Towards an Integrated Rough Set and Data Modelling Framework for Data Management and Knowledge Extraction

Salem Chakhar[1,2] and Zouhaier Brahmia[3]

[1]Portsmouth Business School, Faculty of Business Law, University of Portsmouth, Portsmouth, UK.
salem.chakhar@port.ac.uk
[2]Centre for Operational Research and Logistics, University of Portsmouth, Portsmouth, UK.

[3]MIRACL laboratory, Faculty of Economics and Management, University of Sfax, Tunisia.
zouhaier.brahmia@fsegs.rnu.tn

Abstract. Data models and database systems are excellent tools to store and manage data. However, most of available data models and database systems lack effective techniques to extract relevant knowledge form raw data. Combining data modelling approaches and machine learning techniques represent a promising road to design and develop integrated data management and knowledge extraction systems. In this paper, first we propose a Rough Semantic data Model (RSM) based on a coupling between semantic data modelling and rough sets concepts. Then, we introduce the design of a framework that supports RSM and provides data management and knowledge extraction functionalities.

Keywords: Database, Rough Set Theory, Data Model, Knowledge Extraction.

1 Introduction

In the database world, there are three main data models to design a database: the relational data model [1], the object-oriented data model [2], and the semantic data model [3,4]. A relational model is a set of relations. Each relation has a distinct name and is a grid-like mathematical structure composed of columns, called attributes, and rows, called tuples. Each attribute (or a column of data) has a name and a data type. Each tuple (or a row) consists of a set of values such that each value per attribute. A relation is also known as a "table" in the standard SQL language and existing relational database management systems (DBMSs).

Data models and database systems are excellent tools to store and manage data. However, most of available data models and database systems lack effective techniques to extract relevant knowledge form raw data. Combining data modelling approaches and machine learning techniques represent a promising road to design and develop integrated data management and knowledge extraction systems. In fact, since the last decade, several research papers (like [5-14]), which have been published in

well-known conferences and journals, have dealt with either machine learning in DBMSs or data management in machine learning systems.

In this paper, we propose a Rough Semantic data Model (RSM) based on a coupling between semantic data modelling and rough sets [15] concepts. The rough sets theory is a mathematical tool for the analysis of a vague description of objects. It operates on a decision table composed of a set of objects described by a set of attributes. It procedures a collection of decision rules derived from rough approximations of subsets identified with decision classes. Notice that rough sets have been used in some research work that deal with database or data management problems, like [16-19], but to the best of our knowledge they have not been used in a work that combines machine learning and database management.

The paper also introduces a framework that supports the RSM model and provides data management and knowledge extraction functionalities.

The rest of the paper is organized as follows. Sec. 2 presents some basic concepts of the rough set theory. Sec. 3 proposes RSM. Sec. 4 provides a general overview of an integrated data management and knowledge extraction framework. Sec. 5 concludes the paper.

2 Principles of Rough Set Theory

Let U be a non-empty set of objects (the universe) and D be a non-empty, finite set of attributes such that $q:U \rightarrow Vq$, where Vq is the domain of attribute $q \in D$. With any subset $K \subseteq D$ there is an associated equivalence relation, called K-indiscernibility relation $IND(K)$ such that:

$$IND(K) = \{(x,y) \in U^2 \mid q(x) = q(y), \forall q \in K\}$$

This relation $IND(K)$ is partitioning U into a set of equivalence classes which is denoted by $U/IND(K)$ or simply U/K. The equivalence classes induced by relation $IND(K)$ are denoted $[x]_K$. Shortly, $[x]_K \in U / K$ is the equivalence class containing x. In RST, any subset $M \subseteq U$ is defined in terms of the elementary sets (equivalence classes) of the partition U/K by lower and upper approximations as follows:

$$K_*(M) = \{x \in U \mid [x]_K \subseteq M\}$$
$$K^*(M) = \{x \in U \mid [x]_K \cap M \neq \phi\}$$

The sets $K_*(M)$ and $K^*(M)$ (or simply M_* and M^*) are called the lower and the upper approximations of M respectively. Therefore, $M_* \subseteq M \subseteq M^*$. The difference between the upper and lower approximations is called the boundary of M and is denoted by $BN_K(M) = M^* - M_*$.

3 Rough Semantic Data Model

3.1 Basic Concepts

Let E be the universe of discourse. A rough entity e in E is a natural or artificial entity that one or several of its properties are rough. At the extensional level, a rough class K in E is a collection of rough entities having some similar properties: $K=\{(e, La(e)):e\in E\wedge La(e)\in T\}$, where $La:E\rightarrow T$ is a mapping from E to a set $T=\{Cl_h:h=,1,2,\ldots p\}$ of labels.

Each RSM attribute is characterized by its name, data type and domain. A data type may be exact (e.g. integer, char) or rough. The domain of an attribute is the set of values the attribute may take. Attributes may be single-valued, i.e., the attribute cannot have more than one value at a given time, or multi-valued, i.e., the attribute can have several values at a given time. In general, the values of a multi-valued attribute may be related with different logical connectors (e.g. AND, OR, XOR) but this is will not be dealt with here.

Two crisp rough classes can be extracted from rough class K:

- K_L Lower approximation class containing objects that certainly belong to K;
- K_U Upper approximation class containing objects that may belong to K.

The boundary K_B of set rough class K is defined as the set difference between the lower and upper approximations of K, i.e., $K_B = K_U - K_L$.

3.2 Quality of Classification and Accuracy of Approximation

The quality of classification of partition T by means of condition attributes set is defined as the ratio of all correctly classified objects to all objects in E. The accuracy of approximation of decision classes is thus computed as the ratio between the number of objects in the lower approximation and the number of objects in the upper approximation.

3.3 Attribute Reducts and Core

A reduct is a subset of condition attributes that can, by itself, fully characterize the knowledge in the decision table. A reduct is minimal (with respect to inclusion) subset of condition attributes in the sense that no attribute can be removed from the reduct without deteriorating the quality of approximation. The intersection of all reducts is called core.

3.4 Data Structuring

The data used within RSM care organized into different subsets:

- Information Table: the original dataset with unlabelled objects.

- Decision Table: Information table with labelled objects.
- Learning Set: A subset from the Decision Table used for training purposes.
- Validation Set: A subset from the Decision Table used for validation purposes.
- Testing Set: A subset from the Information Table with unlabelled objects or new used testing purposes.

3.5 Knowledge Extraction

Different induction algorithms can be used to extract knowledge from the Decision Table. The most popular rule induction algorithm is LEM which generates a minimal set of rules. The extracted knowledge take the form of a set of If-then decision rules. A decision rule is a consequence relation relating a set of conditions (premise) and a conclusion (decision). Each elementary condition is built upon a single condition attribute while a conclusion is defined as assignment to decision classes.

Decision rules are evaluated through a set of quantitative measures including support, strength, accuracy, coverage and length. The obtained rules need to be validated before put into practice. Three validation techniques are commonly used: direct analysis, reclassification and cross-validation. The validated decision rules can then be used to classify unseen decision objects.

4 Towards an Integrated Data Management and Knowledge Extraction Framework

In Figure 1, we propose the high-level architecture of a framework that supports RSM and provides data management and knowledge extraction functionalities. As shown in this figure, two categories of users are considered: DBMS oriented users and Machine Learning (ML) oriented users. This categorisation is not strict as the same user may be concerned by both data management and knowledge extraction.

The core of the proposed architecture is an extended SQL - Machined Learning (SQL-ML) language that supports the conventional SQL query language used by DBMS with additional operators and operations devoted to support knowledge extraction. The main new operations are as follows:

- Approximate <decision_table>: it generates the lower and upper approximations.
- Infer <decision_table>: it infers the decision rules.
- Reduct <decision_table|information_table>: it calculates all attribute reducts.
- Core <reducts_set|decision_table|information_table>: it calculates the core which is the intersection of all attribute reducts.

- Classify <information_table> with <classifier>: it applies the decision rules to classify the objects.
- Cross-Validate <classifier><decision_table><number_of_folders>: it applies the cross-validation technique.

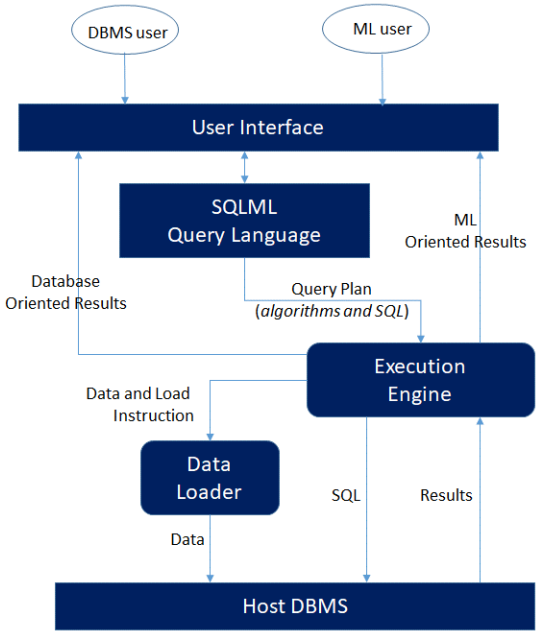

Fig. 1. The Architecture of an Integrated Data Management and Knowledge Extraction Framework

5 Conclusion

In this, we propose (i) RSM, a Rough Semantic data Model that relies on coupling between semantic data modelling and rough sets concepts, and (ii) a framework that supports RSM and provides both data management and knowledge extraction functionalities. In our future work, we will develop a layered system that supports our framework and shows its feasibility. Such a system will help us to experimentally evaluate the usability and the performances of our framework.

References

1. Embley, D.W. (2018). Relational Model. In: Liu, L., Özsu, M.T. (eds) *Encyclopedia of Database Systems (2nd edition)*, pp. 3149-3154. Springer, New York, NY.
2. Urban, S.D., Dietrich, S.W. (2018). Object Data Models. In: Liu, L., Özsu, M.T. (eds) *Encyclopedia of Database Systems (2nd edition)*, pp. 2525-2532. Springer, New York, NY.

3. Bouaziz, R., Chakhar, S., Mousseau, V., Ram, S., and Temouldi, A. (2007). Database design and querying within the fuzzy semantic model. *Information Sciences, 177*(21), pp. 4598-4620.
4. Embley, D.W. (2018). Semantic Data Model. In: Liu, L., Özsu, M.T. (eds) *Encyclopedia of Database Systems (2nd edition)*, pp. 3391-3393. Springer, New York, NY.
5. Li, X., Cui, B., Chen, Y., Wu, W., & Zhang, C. (2017). MLog: Towards declarative in-database machine learning. *Proceedings of the VLDB Endowment, 10*(12), 1933-1936.
6. Domingos, P. (2018, May). Machine learning for data management: problems and solutions. In *Proc. of the 2018 Intl' Conf. on Management of Data* (pp. 629-629).
7. Fard, A., Le, A., Larionov, G., Dhillon, W., & Bear, C. (2020, June). Vertica-ML: Distributed Machine Learning in Vertica Database. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (pp. 755-768).
8. Jankov, D., Luo, S., Yuan, B., Cai, Z., Zou, J., Jermaine, C., & Gao, Z. J. (2020). Declarative recursive computation on an RDBMS: or, why you should use a database for distributed machine learning. *ACM SIGMOD Record, 49*(1), 43-50.
9. Jasny, M., Ziegler, T., Kraska, T., Roehm, U., & Binnig, C. (2020, June). Db4ml-an in-memory database kernel with machine learning support. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (pp. 159-173).
10. Abdennebi, A., Elakaş, A., Taşyaran, F., Öztürk, E., Kaya, K., & Yıldırım, S. (2022). Machine learning-based load distribution and balancing in heterogeneous database management systems. *Concurrency and Computation: Practice and Experience, 34*(4), e6641.
11. Schule, M., Lang, H., Springer, M., Kemper, A., Neumann, T., & Gunnemann, S. (2021, July). In-Database Machine Learning with SQL on GPUs. In *Proceedings of the 33rd International Conference on Scientific and Statistical Database Management* (pp. 25-36).
12. Png, A., & Helskyaho, H. (2022). Oracle Machine Learning in Autonomous Database. In: *Extending Oracle Application Express with Oracle Cloud Features* (pp. 139-191). Apress, Berkeley, CA.
13. Chai, C., Wang, J., Luo, Y., Niu, Z., & Li, G. (2022). Data management for machine learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*. DOI: 10.1109/TKDE.2022.3148237
14. Villarroya, S., & Baumann, P. (2022). A survey on machine learning in array databases. *Applied Intelligence*, 1-24.
15. Pawlak, Z. (1991). *Rough Sets: Theoretical Aspects of Reasoning About Data* (Vol. 9). Springer Science & Business Media.
16. Beaubouef, T., Petry, F. E., & Buckles, B. P. (1995). Extension of the relational database and its algebra with rough set techniques. *Computational Intelligence, 11*(2), 233-245.
17. Lin, T. Y. (1996, July). Rough set theory in very large databases. In *Symposium on Modeling, Analysis and Simulation, IMACS Multi Conference (Computational Engineering in Systems Applications)*, Lille, France (Vol. 2, pp. 936-941).
18. Suraj, Z., & Grochowalski, P. (2008). The rough set database system. In *Transactions on Rough Sets VIII* (pp. 307-331). Springer, Berlin, Heidelberg.
19. Kovalenko, I., Shved, A., Antipova, K., & Davydenko, Y. (2020). Structuring of a transaction database using the rough set theory. In: *CMIS* (pp. 278-287).