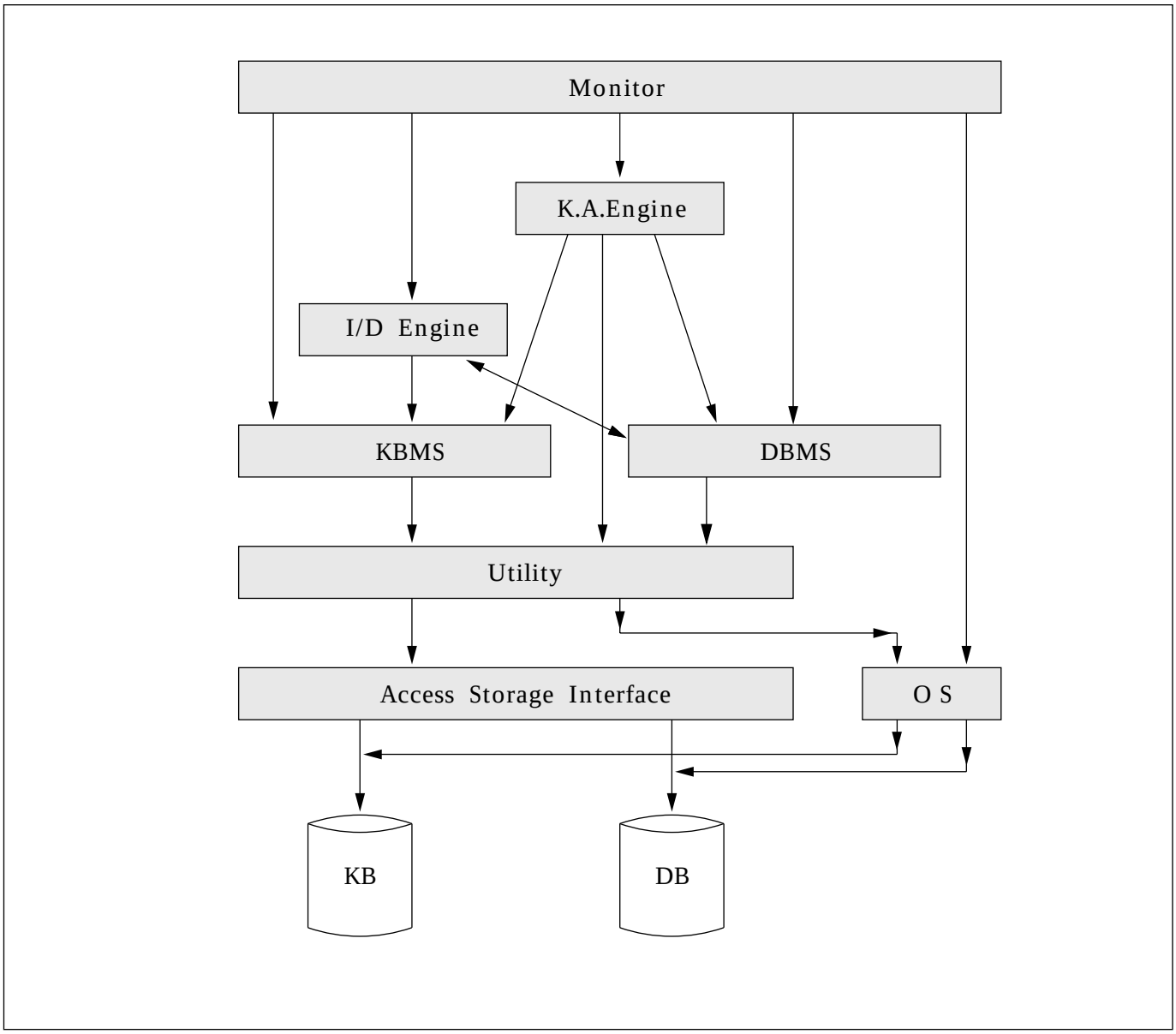



*Figure 1. The System Structure of an ILDB System.*

efficient learning algorithms, which can generally be divided into three categories:[1] (1) supervised classification, (2) unsupervised clustering, and (3) association analysis. This section provides a review of existing techniques in these categories, all of which can be integrated into the system structure in figure 1.

Let $E = D_1 \times \ldots \times D_a$ be a finite attribute space of $a$ dimensions, where each $D_j$ ($j = 1, \ldots, a$) is a finite set of symbolic values or a numeric interval. An instance $e = (v_1, \ldots, v_a)$ is an element of $E$ means $v_j \in D_j$. Each instance in a classification belongs to a known class that, say, has a specific name in $E$. The classification task is to generate a description, say, production rules or a decision tree, that distinguishes instances of each class from other classes.

Attribute-based induction algorithms (such as ID3 [Quinlan 1986], c4.5 [Quinlan 1993]; AQ, CN2 [Clark and Niblett 1989]; AE1 [Hong 1985], and HCV [Wu 1993d]), incremental induction algorithms (such as ID5R [Utgoff 1989], and the version-space method [Mitchell 1977]) fall into the supervised classification category.

Unsupervised clustering (or concept formation [Langley 1987]) deals with the discovery of new concepts from unclassified data. The data input for clustering is similar to that for classi-

fication, but the significant difference is that no class information is available for each instance. Well-known algorithms in unsupervised clustering are CLUSTER/2, UNIMEM, COBWEB (Fisher 1996), CLASSIT, AUTOCLASS, BIRCH (Zhang, Ramakrishnan, and Livny 1997), and the connectionist Kohonen self-organizing map and backpropagation (Dayhoff 1990).

Association analysis (Han, Pei, and Yin 2000; Agrawal and Srikant 1994; Agrawal, Imielinksi, and Swami 1993) starts with a different data environment. Let $I = \{i_1, i_2, \ldots, i_N\}$ be a set of $N$ distinct literals called items and $D$ a set of transactions over $I$. Each transaction is a set of items $i_1, i_2, \ldots, i_k \in I$. An association rule is an implication of the form $A \rightarrow B$, where $A, B \subset I$, and $A \cap B = \varnothing$. $A$ is called the *antecedent* of the rule, and $B$ is called the *consequent*.

*In practical ILDB systems, we would need to construct an algorithm library containing learning algorithms of different paradigms and corresponding documentation for the user to refer to, and choose among, the algorithms.*

A set of items (such as the antecedent or the consequent of a rule) is called an *item set*. The number of items in an item set is the *length* (or size) of the item set. An item set of some length $k$ is referred to as a *k–item set*. Each item set has an associated statistical measure called *support*, denoted as supp. For an item set $A \subset$ I, supp($A$) = $s$, if the fraction of transactions in $D$ containing $A$ is equal to $s$. A rule $A \rightarrow B$ has a measure of strength called *confidence* (denoted as conf), which is defined as the ratio supp($A \cup B$)/supp($A$). The problem of association analysis is to generate all rules $A \rightarrow B$ that have both support and confidence greater than or equal to some user-specified thresholds, called *minimum support* (minsupp) and *minimum confidence* (minconf), respectively.

Association analysis from large databases has received much attention recently. To discover a useful and interesting association analysis, a wide range of problems have been investigated over such diverse topics as generalized association rules (Aggarwal and Yu 1998; Tsur et al. 1998; Agrawal and Srikant 1994; Agrawal, Imielinksi, and Swami 1993), measurements of interestingness (Aggarwal and Yu 1998), quantitative association rules (Srikant and Agrawal 1996), multiple-level association rules (Han and Fu 1999), and association with sequential patterns (Agrawal and Srikant 1995).

Every learning paradigm has its own advantages and disadvantages. For example, attribute-based induction algorithms cannot produce first-order rules on their own. Inductive logic programming research can contribute in this regard, although it is in general less efficient for problems where attribute-based induction works. However, because it requires background knowledge before learning can be carried out, inductive logic programming has not found much realistic application with existing database technology. When the available data sets are not well organized or contain too much noise, both attribute-based induction and inductive logic programming perform poorly, although they can sometimes deal with a small amount of noise. Connectionist and statistical methods are able to show their significance in these environments. When examples in a data set are not classified with distinctive concepts, only connectionist and statistical clustering methods can be adopted to carry out unsupervised learning. In practical ILDB systems, we would need to construct an algorithm library containing learning algorithms of different paradigms and corresponding documentation for the user to refer to, and choose among, the algorithms.

## Deduction of Induction Results

Because real-world databases are typically incomplete and noisy, induction results cannot be assumed to be perfect. When induction results take the form of rules, interpreting them to classify a new instance needs to face three possible cases that demand different actions. First is *no match:* No rules match the instance. Second is *single match:* One or more rules indicate the same class match. Third is *multiple match:* More than one rule matches the instance and indicates different classes.

The third case does not apply to decision trees produced by ID3-like algorithms, but when the trees are decompiled into production rules (Quinlan 1993), the production rules will face the same problems.

In the single-match case, the choice of class to the instance is naturally the class indicated by the rules. Deduction-time processing deals mainly with the conflict resolution in the third case and class estimation for the first case. Existing techniques for dealing with the first and third cases are both exclusively based on probability estimation. Among them, the *measure of fit* for dealing with the no match case and the *estimate of probability* for handling the multiple-match case developed in AQ15 (Michalski et al. 1986) have been adopted widely in knowledge discovery and data mining.

The measure-of-fit and estimate-of-probability methods perform well with problem domains where no real-valued attributes are involved. However, when a problem contains attributes that take values from continuous domains (that is, real numbers or integers), their performance, especially in terms of accuracy, decreases. In existing induction algorithms, dealing with continuous domains is based on the discretization of them into a cer-

tain number of intervals. There are quite a few strategies available for discretization, such as the information gain–based methods (Dougherty, Kohavi, and Sahami 1995; Quinlan 1993; Fayyad and Irani 1992). Once each continuous domain has been discretized into intervals, the intervals are treated as discrete values in induction and deduction. Discretization is the standard way existing induction systems have taken. However, discretization of continuous domains does not always fit accurate interpretation. To say an age greater than 50 is old or a temperature above 32 degrees Centigrade is high is fuzzy. In such cases, fuzzy interpretation of the discretized intervals at deduction time could be valuable. Rather than taking the cut points decided by a discretization method as sharp borders for intervals, we can instead place some kind of curve at each cut point as a fuzzy border. With these fuzzy borders, a value can be classified into a few different intervals at the same time, with varying degrees. Thus, a single-match case could change to a multiple match, and a no-match case could change to a single or even multiple match. Deduction with fuzzy borders of discretized intervals is called *fuzzy matching*. In the multiple-match case, we can take the interval with the greatest degree as the value's discrete value. Wu (1999) describes an implementation of the fuzzy matching techniques.

## Practical Issues

When building practical ILDB systems, we need to face the following important problems, in addition to noise handling and dealing with both numeric and nominal data, which have received wide attention in the design of various data-mining systems.

**Instance selection:** Dealing with very large databases is one of the defining challenges in data-mining research and development. No matter how powerful computers are or will be in the future, data-mining researchers and practitioners must consider how to manage ever-growing data that can be too large (for example, with terabytes of data) to be processed at one time. *Instance selection* is about approaches that select or search for a portion of a large database that can be used in data mining instead of the whole database. One of the major approaches for instance selection is sampling, in which a sample is selected for testing and analysis. Other major approaches include windowing, data reduction, and selection of representative instances.

**Structured induction:** The basic idea of *structured induction* (Shapiro 1987) is to decompose a complex problem that might be very large in size into a number of subproblems by using domain knowledge and apply an induction algorithm to each of the subproblems. When induction results from all these subproblems are put in a single knowledge base, we need chaining mechanisms to perform deduction on examples from the complex problem. There has been some significant work in the design of efficient chaining algorithms for expert systems (Wu 1993a), but dealing with no match and multiple match cases needs more attention in the data-mining context.

**Constructive induction:** None of the AQ-like, ID3-like and HCV-like algorithms (Wu 1993c) need explicit, built-in background knowledge, which is why they are sometimes called *empirical learning methods*. Such learning methods are different in nature from the knowledge-rich learning methods, such as AM and EURISKO developed by Lenat (1983, 1979), explanation-based learning, and inductive logic programming.

However, there is always implicit background knowledge embedded in the formulation of solution spaces and in the representation of examples. When a solution space turns out to be inadequate, representation modification is needed, and the modification process typically involves searching for useful new descriptive features (constructive induction) in terms of existing features or attributes. AQ17 (Wnek and Michalski 1994) of the AQ-like family has been developed to implement iterative construction of new attributes based on existing ones. Zheng (1995) has also tried a method called X-of-N attributes in constructive decision-tree construction.

Constructive learning has become a strong theme in inductive learning research. One of the difficulties in constructive learning is that the complexity in some cases (such as iterative feature construction) is extreme, but there are situations in which it is a necessary part of learning.

**Integration of object-oriented design to learn semantic information from relational schemata involving complex objects:** The entity-relationship (E-R) model is one of the most successful methods of formulating useful abstract models in the conceptual structure design of databases and the key design aid for conventional databases implemented under a wide variety of commercially available systems. By focusing on the entities and their relationships, the E-R model structures the way designers approach the problem of creating extensible databases. However, there are two substantial problems here. One is that trans-

forming an E-R model into a relational model during the logical design of databases results in the loss of some semantic information that exists in the E-R model. In other words, the entities and relationships are not distinguished in the relational data model. The other problem is that the relationship types of the E-R model are too simple to express complex semantic features of relationships among and within entities. It is impossible for the relational data model to describe the changes in relationship(s) and other entities caused by an entity in an E-R model. The integration of object-oriented design is expected to provide facilities for expressing and acquiring these kinds of semantic information. The semantic information can be used to guide and facilitate knowledge discovery in ILDB systems.

**Incremental induction in the case of large, dynamic real-world databases:** There are several common problems in all kinds of inductive learning algorithm: (1) When a database is very large, how can these algorithms speed up their learning processes? (2) When a database is not a static repository of data—for example, examples can be added, deleted, or changed—the induction on the example set cannot be a one-time process, so how can induction algorithms deal with the changing examples? (3) When some inconsistency (for example, noise) is found in a database or a knowledge base just produced, how can they remove it?

One possible way to solve those problems is *incremental learning,* which means dividing a large example set into a number of subsets and treating each subset each time. Although no existing algorithms have found a complete solution to these problems, many research efforts have been made along this direction. For example, AQ15, AE5, ID5R (Utgoff 1989), and the windowing technique in ID3 can be viewed as good examples of research in incremental learning. However, how to deal with the inconsistency between new data and the data previously used is still an open question. The inconsistency might be caused by dynamic changes of the data attributes.

Generally speaking, incremental induction can take more time (but less run-time space) because it needs to restructure decision trees or rules when some new examples do not fit the decision trees or rules developed so far.

**Stronger integration of database technology and data mining.** One of the criticisms from the database community about current data-mining research is that database technology can contribute more than just data preparation for data mining. How to integrate data-mining techniques with existing database technology has been a popular topic for both communities. Knowledge discovery and data mining can augment the ability of existing database systems to represent, acquire, and process a collection of expertise such as those that form part of the semantics of many advanced applications. In the meanwhile, there should be ways to further explore new information from data processing. For example (Wu and Craske 1997), query results from databases with existing DBMSs could be a good source of information for providing hypotheses for knowledge discovery and data mining.

## Conclusions

Knowledge acquisition from databases has been worked over by researchers in several disciplines, including AI and databases, for a decade and is still an important research frontier for both machine learning and database technology.[2] Although a lot of work has been done and some commercial data-mining packages are available already, existing work has not paid enough attention to the integration of database and knowledge base technology with machine-learning techniques.

With the World Wide Web's emergence as a large, distributed data repository and the realization that online transaction databases can be analyzed for commercial gains, data mining in large databases has attracted wide interest from both academia and the industry and, in the meanwhile, has also uncovered new challenges (Ramakrishnana and Gramam 1999). Data mining has its distinctive goal from related fields such as machine learning, databases, and statistics and accordingly requires distinctive tools. An ILDB system is one such tool to implement automatic knowledge acquisition from databases.

### Acknowledgments

I am indebted to the anonymous reviewers for their constructive and supportive comments on earlier versions of this article. This article is a modified and extended version of the author's invited keynote address at the 1997 International Conference on the Practical Application of Knowledge Discovery and Data Mining, London, United Kingdom, 23 to 25 April 1997. The original keynote address has been updated to cover the current state of the art.

### Notes

1. The discovery of quantitative laws among features-parameters can possibly be listed as a fourth category, but this discovery is more on the statistical side than AI-related data mining.

2. The First Knowledge Discovery and Data-Mining

Workshop was held in Detroit, Michigan, in August 1989 in conjunction with the 1989 International Joint Conference on Artificial Intelligence.

## References


Aggarwal, C., and Yu, P. 1998. A New Framework for Itemset Generation. In Proceedings of the ACM Symposium on Principles of Database Systems. New York: Association of Computing Machinery.

Agrawal, R., and Srikant, R. 1995. Mining Sequential Patterns. Paper presented at the Eleventh International Conference on Data Engineering (ICDE), 3–14 March, Taipei, Taiwan.

Agrawal R., and Srikant, R. 1994. Fast Algorithms for Mining Association Rules. Paper presented at the Twentieth International Conference on Very Large Databases, 12–15 September, Santiago, Chile.

Agrawal, R.; Imielinski, T.; and Swami, A. 1993. Mining Associations between Sets of Items in Massive Databases. In Proceedings of the ACM SIGMOD International Conference on Management of Data, 207–216. New York: Association of Computing Machinery.

Clark P., and Niblett, T. 1989. The cn2 Induction Algorithm. *Machine Learning* 3:261–283.

Dayhoff, J. E. 1990. *Neural Network Architectures: An Introduction*. New York: Van Nostrand Reinhold.

Dougherty, J.; Kohavi, R.; and Sahami, M. 1995. Supervised and Unsupervised Discretization of Continuous Features. In *Proceedings of the Twelfth International Conference on Machine Learning,* 194–202. San Francisco, Calif.: Morgan Kaufmann.

Fayyad, U. M., and Irani, K. B. 1992. On the Handling of Continuous-Valued Attributes in Decision Tree Generation. *Machine Learning* 8:87–102.

Fisher, D. 1996. Iterative Optimization and Simplification of Hierarchical Clusterings. *Journal of Artificial Intelligence Research* 4:147–180.

Han, J., and Fu, Y. 1999. Multiple-Level Association Rules. *IEEE Transactions on Knowledge and Data Engineering* 11(5): 798–805.

Han, J.; Pei, J.; and Yin, Y. 2000. Mining Frequent Patterns without Candidate Generation. In Proceedings of the 2000 ACM-SIGMOD International Conference on Management of Data (SIGMOD '00). New York: Association of Computing Machinery. Forthcoming.

Hong, J. 1985. AE1: An Extension Matrix Approximate Method for the General Covering Problem. *International Journal of Computer and Information Sciences* 14(6): 421–437.

Langley, P. 1987. Machine Learning and Concept Formation. *Machine Learning* 2:99–102.

Lenat, D. B. 1983. EURISKO: A Program That Learns New Heuristics and Domain Concepts—The Nature of Heuristics III: Program Design and Results. *Artificial Intelligence* 21:61–98.

Lenat, D. B. 1979. On Automated Scientific Theory Formation: A Case Study Using the AM Program. In *Machine Intelligence 9,* eds. J. Hayes et al. New York: Halstead.

Michalski, R.; Mozetic, I.; Hong J.; and Lavrac, N. 1986. The Multipurpose Incremental Learning System AQ15 and Its Testing Application to Three Medical Domains. In Proceedings of the Fifth National Conference on Artificial Intelligence, 1041–1045. Menlo Park, Calif.: American Association for Artificial Intelligence.

Mitchell, T. 1977. Version Spaces: A Candidate Elimination Approach to Rule Learning. In Proceedings of the Fifth International Joint Conference on Artificial Intelligence, 305–310. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Quinlan, J. R. 1993. C4.5: *Programs for Machine Learning.* San Francisco, Calif.: Morgan Kaufmann.

Quinlan, J. R. 1986. Induction of Decision Trees. *Machine Learning* 1:81–106.

Ramakrishnan, N., and Grama, A. Y. 1999. Data Mining: From Serendipity to Science. *IEEE Computer* 32(8): 34–37.

Shapiro, A. D. 1987. *Structured Induction in Expert Systems.* Wokingham, U.K.: Turing Institute Press.

Srikant, R., and Agrawal, R. 1996. Mining Quantitative Association Rules in Large Relational Tables. In Proceedings of the ACM-SIGMOD 1996 Conference on Management of Data, 1–12. New York: Association of Computing Machinery.

Tsur, D.; Ullman, J.; Abiteboul, S.; Clifton, C.; Motwani, R.; Nestorov, S.; and Rosenthal, A. 1998. Query Flocks: A Generalization of Association-Rule Mining. In Proceedings of the ACM SIGMOD International Conference on Management of Data, 1–12. New York: Association of Computing Machinery.

Utgoff, P. E. 1989. Incremental Induction of Decision Trees. *Machine Learning* 4:161–186.

Wnek, J., and Michalski, R. S. 1994. Hypothesis-Driven Constructive Induction in AQ17-HCI: A Method and Experiments. *Machine Learning* 14:139–168.

Wu, X. 1999. Fuzzy Interpretation of Discretized Intervals. *IEEE Transactions on Fuzzy Systems* 7(6): 753–759.

Wu, X. 1995. Knowledge *Acquisition from Databases.* Norwood, N.J.: Ablex.

Wu, X. 1993a. LFA: A Linear Forward-Chaining Algorithm for AI Production Systems. Expert Systems: The International *Journal of Knowledge Engineering* 10(4): 237–242.

Wu, X. 1993b. A Prolog-Based Representation for Integrating Knowledge and Data. *Informatica: An International Journal of Computing and Informatics* 17(2): 137–144.

Wu, X. 1993c. Inductive Learning: Algorithms and Frontiers. *Artificial Intelligence Review* 7(2): 93–108.

Wu, X. 1993d. The HCV Induction Algorithm. In Proceedings of the Twenty-First ACM Computer Science Conference, 168–175. New York: Association of Computing Machinery.

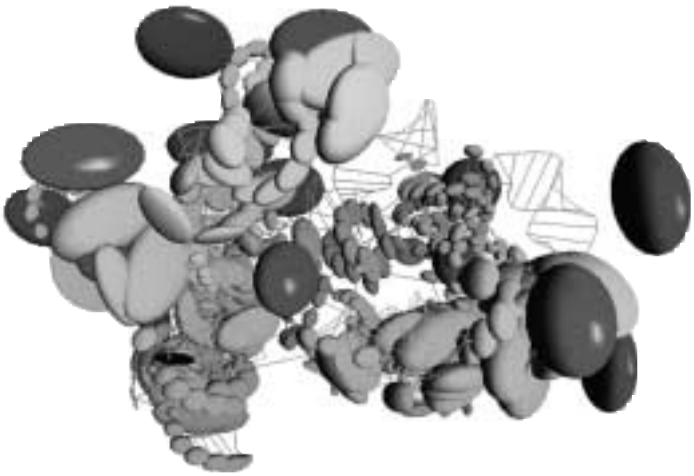
Wu, Y., and Craske, N. 1997. Discovery from Queries. In Proceedings of the 1997 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX-97), 120–122. Washington, D.C.: IEEE Computer Society.

Zhang, T.; Ramakrishnan, R.; and Livny, M. 1997. BIRCH: A New Data-Clustering Algorithm and Its Applications. *Data Mining and Knowledge Discovery* 1(2): 141–182.

Zheng, Z. 1995. Constructing Nominal X-of-N Attributes. In Proceedings of the 1995 International Joint Conference on Artificial Intelligence, 1064–1070. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.



**Xindong Wu** is currently an associate professor in computer science at the Colorado School of Mines. He holds a Ph.D. in AI from the University of Edinburgh. Wu is the executive editor of *Knowledge and Information Systems* and also serves on the editorial boards of the *IEEE Transactions on Knowledge and Data Engineering* and *Data Mining and Knowledge Discovery.* He is currently chair of the Steering Committee of the annual Pacific-Asia Conference on Knowledge Discovery and Data Mining. His e-mail address is xindong@computer.org.



# *Intelligent Systems for Molecular Biology*

The ISMB conference series provides a general forum for disseminating the latest developments in bioinformatics. ISMB is a multidisciplinary conference that brings together scientists from computer science, mathematics, molecular biology, and statistics. Its scope includes the development and application of advanced computational methods for biological problems.

Published by The AAAI Press
445 Burgess, Drive, Menlo Park, California 94025
http://www.aaaipress.org/
650-328-3123 • 650-321-4457 (fax)

*(AAAI members may deduct 20% from list price)*

*ISMB-2000—San Diego, California*

**Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology**

*Edited by Russ Altman, Timothy Bailey, Philip Bourne, Michael Gribskov, Thomas Lengauer, Ilya Shindyalov, Lynn Ten Eyck, and Helge Weissig*

ISBN 1-57735-115-0 436 pp., index, $45.00 softcover

*ISMB-99—Heidelberg, Germany*

**Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology**

*Edited by Thomas Lengauer, Reinhard Schneider, Peer Bork, Douglas Brutlag, Janice Glasgow, Hans-Werner Mewes, and Ralf Zimmer*

ISBN 1-57735-083-9 324 pp., index, $45.00 softcover

*ISMB-98—Montréal, Quebec, Canada*

**Proceedings of the Sixth International Conference on Intelligent Systems for Molecular Biology**

*Edited by Janice Glasgow, Tim Littlejohn, François Major, Richard Lathrop, David Sankoff, and Christoph Sensen*

ISBN 1-57735-053-7 234 pp., index, $45.00 softcover

*ISMB-97—Halkidiki, Greece*

**Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology**

*Edited by Terry Gaasterland, Peter Karp, Kevin Karplus, Christos Ouzounis, Chris Sander, and Alfonso Valencia*

ISBN 1-57735-022-7 382 pp., index, $50.00 softcover

*ISMB-96—St. Louis, Missouri*

**Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology**

*Edited by David J. States, Pankaj Agarwal, Terry Gaasterland, Lawrence Hunter, and Randall F. Smith*

ISBN 1-57735-002-2 274 pp., index, $50.00 softcover

*ISMB-95—Cambridge, England*

**Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology**

*Edited by Christopher Rawlings, Dominic Clark, Russ Altman, Lawrence Hunter, Thomas Lengauer, and Shoshana Wodak*

ISBN 0-929280-83-0 427 pp., index, $50.00 softcover

*ISMB-94—Stanford, California*

**Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology**

*Edited by Russ Altman, Douglas Brutlag, Peter Karp, Richard Lathrop, and David Searls*

ISBN 0-929280-68-7 401 pp., index, $45.00 softcover

*ISMB-93—Bethesda, Maryland*

**Proceedings of the First International Conference on Intelligent Systems for Molecular Biology**

*Edited by Lawrence Hunter, David Searls, and Jude Shavlik*

ISBN 0-929280-47-4 468 pp., index, $45.00 softcover