A data warehousing approach for building recommender systems

Z. Chen
Department of Computer Science
University of Nebraska at Omaha
Omaha, NE 68182-0500

Abstract

A data warehousing approach for recommender systems is proposed. We sketch an architecture for integrated OLAP and data mining in data warehousing environments, and argue why this architecture can be extended for building recommender systems. Since producing recommendations can be considered as conceptual query answering, the relationship between conceptual query answering and intensional answers is also briefly examined.

Introduction: Rationale of a new approach

Recommender systems have appeared, based on a synthesis of ideas from artificial intelligence, human-computer interaction, sociology, information retrieval, and the technology of the World Wide Web (WWW). Recommender systems assist and augment the natural process of relying on friends, colleagues, publications, and other sources to make the choices that arise in everyday life (Workshop 98).

Our interest in recommender systems is stemmed from recent research related to integrated On-Line Analysis Processing (OLAP)/data mining in data warehousing environments. (For a nice survey of OLAP and data warehousing, see Chaudhuri and Dayal 1997. For basic literature of data mining, see Piatetsky-Shapiro and Frawley 1991; Fayyad, Piatetsky-Shapiro, Smyth and Uthurusamy 1996).

An important feature of data warehousing is its connection with the Internet. The widespread adoption of the Internet technology will profoundly affect OLAP. The basic concepts of data warehousing and aggregation have made their way on the Web. Some of the most popular Web sits on the Internet are basically data warehouses. Since recommender systems heavily depend on information retrieval and technology of WWW, the close connection between WWW and data warehousing makes the data warehousing architecture a natural choice for recommender systems.

Although not necessarily the full strength of data warehousing techniques is needed for recommender systems, and although recommender systems have some functions beyond data warehousing, the data warehousing architecture can be used as the starting point for construction of recommender systems.

In the rest of this paper, we will provide a little more detail for the proposed approach.

An architecture for integrated OLAP and data mining

The architecture we have based on can be considered as an revision and extension of an integrated OLAP/data mining architecture proposed by Han (1997, 1998). The resulting architecture is shown in Fig. 1. (All lines indicate connections in both directions.)

This architecture differs from Han's proposal in numerous ways:

(a) Our architecture is more general in that our discussion is concerned with materialized views (which are stored views) in general, rather than restricted to data cubes (which can be viewed as a restricted form of materialized views) as in the original diagram.

(b) We have considered both on-line data mining (OLDM) as well as off-line data mining (OFLDM). The distinction of these two kinds of data mining is necessary, because the latter differs from the former in that it may involve more time consuming process, and it may also use historical data. Note data mining in its "traditional" sense usually belongs to the latter. Also note the similarity between the result obtained from previous data mining on the one hand and materialized views on the other, because both are obtained from previous data and both are ready to be used for answering users' *ad hoc* queries.

(c) Since our focus is on the connection of these components (such as OLAP, MV, OLDM, OFLDM, etc.), we will not include application programs interface (API) into current discussion.

(For a more detailed discussion on our proposed architecture, see Chen 1998a).

Building recommendation systems as extended warehouses

The architecture introduced above has good potential of being suitable for recommender systems, due to several reasons.

- The data warehouse environment serves as a large, common knowledge base in many knowledge domains relevant to potential users' interests, and the knowledge can be accessed using information retrieval techniques.
- The architecture is able to observe user behavior and derive user intention/ preference, because the On-Line Data Mining module performs not only on-line intelligent analysis of data, but also analysis of the user profile through queries the user submitted.

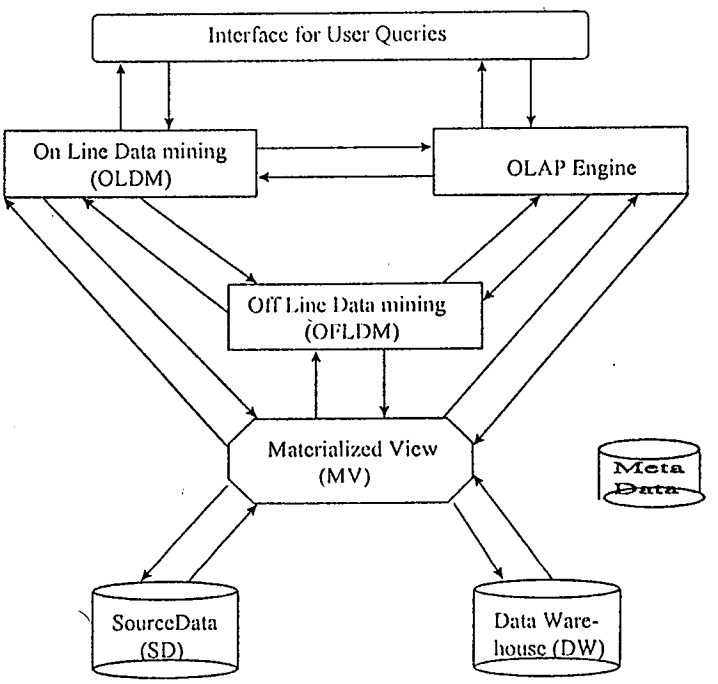

Fig. 1

- In addition, users' requests for recommendations can be treated as conceptual queries, and they can be answered by taking advantage of intensional answers (Motro 1994). The intensional answers can be produced using both on-line and off-line data mining techniques. Since many users can access the data warehouse at the same time, they can provide advice to each other.

Since the last issue (conceptual query answering) has direct connection with recommender systems, in the following we take a brief examination on this issue.

Conceptual answering in data warehousing environment

We should first examine the nature of the connections between conceptual queries and intensional answers in database management systems. Both of these two concepts are closely related to knowledge discovery in databases (KDD) and data mining, which are concerned with the overall process and specific techniques of the nontrivial extraction of implicit, previously unknown, and potentially useful information from data.

An *intensional answer* to a query is a set of characterizations of the set of database values that satisfy the query (the actual data retrieved are referred to as the extensional answer.) Intensional answers are derived entirely from the extensional information in the database (Motro, 1994; Han *et al.*, 1996). For example, for the query of finding excellent students, an intensional answer could be some common features (such as good GPA) shared by these students, rather than the names of these students.

The task of *conceptual* (or *intelligent*) query answering is to map users' conceptual queries to actual database queries and to produce answers for the users' queries. Conceptual queries have been extensively studied in Information Retrieval (IR) community, and have drawn increasingly attention from database research community as well. An example of conceptual query used in recommender systems is:

> What kinds of *expensive houses* should be recommended to a *young, rich couple* who wish to re-locate in West Omaha?

Note that in thie query terms such as "expensive houses" and "young, rich couple" are not database attributes nor actual values, thus need to be mapped to actual database values.

As for the format of the answers for conceptual answers, we follow the proposal made by Imielinski (1987). In his research, the structure of an answer is identical to the structure of database itself, with an extensional part and an intensional part. Such answers have both conceptual and computational advantages. In the previous example of home-buyers, we may answer this query by retrieving all the actual tuples (which is the extensional answer) along with a set of characteristics of recommended houses (which is the intensional answer). In fact, an answer could also be of a mixed format. For example, the above query could be answered by "All the new houses west of 180 street and the house located at 22400 Dodge Street." The first part of this answer is intensional while the second part is extensional. Therefore, just like answers for conventional queries could be extensional or intensional, answers for conceptual queries may also fall in two categories: to find actual tuples, or to find descriptive features for the conceptual information needs the user requested.

Conceptual query can be answered directly if corresponding intensional answers exist. For cases where some intensional answers not available for answering conceptual queries, we can use a query-invoked process to produce necessary intensional answers to answer the conceptual query, thus combining lazy and eager strategies. We may also rewrite a conceptual query, thereby converting the problem of conceptual query answering into the existing work of answering queries using materialized views. The integrated architecture in a data warehousing environment makes this approach feasible. (For a more detailed discussion on conceptual query and intensional answers, see Chen 1998b).

Levy *et al.* (1995) discussed issues related to answering query using materialized views, including the problem of finding a rewriting of a query that uses the (materialized) views, the problem of finding minimal rewritings, and finding complete rewritings (ie., rewriting that use only the views). Note that their work does not consider conceptual queries, but some important methods and results may be incorporated into our approach.

Conclusion

Important issues related to our approach include how to exploit advantages of the proposed architecture summarized earlier, as well as how to deal with various problems may be encountered. For example, in a data warehousing environment, a significant portion of information filtering can be carried out by using customerized views. But details are yet to be explored.

References

Chaudhuri, S. and Dayal, U. 1997. An overview of data warehousing and OLAP technology, *SIGMOD Record*, 26(1): 65-74.

Chen, Z. 1998a. An integrated architecture of OLAP and data mining. Working paper, University of Nebraska at Omaha.

Chen, Z. 1998b. A framework for conceptual query answering using intensional answers. Working paper, University of Nebraska at Omaha.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, R. (eds.) 1996. *Advances in Knowledge Discovery and Data Mining.* Menlo Park, CA: AAAI/MIT Press.

Han, J. 1997. OLAP mining: an integration of OLAP with data mining. In *Proc. 1997 IFIP Conf. on Data Semantics (DS-7)*, 1-11.

Han, J., 1998. Towards On-Line Analytical Mining in large databases, *SIGMOD Record.* To appear.

Han, J. Huang, Y., Cercone, N. and Y. Fu, Y. 1996. Intelligent query answering by knowledge discovery techniques. *IEEE Transactions on Knowledge and Data Engineering*, 8(3): 373-390.

Imielinski, T. 1987. Intelligent query answering in rule based systems, *J. Logic Programming*, 4(3): 229-257.

Levy, A. Y., Mendelzon, A. O., Sagiv, Y. and Srivastava, D. 1995. Answering queries using views. In *Proc. PODS*, 95-104.

Motro, A. 1994. Intensional answers to database queries. *IEEE Trans. Knowledge and Data Engineering*, 6(3): 444-454.

Piatetsky-Shapiro, G. and Frawley, W. J. (eds.) 1991. *Knowledge Discovery in Databases.* Menlo Park, CA: AAAI/MIT Press.

Workshop 98, 1998. Recommender systems (AAAI Workshop Program Call for papers).