Data Mining with Distributed Agents in E-Commerce Applications

Y. Lee[1], J. Geller[2], E. K. Park[1], C. Oh[1]
[1]Computer Science Telecommunications
University of Missouri-Kansas City
5100 Rockhill Road, Kansas City MO, 64110
[2]Department of Computer and Information Sciences
New Jersey Institute of Technology
Newark, NJ 07102

Abstract

In this paper we describe the prototype of a yellow page service for customers in a distributed cyber-shopping mall. This application combines distributed data mining with agent technologies. The paper focuses on a framework to support distributed data mining. Data mining approaches have dealt with finding interesting patterns, however, there is little research on developing a framework for effective and efficient *distributed data mining*. Our approach to providing such a framework combines a concept hierarchy *and* an efficient, distributed encoding of that concept hierarchy with existing data mining methods. This marriage results in a new distributed data representation for data mining, called Combined Hierarchical Set (CHS). CHS provides a framework for knowledge discovery including discovery of generalized associations, aggregated associations, and combined associations.

Introduction

One important aspect of e-commerce is the buying and selling of products over the Internet. Large amounts of product information and thousands of web sites, including cyber-shopping malls, result in an information overload that makes it hard for online shoppers to select appropriate items. For instance, a customer might want to "find the most popular adventure game similar to Star Trek Voyager working on a Windows 98 Pentium computer."

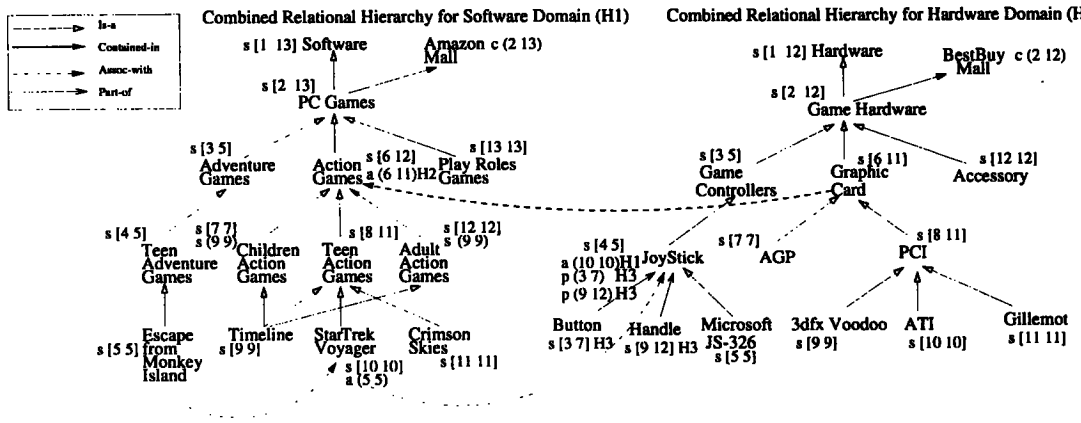
To answer the above question as a data mining query, we will measure popularity by sales figures. Secondly, we need to represent and use the fact that there are several classes of PC games, such as "Strategy," "Action," "Adventure," and "Sports." Among these, Star Trek Voyager is classified as Action game. Thirdly, the game should work on Windows 98 Pentium computers. The ideal answer for the query would be the action game with the maximum sales figure that is compatible with Windows 98 on a Pentium computer. For this inference, we need to perform reasoning using generalization, aggregation and constraint-based association.

We have been working on providing an e-commerce yellow page service for customers of distributed cyber-shopping malls, which have supported such interesting query types. For this service we are using data mining, distributed concept hierarchies and agent technologies.

Data mining has developed advanced techniques for extracting significant patterns or interesting rules in large databases. However, data mining has lots of challenges to surmount. One of the challenging tasks is to discover association rules from massive transaction databases. As the sizes of databases for real world problems grow rapidly, data mining becomes computationally expensive and data mining results are unpredictable (Li and Shasha 1998). The frequent item computation of data mining is one of the time consuming operations. Existing algorithms for finding association rules consider only subsets of interesting items, which are greater than a minimum threshold. However, to find meaningful association rules still requires a large amount of computational power and time (Han et al. 1997). Moreover, the discovered association rules may not reflect the real world situation (Silverstein et al. 1998). Many researchers have realized the requirement of domain knowledge for effective data mining.

One popular representation of domain knowledge is the concept hierarchy. A concept hierarchy represents concepts at different abstraction levels. It provides a framework for natural transition from lower levels to higher levels of abstraction (Han and Fu 1995). Various efforts have been reported to connect knowledge bases and data mining, using concept hierarchies (Han 1995; Han and Fu 1995). Specifically, attribute-oriented induction is an example of using a concept hierarchy to generalize attributes in data mining. Han *et al.* state that the power of generalization leads to a substantial improvement and greater flexibility in querying a database (Han et al. 1996).

Recently, distributed data mining has been studied for large, distributed databases (Cheung et al. 1996; Prodromidis and Stolfo 1998) and it appears to be a natural match for agent technologies (Nodine 1998; Stolfo et al. 1997; Skoron and Stepaniuk 1999). Mul-



Concept	NP	RT	HR	LV	IV
Software	[1 13]	s	H1	1	1/1
PC Games	[2 13]	s	H1	2	1/1
Adventure Games	[3 5]	s	H1	3	1/1
Action Games	[6 12]	s	H1	3	1/1
	(6 11)	a	H2	3	0.8/0.7
Teen Action Games	[8 11]	s	H1	4	1/1
Star Trek Voyager	[10 10]	s	H1	5	1/1
	(5 5)	a	H1	5	0.7/0.6
...	...				

Combined Hierarchical Set (CHR) for H1

Concept	NP	RT	HR	LV	IV
Hardware	[1 12]	s	H2	1	1/1
Game Hardware	[2 12]	s	H2	2	1/1
Game Controllers	[3 5]	s	H2	3	1/1
JoyStick	[4 5]	s	H2	4	1/1
	(10 10)	a	H1	5	0.9/0.8
	(3 7)	p	H3	3	1/1
	(9 12)	p	H3	3	1/1
...	...				

Combined Hierarchical Set (CHR) for H2

Figure 1: Distributed Combined Relational Hierarchies (CRH) and Combined Hierarchical Sets (CHS)

tiple agents collaborate to achieve a common goal such as knowledge discovery. Some major problems with distributed data mining are related to the fact that the domain knowledge is typically represented as trees or graphs (concept hierarchies). It is difficult to distribute such pointer-based knowledge over multiple agents or machines. The Combined Hierarchical Set (CHS) is an efficient representation which supports distributed generalization and effective query processing.

Our approach to the above problems is to combine the association rules (*Assoc-with*) discovered by data mining with a conceptual hierarchy and to model the combined knowledge by a distributed *pointerless* representation. Our conceptual hierarchy (Figure 1) goes beyond the typical *IS-A* hierarchies in that it integrates *IS-A*, *Part-of*, *Contained-in*, and *Assoc-with* in one intuitive hierarchy. In this paper, we use *Part-of* as a synonym of Aggregation (although the inference direction would be opposite). The hierarchy is transformed into a pointerless encoding, CHS.

Since knowledge base lookup during data mining takes a considerable amount of time, an efficient encoding of the concept hierarchy is of major importance. Even if a representation provides a fast lookup, if the space requirements are too high, the encoding is not acceptable. If the encoding is localized, but the data is distributed, additional problems may occur. Ideally, we would like to provide a distributed hierarchy encoding with small space and time requirements.

The CHS is based on a hierarchy representation called *materialized transitive closure* (Agrawal et al. 1989) which approaches the above requirements. In extensive previous research we have adapted materialized transitive closures to high performance computing (Lee and Geller 1996a, 1996b; Lee 1997). Our paper in FLAIRS-96 (Lee and Geller 1996a) shows high speed inheritance in our parallel materialized transitive closure. Thus, the CHS is a distributed, pointerless, multi-relational, materialized transitive closure. The different sites of the distributed representation are accessed and queried using agent technology. Application of our previous work to data mining and the use of agent technology for distributed processing of the concept hierarchy are the new results of this paper.

It is impossible to explain in detail the construction, maintenance and update of the CHS which was developed in a chain of publications (Lee and Geller 1996a, 1996b; Lee 1997). The following example is only intended to give a flavor of the representation. In Figure 1 (H1) a transitive relationship between Action Games and Star Trek Voyager can be verified by looking at the number pairs of Action Games and Star Trek Voyager. Because [10 10] is contained in [6 12] we can conclude that Star Trek Voyager is indeed Action Games.

In this paper, we present **(1)** how to integrate a concept hierarchy with association rules and how to transform the integrated knowledge into the CHS; **(2)** an inference model for data mining and query processing that supports generalization/specialization of association rules and association rules through multiple and distributed hierarchies; **(3)** distributed knowledge discovery and interesting query types with CHS; **(4)** a

prototype of our yellow page system based on multiple-agent techniques.

In Section 2 of this paper, a brief summary of related work in data mining is presented. In Section 3, our representational framework is presented. In Section 4, our approach to agent-based distributed knowledge discovery is discussed. Section 5 contains our conclusions.

Combined Hierarchical Set for Data Mining

In this section, we introduce several reasoning models to uncover distant associations through multi-distance, multi-level, cross-hierarchy combined reasoning. The generalization of association rules through concept hierarchies is used to discover additional useful association rules from transaction databases. We will follow Agrawal's terminology and call an ancestor in a DAG (Directed Acyclic Graph) a *predecessor*. Similarly, a *successor* is a descendant of a node.

Srikant *et al.* (1995) use a notion of generalized association rule: $X \Rightarrow Y$, where $X \subset \mathcal{I}$ (itemset) and $Y \subset \mathcal{I}$ (itemset) with $X \cap Y = \emptyset$ and no item in Y is a predecessor of any item in X. The generalized association was modeled to find "interesting rules" without any redundant information such as $x \Rightarrow predecessor(x)$. Note that $\Rightarrow$ denotes an association rule, $\rightarrow$ denotes *IS-A** (zero or more *IS-A* relations) and $\rightsquigarrow$ denotes combined association relations.

A center point of our previous work was efficient *pure transitive closure reasoning.* This form of reasoning looks as follows. Assume a predecessor σ of a node τ. τ is reachable by a path of relations R^x (all of the same type) from σ. Then the relation R^x holds between σ and τ. As an example of pure transitive closure reasoning, from *Timeline IS-A PC Games;* and *PC Games IS-A Software;* we infer that *Timeline IS-A Software;*

The following definitions of generalization and aggregation of association rules are defined based on pure transitive closure.

Definition 1: Relational *IS-A* Notation Below, we will be using $\sigma\ R^s\ \tau$ as relational notation for $\sigma \rightarrow \tau$ where τ is a predecessor of σ.

Definition 2: Generalized Association If ($X \rightarrow Y$ and $Y \Rightarrow Z$) or ($X \Rightarrow Y$ and $Y \rightarrow Z$) or ($X \rightarrow A$ and $A \Rightarrow B$ and $B \rightarrow Z$), then $X \Rightarrow Z$. The step from X to Y or Y to Z in the associations is referred to as "generalization." In the general case, if it holds that ($X_1\ R^s\ X_m$) and ($X_m \Rightarrow X_l$) and ($X_l\ R^s\ X_n$) [by Definition 1] then $X_1 \Rightarrow X_n$ where $m \neq l \neq n$.

For example, if *Star Trek Voyager IS-A Teen Action Game* and *Microsoft JS-326 IS-A Joy Stick* and *if a customer buys a Star Trek Voyager game then he/she will also buy a Microsoft JS-326* then we can infer that *if a customer buys Teen Action Game then she will also buy a Joy Stick.*

In order to model the yellow page application we need an additional formal construct. Intuitively, we are now aggregating over several cyber-shopping malls. Each mall has its own hierarchy (H_i).

Definition 3: Aggregated Association If H has multiple parts $H_1, H_2, \ldots H_n$, and X and Y are associated as $X \Rightarrow Y$ in every H_i, which can be rewritten as ($H\ R^p\ H_i$) and ($X \Rightarrow Y$ in H_i) for $1 \leq i \leq n$ [by Definition 1], then we define an "aggregated association" between X and Y in H.

For example, *at Yahoo Mall, if a customer buys a Sony flat TV set then she will also buy a Sony DVD player;* and *at Cybernet Mall, if a customer buys a Sony flat TV set then he/she will also buy a Sony DVD player;* then we can infer that *at Online Mall, if a customer buys a Sony flat TV then she will also buy a Sony DVD player;*

We are modeling the different web-accessible shopping malls as parts (*Part-of*) the distributed shopping mall for which we are constructing the yellow pages. The *Part-of* relation has been well supported in our previous work (Halper et al. 1998) as has the combined *IS-A – Part-of* transitive closure reasoning.

Definition 4: Combined Transitive Closure Reasoning If several relation types are involved, such as *Part-of* and *IS-A* above, we refer to the transitive reasoning as combined transitive closure reasoning, specified as $R^{\{s,p,c\}}$. We use a hierarchical priority ordering among the hierarchical relations (Winston et al. 1987), such that combined inclusion relation syllogisms are valid if and only if the conclusion expresses the lower relational priority appearing in the premises. This has been discussed in our previous papers (Lee and Geller 1996a, 2000).

As an example of combined transitive reasoning, from *if a PC monitor IS-A screen* and *if a PC monitor is Part-of a PC* and *if information on the PC is Contained-in the Yahoo Mall* we infer that *information on a screen is Contained-in the Yahoo Mall.*

Definition 5: Combined and Constraint-based Association If it holds that $X_1\ R^{\{s,p,c\}}\ X_m$ [by Definition 4] and $X_m\ [c]\Rightarrow X_n$ where constraint c is satisfied, then we infer a "combined and constraint-based association" $X_1\ [c]\rightsquigarrow X_n$. See Section 3 for details of the constraint.

As an example of combined and constraint-based association, from *if a Star Trek Voyager game is Assoc-with an Escape from Monkey Island game and is rated as a teen game* and *if information on a Star Trek Voyager is Contained-in the teen page of Amazon Mall* then *information on an Escape from Monkey Island game is most likely to be Contained-in the teen page of Amazon Mall.*

Construction of Combined Hierarchical Set

In this section, we present details of the CHS, which supports extended inference as described above. The CHS is a materialized form of a combined relational hierarchy. A *combined relational hierarchy* allows multiple relations to coexist in one hierarchy (Figure 1).

This permits transitivity through several different relationships (e.g., *Part-of* and *IS-A*).

In this section we show (1) the primary building blocks of CHS; (2) how the CHS encoding supports the inferences mentioned in Section 2; and (3) what feature of the encoding makes it possible to map a hierarchy onto a space of distributed agents.

Dealing with (1), we have shown reasoning algorithms in our previous research (Lee and Geller 1996b) for relational hierarchies combining *IS-A*, *Part-of*, and *Contained-in* which are now extended to association rules. To deal with the other questions, a three step mapping (Figure 2) that we have developed in (Lee and Geller 1996a) is greatly extended in this paper.

The distributed CHS representation is based on a directed acyclic graph (DAG). Every node is annotated with data elements, $<\mathcal{N}, \Re, \mathcal{L}, \Omega, \mathcal{H}, \Phi>$ consisting of a Set of Number Pairs ($\mathcal{N}$), a Relation Type ($\Re$), a Node Level ($\mathcal{L}$), a Constraint Rule (Ω), a Hierarchy Identifier ($\mathcal{H}$), and an Inference Value Pair (Φ).

Set of Number Pairs ($\mathcal{N}$): It is possible to perform transitive closure reasoning and combined association with a set of DAG nodes, but without the DAG links, by using a materialized transitive closure, which consists of sets of number pairs attached to the nodes. In previous work we and others have extensively published on this method (Schubert 1979; Agrawal et al. 1989; Lee and Geller 1996a, 1996b).

Relation Type ($\Re$): Intuitively, when combining several kinds of relationships (*IS-A*, *Part-of*, *Contained-in*, *Assoc-with*) into one single hierarchy, confusion between those relationships would occur. Thus we annotate each relationship by a relation type to maintain its identity in the CHS. Space limitations force us to refer to previously published material for formal details (Lee and Geller 1996b).

Node Level ($\mathcal{L}$): The *node level of a node A* represents the level at which A is located in a DAG G. The level of a node is determined based on a spanning tree of G. The *node level* is used to measure the degree of generalization or specialization and to support multi-level generalization and transitivity.

Constraint Rule (Ω) is used to restrict association between nodes by specifying conditions. We define the rule as $R = R_i\ Op_2\ R_j \mid R_i \mid Op_1\ R_i$ and $R_i = Attr\ AOp\ Attr$-or-Val where $Attr$ is an attribute, $Op_1 \in \{\text{NOT}\}$, $Op_2 \in \{\text{AND, OR}\}$, $AOp \in \{\leq, <, \neq, =, \geq, >\}$, and $Attr$-or-Val is either an attribute or its value. For instance, a Teen game is for someone whose age is ranged between 13 and 19 (Age $\geq$ 13 AND Age $\leq$ 19).

Hierarchy Identifier ($\mathcal{H}$) is used to distinguish among hierarchies when multiple hierarchies are involved in inference during data mining.

Inference Value Pair (Φ) represents the support and confidence values at a node in combined association. The definition of Φ follows Srikant *et al.*'s definitions for expected support and confidence of generalized association (Srikant and Agrawal 1995).

The CHS is constructed based on our three step mapping (Figure 2). In the first step, we combine a concept hierarchy of the real world and a set of discovered association rules (with high confidence values) into a combined relational hierarchy by representing association rules as Assoc-with and Aggregation relations annotated with support and confidence values. By this step, the *hierarchy identifiers* ($\mathcal{H}$) and *inference value pairs* (Φ) are computed. Refer (Lee and Geller 1996a, 1996b) for details.

In the second step (Figure 2), the *hierarchy* of nodes is mapped onto a *set* of those nodes, so that every node is annotated with one or more number pairs. Every relation has an associated relation type and an associated priority value (Lee and Geller 1996b). The relation type of a number pair is sometimes transformed during propagation. Since the priority of the relation type S (*IS-A*) is the lowest among all the relation types, the relation type of the propagated pair will always be replaced by the relation type R associated with the link. Refer to (Lee and Geller 1996b) for more details on this. By this step, the set of *number pairs* ($\mathcal{N}$), the *relation type* ($\Re$) and the *node level* ($\mathcal{L}$) are computed. In (Lee and Geller 1996b), we have shown that our representation (Figure 1) is sufficient for the efficient *parallel* execution of all necessary retrieval and update operations (Lee and Geller 1996a).

In the third step (Figure 2), the node set and the associated $<\mathcal{N}, \Re, \mathcal{L}, \Omega, \mathcal{H}, \Phi>$ annotations (CHS) are mapped onto the distributed agents residing in a distributed computing environment. After Step 2 the links are no longer necessary and may be omitted. Since all relational information is contained in the number pairs, the combined hierarchical set representation makes it easy to represent concept hierarchies on distributed agents. The combined hierarchical set representation is completely order independent, i.e., combined hierarchical sets are used without loss of relevant hierarchy information. This simplifies the distributed knowledge extraction and update operations necessary to maintain the concept hierarchy. More details on our primary representation of CHS can be found in (Lee and Geller 1993, 1996a). The bi-directional arrows between combined hierarchical set and distributed subsets mean that CHS would be updated with feedback or changes from distributed agents (Figure 2).

Agent-based Distributed Knowledge Discovery

Distributed data mining is highly attractive due to the capability of handling huge amounts of data in modern distributed database environments (Cheung et al. 1996). A prototype of our system was implemented using IBM's Aglet agent building framework (Lange et al. 1997) on a network of Windows NT computers. Figure 3 shows the interface of our prototype. The described inference models in Section 2 are designed and implemented with agent-based distributed computing. The prototype supports CHS-based data mining,

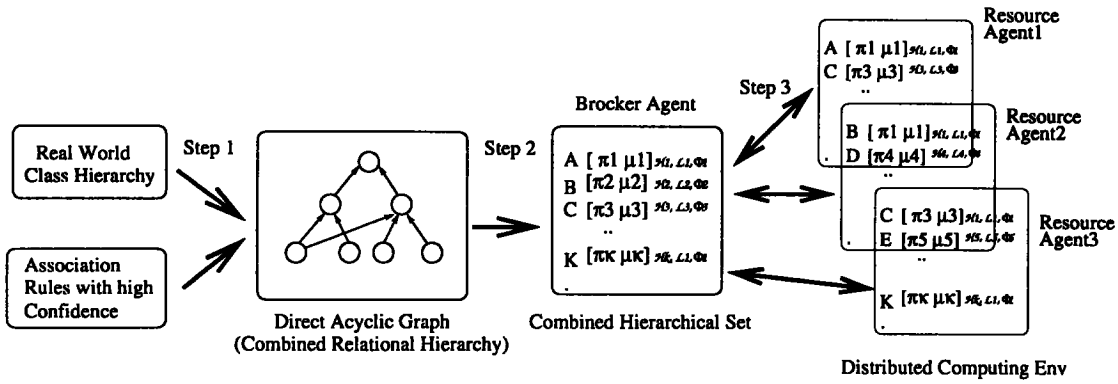


Figure 2: Three Step Mapping

owing to its feature of mobility needed for distributed computing. CHS is designed such that the combined hierarchical sets of large item sets can be efficiently distributed and the amount of intersite data exchanges is minimized. Each agent maintains its own part of the CHS corresponding to any changes of its local database of the cyber-shopping mall, and supports interactive queries from other agents. Each agent shares discovered knowledge with other agents for interesting query answering. The system answers to a query by collaborating agents: broker agent, resource agent, and query agent. The retrieval capability based on CHS and multiple agents is greater than the sum of the capabilities of the individual information sources.

The query agent accepts queries from a user on information available in cyber-shopping malls and presents the results in a structured form. Hierarchically structured information allows users easy access to an appropriate level of abstraction of shopping mall information. The query agent presents an intuitive interface dynamically generated from an item list available in connected cyber shopping malls and allows users to specify attributes for chosen items. For instance, the user can choose PC game as an item and specify its attributes and values, such as "teen" as an age group or "action" as a game genre and its constraints such as price $\leq$ \$50, etc. The query results are presented in an integrated structural form with the support of the broker agent (See Figure 3). The query agent unifies the query results obtained from specific merchants and cyber-shopping malls. For instance, for a query "which game controllers are compatible with my Star Trek Voyager," the broker agent contacts a software resource agent and a hardware resource agent by knowing that a game is a software and a game controller is a hardware. The resource agent specialized in software retrieves information on Star Trek Voyager from a software database and H1 (Figure 1) and another resource agent specialized in hardware retrieves information on game controllers from a hardware database and H2 (Figure 1). Since H2 has information on an association between Star Trek Voyager and Joy Stick, the query agent can answer the query by integrating knowledge from the distributed sources. Furthermore, the query agent can suggest some items based on an identified user model, such as which design was more popular with the young generation and which brand was sold most this season.

The broker agent retrieves the appropriate information sources to satisfy a given query. The broker agent also maintains the integrated global view of domain knowledge from multiple cyber-shopping malls. The integrated knowledge is encoded as CHS which provides relationships between information for retrieval and presentation of discovered information. The agent determines strategies on gathering and integrating information for the query: retrieval from predefined cache or dynamic retrieval from cyber-shopping malls. The broker agent generates the combined hierarchical set and dynamically updates the encoding based on the evolving information from cyber-shopping malls.

The resource agents manage information available in every cyber-shopping mall, retrieve resources and verify the relationships between them locally. These agents maintain a local CHS representation and a database to store the detailed product information. These resource agents are specialized with the basic knowledge relevant to the domain. For instance, the agent knows that the cost of PC games would be at least \$30 while the cost of a PC would be at least \$800, and how to compute the delivery charges.

Conclusion

We described a yellow page service based on combined relationships in data mining. For this, we introduced a framework which can be used for extracting useful knowledge and performing interesting queries with increased inferencing power. The basis for implementing this framework is an efficient pointerless representation, called CHS, which combines an *IS-A* hierarchy with *Part-of*, *Contained-in*, *Assoc-with* and aggregation knowledge. Owing to the nature of this representation, it is possible to distribute a CHS on a distributed computing environment. We also implemented a query interface on top of CHS using multiple agent techniques.

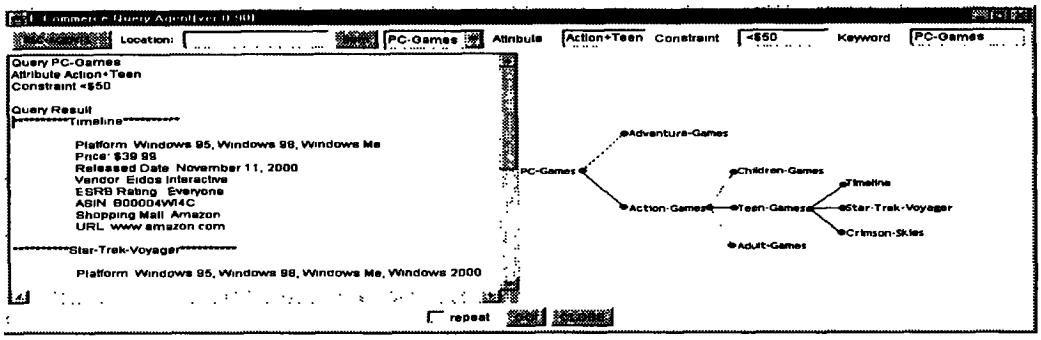


Figure 3: Query Agent Interface

References

Agrawal, R.; Borgida, A.; and Jagadish, H. V. 1989. Efficient management of transitive relationships in large data and knowledge bases. In *ACM SIGMOD International Conference on the Management of Data*, 253–262.

Cheung, D.; Han, J.; Ng, V.; Fu, A.; and Fu, Y. 1996. A fast distributed algorithm for mining association rules. In *Proceedings of the Fourth International Conference on Parallel and Distributed Information Systems*, 31–42.

Halper, M.; Geller, J.; and Perl, Y. 1998. An OODB part-whole model: semantics, notation, and implementation. *Data & Knowledge Engin.* 27(1):59–95.

Han, J., and Fu, Y. 1995. Discovery of multiple-level association rules from large databases. In *Proceedings of the 21st International Conference on Very Large Data Bases (VLDB'95)*, 420–431.

Han, J.; Fu, Y.; Wang, W.; Koperski, K.; and Zaiane, O. 1996. DMQL: A data mining query language for relational databases.

Han, E.; Karypis, G.; and Kumar, V. 1997. Scalable parallel data mining for association rules. *ACM. Sigmod Record (Acm Special Interest Group on Management of Data)* 26(2):277–288.

Han, J. 1995. Mining knowledge at multiple concept levels. In *Proceedings of the Fourth International Conference on Information and Knowledge Management (CIKM'95)*, 19–24.

Lange, D.; Oshima, M.; Karjoth, G.; and Kosaka, K. 1997. Aglets: Programming mobile agents in java. In *In 1st International Conference on Worldwide Computing and Its Applications (WWCA'97).*

Lee, Y., and Geller, J. 1993. Representing transitive relationships with parallel node sets. In Bhargava, B., ed., *Proceedings of the IEEE Workshop on Advances in Parallel and Distributed Systems.* Los Alamitos, CA: IEEE Computer Society Press. 140–145.

Lee, Y., and Geller, J. 1996a. Constant time inheritance with parallel tree covers. In *Proceedings of the Florida AI Research Symposium (FLAIRS)*, 243–250.

Lee, Y., and Geller, J. 1996b. Parallel transitive reasoning in mixed relational hierarchy. In *Proceedings of the Conference on Knowledge Representation and Reasoning*, 576–587.

Lee, Y., and Geller, J. 2000. Efficient transitive closure reasoning in a combined class/part/containment hierarchy. *Accepted by Knowledge and Information Systems.*

Lee., Y. 1997. *Massively Parallel Reasoning in Transitive Relationship Hierarchies.* Ph.D. Dissertation, CIS Department, New Jersey Institute of Technology.

Li, B., and Shasha, D. 1998. Free parallel data mining. *ACM. Sigmod Record: Management of Data* 27(2):541–543.

Nodine, M. 1998. The infosleuth agent system. In *Cooperative Information Agents II. Learning, Mobility and Electronic Commerce for Information Discovery on the Internet. Second International Workshop.*

Prodromidis, A., and Stolfo, S. 1998. Pruning meta-classifiers in a distributed data mining system. In *Proceedings of the First National Conference on New Information Technologies*, 151–160.

Schubert, L. K. 1979. Problems with parts. In *Proc. of the 6th International Joint Conference on Artificial Intelligence.* Los Altos, CA: Morgan Kaufmann Publishers. 778–784.

Silverstein, C.; Brin, S.; and Motwani, R. 1998. Beyond market baskets: generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery* 2(1):39–68.

Skoron, A., and Stepaniuk, J. 1999. Information granules in distributed environment. In *New Directions in Rough Sets, Data Mining, and Granular-Soft Computing.*, 357–365. Springer-Verlag.

Srikant, R., and Agrawal, R. 1995. Mining generalized association rules. In *Proceedings of the 21st Int'l Conference on Very Large Databases.*

Stolfo, S.; Prodromidis, A.; Tselepis, S.; Lee, W.; Fan, D.; and Chan, P. 1997. Jam: Java agents for meta-learning over distributed databases. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining.*

Winston, M. E.; Chaffin, R.; and Herrmann, D. 1987. A taxonomy of part-whole relations. *Cognitive Science* 11(4):417–444.