Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 9, September 2013, pg.198 – 204

RESEARCH ARTICLE

Performing the Data Reliability Estimation in a Data Warehouse Opened on the Web Enable Data Warehouse

K.Valarmathi[1], T.Selvakannan[2]

[1]Research Scholar, Department of Computer Science, Vivekanandha College, Elayampalayam, Tiruchengode, Tamil Nadu, India
[2]Assistant Professor, Department of Computer Science, Vivekanandha College, Elayampalayam, Tiruchengode, Tamil Nadu, India

[1] valarmathikuppusamy@gmail.com; [2] mrsellu@gmail.com

ABSTRACT: - This paper presents an ontology-driven workflow that feeds and queries a data warehouse on the Web enable data. Data are extracted from data tables in Web documents. As web documents are very heterogeneous in nature, a key issue in this workflow is the ability to assess the reliability of retrieved data. We first recall the main steps of our method to annotate and query Web data tables driven by domain ontology. Then we propose a clustering based generic method to assess data reliability from a set of criteria using the theory of belief functions. Customizable criteria and insightful decisions are provided.. Finally, we show how we extend the workflow to integrate the reliability assessment step.

Keywords --- Clustering Algorithm; Aggregation

I.INTRODUCTION

The huge amount of technical and scientific documents available on the Web includes many data tables. In addition to local data sources, they represent big potential external data sources for the data warehouse of a company dedicated to a given domain of application. To lighten the burden laid upon domain experts when selecting data from the data warehouse for a particular application, it is necessary to give them indicative reliability evaluations. In this paper, we present a framework to estimate the reliability of data tables collected from the Web. Compared to more ad-hoc estimation, the presented generic method can give insights to the expert as to why a particular data table is tagged as reliable or not reliable. Due to its generic nature, this method can be reused in other data warehouses using the semantic web recommended languages. Reliability estimation is an essential part of the Semantic Web architecture, and many research works [1] focus on issues such as source authentification, reputation, etc. For example, [2] advocates a multi- faceted approach to trust models. They propose an OWL based ontology of trust related concepts. The idea is to provide systems using the annotation power of a user community to collect information about reliability. Our approach is different, as we do not rely on users but rather on information about the Web data table origins to compute reliability estimations. Among methods proposing solutions to evaluate trust or data quality in web applications, the method presented in [3] is close to the method presented in the paper. It uses possibility theory evidence theory, whereas we base our method on evidence theory. Another difference is that in our approach global information is obtained by a fusion of multiple uncertainty models, while in [3] global information results from the propagation of uncertainty models through a aggregation function. Each method has its pro and cons: it is easier to integrate interactions between criteria in aggregation functions, while it is easier to retrieve explanations of the final result in our approach. In this paper, we details our method and its integration in @Web, along with the whole workflow used in @Web. The current version of @Web (see [4, 5]), a Web enabled data warehouse, has been implemented using the W3C recommended languages (see [6] for details about these languages): OWL to represent the domain ontology, RDF to annotate Web tables and SPARQL to query annotated Web tables. We first recall in Section 2 the purpose and architecture of the data warehouse. Section 3 details the proposed method to assess Web data table reliability. In Section 4, we show how this reliability assessment is presented and explained to the user. Finally, in Section 5, we explain how Web is extended to implement the reliability management.

A. Web presentation

Web is a data warehouse opened on the Web [4, 5] centered (in its current version)on the integration of heterogeneous data tables extracted from Web documents. The focus has been put on Web tables for two reasons: (i) experimental data are often summarized in tables, (ii) table structured data are easier to integrate than, e.g., in text or plots. The main steps of Web table integration are summarised in Fig. 1. A central role in data integration in @Web is played by the domain ontology. This ontology describes the concepts, their relations and the associated terminology of a given application domain. @Web can therefore be instantiated for any application domains (e.g., food predictive microbiology, food chemical risks, aeronautics [5]), provided a proper domain ontology is defined. Once the ontology is built, @Web workflow includes the different steps shown in Fig. 1 to integrate new data in the warehouse. Concepts found in a data table and semantic relations linking these concepts are automatically identified. Data tables are then annotated with the identified concepts, allowing users to interrogate and query the data warehouse in an homogeneous way.

B. Web generic ontology

The current OWL ontology representation used in the Web system is composed of two main parts: a generic part, called core ontology, which contains the structuring concepts of the Web table integration task, and a specific part, commonly called domain ontology, which contains the concepts specific to the considered domain. The core ontology is composed of symbolic concepts, numeric concepts and relations between these concepts. It is separated from the definition of the concepts and relations specific to a given domain, i.e., the domain ontology. All the ontology concepts are materialized by OWL classes. For example, in the microbiological ontology, the respectively symbolic and numeric concepts Microorganism and pH are represented by OWL classes, respectively subclass of the generic classes Symbolic Concept and Numeric Concept.

II.RELATED WORK

Estimating data reliability is a major issue for many scientists, as these data are used in further inferences. During collection, data reliability is mostly ensured by measurement device calibration, by adapted experimental design and by statistical repetition. However, full traceability is no longer ensured when data are reused at a later time by other scientists. If a validated physical model exists and data values fall within the range of the model validated domain, then data reliability can be assessed by comparing data to the model predictions. However, such models are not always available and data reliability must then be estimated by other means. This estimation is especially important in areas where data are scarce and difficult to obtain (e.g., for economical or technical reasons), as it is the case, for example, in Life Sciences. The growth of the web and the emergence of dedicated data warehouses offer great opportunities to collect additional data, be it to build models or to make decisions. The reliability of these data depends on many different aspects and meta-information: data source, experimental protocol, developing generic tools to evaluate this reliability represents a true challenge for the proper use of distributed data.

In classical statistical procedures, a preprocessing step is generally done to remove outliers. In procedures using web facilities and data warehouses, this step is often omitted. The method presented here answers these needs, by addressing two issues: first we propose a generic approach to evaluate global reliability from a set of criteria, second we consider the problem of ordering the reliability assessments so that they are presented in a useful manner to the end-users. Indeed, the goal of the present work is to propose a partly automatic decision-support system to help in a data selection process. As evaluating data reliability is subject to some uncertainties, we propose to model information by the means of evidence theory, for its capacity to model uncertainty and for its richness in fusion operators. Each criterion value is related with a reliability assessment by the means of fuzzy sets latter transformed in basic belief assignments, for the use of fuzzy sets facilitates expert elicitation. Fusion is achieved by a compromise rule that both copes with conflicting information and provides insights about conflict origins. Finally, interval valued evaluations based on lower and upper expectation notions are used to numerically summaries the results, for their capacity to reflect the imprecision (through interval width) in the final knowledge. As an application area, we focus on Life Science and on reliability evaluation of experimental data issued from arrays in electronic documents. Section II explains what we understand by reliability and discusses related notions and works. Section III is dedicated. to an analysis of the information available to infer data reliability (with a focus on experimental data). Section IV describes the method used to model this information and to merge the different criteria using evidence theory. Section V addresses the question of data ordering by groups of decreasing reliability and subsequently the presentation of informative results to end-users. Section VI is devoted to the practical implementation of the approach to the case of the @Web data warehouse [2], [3]. It also presents a use case in the field of predictive microbiology.

III.METHOD OF PROCESS

Generic combined with clustering s are efficient search methods based on principles of natural selection and population genetics. They use randomized operators operating on a population of candidate solutions to generate a new population of candidates in the search space (Goldberg [8]). For any GA, a chromosome representation is needed to describe each individual in the population of interest. Each individual or chromosome is made up of a sequence of genes from a certain alphabet. Though the alphabet was limited to binary digits in Holland's original design [13], other very useful problem-specific representations of an individual or Chromosome for function optimization has also been proposed. GAs can search the solution space for optimal solutions very efficiently by using evaluation and genetic operator functions to maintain the useful schema in the population. For example, in a chromosome with a binary string representation of length eight, the string 101#####, where the # represents a "wild card", either 0 or 1, is a schema. Other types of schema are possible also. Individuals exhibiting a schema which results in higher fitness will have a higher probability of survival by the selection process in each generation and thereby will have a higher probability of being selected for mating and generating offspring which are likely to exhibit the same schema. (Mating is accomplished by the crossover operator function, in which the pair of mating chromosomes exchanges substrings to produce a pair of offspring.) The new offspring usually include improved solutions since they tend to inherent the good schema, i.e., the good schema persist in the population over multiple generations. This has been discussed in detail by Michalewicz [18]. This section provides only a brief introduction of GAs; interested readers are referred to the excellent book by Goldberg [8].

A. GA Representation for Integer Reliability Problems

For a chromosome we use a vector of m integer numbers to represent the redundancy of the m subsystems. For instance, the chromosome 23341 represents a system with five subsystems, the first of which contains two elements, the second subsystem three elements, etc. The most interesting aspect of evolution (which includes reproduction, crossover and mutation etc.) is that of natural selection, which can be accomplished by the following steps:

Step 1. Randomly generate a population of chromosomes.
Step 2. Evaluate the fitness function for each individual in the population.
Step 3. If the stopping criteria have been achieved, then stop; else, go to the Step 4.
Step 4. Perform reproduction, crossover and mutation within the population.
Step 5. Form the new generation from the individuals resulting from Step 4. Go to Step 2.

A specified number of individuals in the initial population are randomly generated, after deciding upon an upper limit ci on the number of elements in each subsystem i (Such an upper limit can always be computed based upon the constraints, if no priori estimate is available.) Typically, GAs evaluate individuals in the population by a so-called "fitness function" which is a composite of both the objective value (in this case, system reliability) and the penalty arising from the violation of constraints. In this paper, the fitness is defined as follows.

```
If one or more of the three constraints have been violated
Then Fitness=0
Else
Fitness=Objective value
End
```

The genetic operators include crossover and mutation. In Figure 4, the crossover point was randomly selected and the offspring generated by swapping the partial strings of parent 1 and parent 2. In the mutation operation, each individual's chromosome is mutated with a specified small, but positive, probability. A gene within the string is randomly selected to be mutated, and the selected gene's new value randomly selected within a range from 1 to the upper limit for redundancy of elements in the associated subsystem. For example, in Figure 4 two parents (23241 and 31422) are selected, and the third gene of the strings was selected for crossover, yielding the strings 32422 and 31241. With a specified probability, the resulting children are selected for mutation. In this case, the third gene of child 1 was selected to be mutated, and the new value, 1, was randomly generated for this gene, yielding the string 23122. That is, the number of the elements in the third subsystem has been decreased from 4 to 1 by the mutation operation.

IV.EXPERIMENT RESULTS

We conducted experiments to evaluate the presented concepts for a clustering based generic of our query execution approach. For our tests we adopt the Berlin SPARQL Benchmark (BSBM) [10]. The BSBM executes a mix of 12 SPARQL queries over generated sets of RDF data; the datasets are scalable to different sizes based on a scaling factor. Using these datasets we set up a Web Enabled dataserver6 which publishes the generated data following the Web Enabled data principles. With this server we simulate the Web of in our experiments. To measure the impact of URI perfecting and of our iterate paradigm extension we use the SWClLib to execute the BSBM query mix over the simulated Web. For this evaluation we adjust the SPARQL queries provided by BSBM in order to access our simulation server. We conduct our experiments on an Intel Core 2 Duo T7200 processor with 2 GHz, 4

MB L2 cache, and 2 GB main memory. Our test system runs a recent 32 bit version of Gentoo Linux with Sun Java 1.6.0. We execute the query mix for datasets generated with scaling factors of 10 to 60; these datasets have sizes of 4,999 to 26,108 triples, respectively. For each dataset we run the query mix 6 times where the first run is for warm up and is not considered for the measures. Figure 6(a) depicts the average times to execute the query mix with three different implementations of SWClLib: i) without URI perfecting, ii) with perfecting, and iii) with the extended integrators that postpone the processing of input solutions. As can be seen from the measures URI perfecting reduces the query execution times to about 85%; our non-blocking integrators even halve the time.

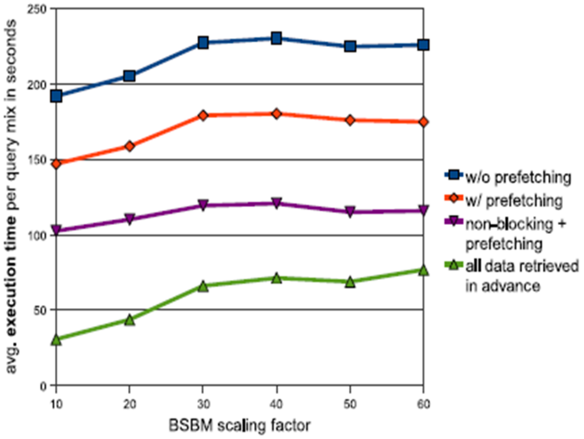

Fig.1

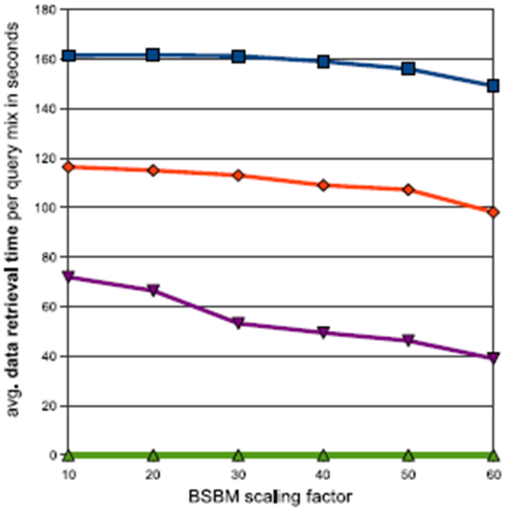

Fig.2

The chart in Figure puts the measures in relation to the time it takes to execute the query mixes without the need to retrieve data from the Web. We obtained this optimum by executing each query twice over a shared dataset; we measured the second executions which did not require to look up URIs because all data has already been retrieved in the first pass. These measures represent only the time to actually evaluate the queries as presented in Section 3. Hence, these times are a lower bound for possible optimizations to the integrator-based execution of our approach to query the Web of Linked Data. Using this lower bound we calculate the times required for data retrieval in the three implementations. These times are the differences between execution times measured for the three implementations and the lower bound, respectively. Figure depicts these numbers which illustrate the significant impact of the possibility to postpone the processing of certain input solutions in our non-blocking iterators. The chart additionally illustrates that the data retrieval times compared to the whole query execution time decreases for larger datasets, in particular in the case of the non-blocking iterates.

V.CONCLUSION

This paper considers three typical types of reliability problems, which include the series system, the series-parallel system, and the complex (bridge) system. The objective of these problems is to maximize the system reliability subject to various nonlinear constraints. Unlike most well-known heuristic methods, GAs are able to solve both integer reliability problems and mixed-integer reliability problems. Furthermore, their applicability is not limited to series-parallel systems. As shown in the previous section, the optimal solutions (except for problem P2a) by GAs are all superior to or tie the best solutions by other well-known heuristic methods for both integer reliability problems (in which component reliabilities are given and redundancy allocation is to be decided) and mixedinteger reliability problems (in which both the component reliabilities and redundancy allocation are to be decided simultaneously). In agreement with the success of numerous applications of GAs in various other classes of problems, our limited experience with these reliability problems has shown that GAs are very competitive with other heuristic methods. They are especially appropriate for design of nonstandard series-parallel systems. In addition, as reported in this paper, the multiple solutions found by the GA sometimes vary significantly in the component reliabilities and/or redundancy allocation for systems. This 15 offers the design engineer a variety of options from which to choose with negligible differences in the system reliability

REFERENCES

1. Berners-Lee, T.: Design Issues: Linked Data. Online, Retrieved May 25, 2009,

2. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. Journal on Semantic Web and Information Systems (in press) (2009)

3. Franklin, M.J., Halevy, A.Y., Maier, D.: From databases to dataspaces: A new abstraction for information management. SIGMOD Record 34(4) (December 2005) 27–33

4. Prud'hommeaux, E., Seaborne, A.: SPARQL query language for RDF. W3C Recommendation (January 2008) Retrieved June 11, 2009, from

http://www.w3.org/TR/rdf-sparql-query/.

5. Sheth, A.P., Larson, J.A.: Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys 22(3) (September 1990) 183–236

6. Garcia-Molina, H., Widom, J., Ullman, J.D.: Database Systems: The Complete Book. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (2002)

7. Graefe, G.: Query evaluation techniques for large databases. ACM Computing Surveys 25(2) (June 1993) 73–169

8. Pirahesh, H., Mohan, C., Cheng, J., Liu, T.S., Selinger, P.: Parallelism in relational data base systems: Architectural issues and design approaches. In: Proceedings of the 2nd International Symposium on Databases in Parallel and Distributed Systems (DPDS), New York, NY, USA, ACM (1990) 4–29

9. Hartig, O., M¨uhleisen, H., Freytag, J.C.: Linked data for building a map of researchers. In: Proceedings of 5th Workshop on Scripting and Development for the Semantic Web (SFSW) at ESWC. (June 2009)

10. Bizer, C., Schultz, A.: Benchmarking the performance of storage systems that expose SPARQL endpoints. In: Proceedings of the Workshop on Scalable Semantic Web Knowledge Base Systems at ISWC. (October 2008)

11. Quilitz, B., Leser, U.: Querying distributed RDF data sources with SPARQL. In: Proceedings of the 5th European Semantic Web Conference (ESWC). Volume 5021 of Lecture Notes in Computer Science., Springer Verlag (June 2008) 524–538

12. Langegger, A., W¨oß, W., Bl¨ochl, M.: A semantic web middleware for virtual data integration on the web. In: Proceedings of the 5th European Semantic Web Conference (ESWC). Volume 5021 of Lecture Notes in Computer Science., Springer Verlag (June 2008) 493–507

13. Oren, E., Delbru, R., Catasta, M., Cyganiak, R., Stenzhorn, H., Tummarello, G.: Sindice.com: A document-oriented lookup index for open linked data. International Journal of Metadata, Semantics and Ontologies 3(1) (2008)

14. Ding, L., Finin, T.W., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V., Sachs, J.: Swoogle: A search and metadata engine for the semantic web. In: Proceedings of the 13th ACM Conference on Information and Knowledge Management (CIKM), ACM (November 2004) 652–659

15. d'Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., Lopez, V., Guidi, D.: Toward a new generation of semantic web applications. IEEE Intelligent Systems 23(3) (2008) 20–28

16. Berners-Lee, T., Chen, Y., Chilton, L., Connolly, D., Dhanaraj, R., Hollenbach, J., Lerer, A., Sheets, D.: Tabulator: Exploring and analyzing linked data on the semantic web. In: Proceedings of the 3rd SemanticWeb User InteractionWorkshop(SWUI) at ISWC. (November 2006)
[17] S. Ramchurn, D. Huynh, and N. Jennings, “Trust in multi-agent systems,” The Knowledge Engineering Review, vol. 19, pp. 1–25, 2004.
[18] P. Buche, J. Dibie-Barth´elemy, and H. Chebil, “Flexible sparql querying of web data tables driven by an ontology,” in FQAS, ser. Lecture Notes in Computer Science, vol. 5822, 2009, pp. 345–357.
[19] G. Hignette, P. Buche, J. Dibie-Barth´elemy, and O. Haemmerl´e, “Fuzzy annotation of web data tables driven by a domain ontology,” in ESWC, ser. Lecture Notes in Computer Science, vol. 5554, 2009, pp. 638–653.
[20] D. Mercier, B. Quost, and T. Denoeux, “Refined modeling of sensor reliability in the bellief function framework using contextual discounting,” Information Fusion, vol. 9, pp. 246–258, 2008.
[21] R. Cooke, Experts in uncertainty. Oxford, UK: Oxford University Press, 1991.
[22] S. Sandri, D. Dubois, and H. Kalfsbeek, “Elicitation, assessment and pooling of expert judgments using possibility theory,” IEEE Trans. on Fuzzy Systems, vol. 3, no. 3, pp. 313–335, August 1995.
[23] F. Delmotte and P. Borne, “Modeling of reliability with possibility theory,” IEEE Trans. on Syst., Man, and Cybern. A, vol. 28, no. 1, pp. 78–88, 1998.
[24] F. Pichon, D. Dubois, and T. Denoeux, “Relevance and truthfulness in information correction and fusion.” in International Journal of Approximate Reasoning, vol. Accepted for Publication, 2011.
[25] J. Sabater and S. Sierra, “Review on computational trust and reputation models,” Artificial Intelligence Review, vol. 24, no. 33-60, 2005.
[26] J. Golbeck and J. Hendler, “Inferring reputation on the semantic web,” in Proceedings of the 13th international World Wide Web conference, New York, NY, USA, 2004.