International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 8, No.4 (July-2019)

http://dx.doi.org/10.12785/ijcds/080410

SCATTER: Fully Automated Classification System across Multiple Databases

Tahar Mehenni[1]

[1]*Computer Science Department, Mohamed Boudiaf University, M'sila, Algeria*

Received 13 Mar. 2019, Revised 21 Apr. 2019, Accepted 30 May 2019, Published 1 July 2019

Abstract: Data mining approaches performed recently use data coming from a single table and are not adapted to multiple tables. Moreover, computer network expansion and data sources diversity require new data mining systems handling databases heterogeneity in multi-database systems. In this paper, we propose SCATTER: a fully automated classification system from multiple heterogeneous databases. SCATTER is composed of three components. The first component uses schema matching techniques to find foreign-key links across the multi-database system. The second component tries to find the most useful links that are critical for producing accurate classes across multiple databases. The last component is a decision tree classification algorithm which exploits the useful links discovered automatically across the databases. Experiments performed on real databases were very satisfactory with an average accuracy of 86.5% and showed that SCATTER system succeeded in achieving a fully automated classification from multiple heterogeneous databases.

Keywords: Classification, Heterogeneous databases, Link discovery, Link usefulness, Multi-database mining

1. Introduction

Relational databases are used nowadays extensively, and thus constitute one of the richest sources of information in the world. Moreover, the diversity of data sources resulted from the expansion of computer network caused a great need to discover knowledge from multiple databases obtained from multiple sources and stored in different sites. Fig. 1 illustrates many inter-linked databases with a main database making a "multi-database system".

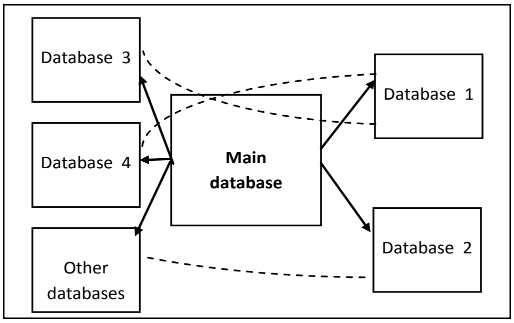


Figure 1. Architecture of the multi-database system (the arrows are links between the main database and the other databases, and the dotted lines are links between the other databases that do not refer to the main database).

Examples arise naturally in the world, where searching data that is scattered across multiple databases is needed. For instance, in order to develop a new product, companies will join their forces and retrieve information about their competitors. Thus, they need to combine their databases to perform a data mining task.

Traditional data mining approaches from multiple databases integrate all the databases, and then apply the selected algorithm to build the model [1, 2]. However, the huge dataset resulted after the integration and the data heterogeneity of the multiple databases will be difficult to process. Therefore a different approach is needed. The main idea of this approach is to build joins over the different databases using some useful links. Unfortunately, two major challenges are present for this approach:

- Solving the data heterogeneity problem: databases are joinable by some attributes which are called keys. However, these keys are heterogeneous. Thus, we need techniques to solve this heterogeneity in order to define approximate foreign-key links across databases.
- Finding useful links: we can discover several foreign-key links that connect multiple relations. However, some links are interesting as useful

E-mail:tahar.mehenni@univ-msila.dz

bridges, but others may link unrelated objects. Moreover, finding all the links and trying to establish connections between all the databases is very expensive in time. Thus, we have to find techniques to estimate the link usefulness between all the databases in order to perform an efficient communication strategy across the multiple databases.

In this paper, we try to take in consideration the above challenges and propose a fully automatic classification system over multiple heterogeneous databases. The proposed system is called SCATTER which means to disperse and go in various directions. To solve the heterogeneity problem, we partially follow the work of [3], where the author presented a framework that automatically identifies approximate foreign-key joins in the multiple heterogeneous databases. Moreover, our system performs better in finding the most useful joins across the data sources, thanks to the regression model used in predicting the link usefulness [4,5,6]. To perform the classification task, we use the decision tree classification algorithm that exploits the joins discovered automatically across the databases [4,5,6]. Experiments performed on five real databases were very satisfactory and show that the proposed system succeeded in achieving a fully automatic classification across multiple heterogeneous databases.

The remaining of the paper is organized as follows. Section 2 presents some basic concepts and discusses the related work. An overview of the main components of the proposed system is presented in section 3. Sections 4, 5 and 6 give more details to SCATTER's components. Experiments and discussion of the results are given in Section 7 and finally, the study is concluded in section 8.

2. RELATED WORK

A multi-database system is a set of multiple heterogeneous databases located in different sites. A multi-database is a distributed system that acts as a front end to multiple local Database Management System (DBMS) or is structured as a global system layer on top of local DBMSs [7].

Multi-database mining is the process of analyzing the data in multi-databases, and finding useful and novel knowledge, which is highly supported by all or the most of databases [8]. While databases may be heterogeneous, many methods exist for discovering knowledge from multiple data sources. These methods fall into two wide categories, namely single database mining and multi-database mining [9].

In single database mining, data from different data sources has been aggregated to a centralized repository for the task of mining. Single database mining could not be considered a good solution for mining multiple databases because of the following limitations [10]:

1. It is based on the traditional data warehouse architecture.

2. It is very expensive in time and budget to process the entire data set obtained from the whole databases.

3. Even if the data can be quickly centralized using relatively fast network, the privacy issue with this method is not satisfied.

5. Putting all the data from the relevant databases into a single data set can destroy some important information that reflects the individuality of the different databases.

6. And the important limitation is the heterogeneity problem, where the aggregation of all the heterogeneous databases to obtain a whole single database could be simply an unfeasible solution.

The above limitations show that the traditional process of single database mining is inadequate and an alternate way for mining multiple data sources must occur.

The objective of multi-database mining is to perform the data mining task based on the type and availability of the distributed data sources without moving to the central repository. It mines important local patterns from individual data sources, forwards the pattern base and reduces the data movement [11].

Multi-database mining aims to discover global patterns in multi-database systems. Global patterns are well discussed in [12,13,14,15,16] where authors introduced different strategies. Reference [17] presented an algorithm for selecting the most relevant databases to a multi-database mining task, however, to avoid the database-dependency, several works presented a database-independent database classification [18,19,20].

Little work is published on classification task across multiple and heterogeneous databases. In MDBM [5], authors proposed a neural network regression based method, for predicting the link gainfulness, and suggested a rule-based classification algorithm. DTHR [4] is a decision tree based classification approach where Support Vector Regression model is proposed for identifying the most useful links to build a multi-relational decision tree over the whole relational databases. Unfortunately, authors of [4] and [5] assume that the problem of heterogeneity is solved and present more efficient classifiers across the multiple heterogeneous databases.

In [3] the author proposed HeteroClass, a framework where the author tries to find approximate foreign key joins using schema matching and structure discovery to solve the heterogeneity problem. Then, an ensemble classifiers algorithm is proposed for the classification task. However, HeteroClass still using the traditional way that is integrating all the databases after solving the heterogeneity problem.

In [21], authors present a review of recent progresses in the mining field from multiple data sources. In order to remove conflicts in the data sources; authors focus on how to manage data sources before performing data mining tasks. They present four techniques for this purpose: local pattern analysis, classification, clustering and fusion of data sources. These approaches still need an efficient communication strategy between all the data sources when performing data mining tasks.

In order to give a solution to the problems mentioned above, we propose SCATTER a fully automated system that performs the classification task from multiple heterogeneous databases. SCATTER will perform a better solution than the previous works, thanks to three integrated strategies. The first strategy is the recent algorithms used to discover links between the multiple databases which are fast and more efficient. The second one is using a novel technique that predicts the most useful links to the data mining task which is an efficient communication strategy, and finally SCATTER performs the classification data mining task using the decision tree algorithm which is an efficient and more accurate approach well studied for decades. In one word, SCATTER is proposed to be a multi-database mining system achieving the classification task more efficiently and without integrating all the databases.

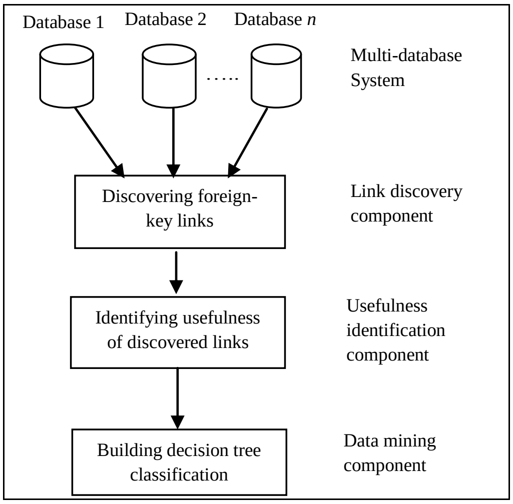
3. Components of Scatter

The main purpose of the proposed system is first, to solve the heterogeneity problem of the multiple data sources, and second, to use an efficient and adaptive data mining approach from the multi-database system.

Our approach is related to the work of [3] but quite different from it in that:

- We first discover approximate foreign key links between databases.
- Then we find the most useful of them using an entropy-based technique.
- We do not integrate all the databases, as in [3]. We proceed by building bridges between them over some selected relations which bring the most useful links. This strategy will reduce communication costs between databases.
- We use an efficient classification approach, which is the decision tree, unlike in [3] where ensemble classifiers algorithm is used.

Fig. 2 gives an overview of the proposed system. It is composed of three components: link discovery, usefulness identification and data mining. The link discovery component consists of discovering foreign-key links in the whole databases. The second component identifies the usefulness of the discovered links. While the third and last component is a decision tree classification approach to perform the data mining task.



Database 1
Database 2
Database *n*
…..
Multi-database System
Discovering foreign-key links
Link discovery component
Identifying usefulness of discovered links
Usefulness identification component
Building decision tree classification
Data mining component

Figure 2. The main components of SCATTER

4. Link Discovery

Link discovery consists in finding all foreign-key links in table *A* that reference a key in table *B*. This task is performed following four steps.

- Step 1: Find all approximate keys in *A* that will participate in any foreign-key links to discover.
- Step 2: For each approximate key of *A*, find any linkable attribute in *B*.
- Step 3: Generate candidate foreign-key links.
- Step 4: keep "semantically correct" foreign-key links.

Finding all approximate keys in table *A* is performed using an approximate key discovery algorithm. Many works are presented in this research field [22], but recently, a novel and highly efficient algorithm, called Pyro, is proposed for discovering approximate dependencies [23]. Pyro detects dependency of attributes in a table using a separate-and-conquer search strategy with sampling-based guidance. We implemented Pyro and used it to find all the approximate keys in a table of the database.

For each approximate key found in table *A*, we try in Step2 to find any linkable attribute in *B*. For this, we find all attributes *b* in *B* that are joinable with attributes *a* of *A*, i.e. they have similar values. Computing the similarity of two attributes is based on set resemblance. Let V_a and V_b the sets of values of the attributes *a* and *b*, the resemblance of the sets V_a and V_b is defined as the ratio of their intersection and their union [3], i.e.

$$r(V_a, V_b) = \frac{|V_a \cap V_b|}{|V_a \cup V_b|} \qquad (1)$$

In order to compute similarity of values, exact matching is used for the numerical values and q-grams for the textual ones.

Step3 establishes an exhaustive list of all the candidate foreign-keys identified in step 2. An attribute *a* in table *A* is a candidate foreign-key *b* in table *B*, if *a* and *b* are joinable and similar.

To have an adequate similarity value in order to keep the most similar attributes, many experiments were performed and compared to define a threshold and use it in the system. The process is to pick a similarity value and compute the number of similar attributes using this value, and then we compare it with the real similar attributes found apriori and discussed with some specialists.

We believe that keeping an adequate value that can determine a reasonable set of similar attributes is not an easy task and may affect the quality of the obtained results. We choose the most adequate value after many tries and discussions and keep it as a threshold.

The final step performed in the link discovery component is keeping semantically correct foreign-keys links. For this purpose, a full schema matching technique is used. Simflood is a simple algorithm proposed in [24], it computes a score for any two attributes using the similarity measure of their names and their neighborhood.

5. USEFULNESS IDENTIFICATION

Once we have identified the meaningful foreign-key joins, it will be interesting to find out their usefulness before joining their relations for the classification task.

For this purpose, a regression model is built to predict the useful links between two relations from different databases. Following the work of [4], the usefulness of a link *l* is defined as the maximum gain ratio obtained from an attribute A_l through the link *l*, as follows.

$$\text{usefulness}(l) =$$

$$\max_{A_l \in R_l} \frac{\text{entropy}(P, N) - \sum_{i=1}^{k} \frac{P_i + N_i}{P + N} . \text{entropy}(P_i, N_i)}{-\sum_{i=1}^{k} \frac{P_i + N_i}{P + N} . \log \frac{P_i + N_i}{P + N}} \quad (2)$$

Where

$$\text{entropy}(P, N) =$$

$$-\left(\frac{P}{P+N} \log \frac{P}{P+N} + \frac{N}{P+N} \log \frac{N}{P+N}\right) \qquad (3)$$

P denotes the positive tuples and *N* the negative ones in a table R_l. A_l is the attribute which divides the tuples into *k* partitions; each contains P_i positive tuples and N_i negative ones.

To predict the usefulness of a link, a regression model is used. We choose the support vector regression for its efficiency [25]. Link properties used as parameters for the regression model are multiple. We compute some statistics such as coverage and deployment [4].

6. DATA MINING COMPONENT

The main component of SCATTER is data mining, which is performed using the well known decision tree algorithm for the classification task.

Decision tree algorithm consists of adding decision nodes to the tree using a set of successive refinements to find a good split. The process will end when a stopping condition is met, i.e. a leaf node with class label is identified instead. The refinement begins by computing the information gain for all possible attributes of the active relation and the inactive relations joinable with the active relation having the highest usefulness, and then it selects the best attribute with the highest information gain.

The overview of the data mining component of SCATTER is represented in Fig. 3 where the pseudo code of the decision tree algorithm is given. The procedure *DecisionTreeScatter* begins by testing if a stopping condition is reached, i.e. usually a running time defined previously or a perfect set of examples having nearly the same class label. In this case a leaf node is created and labeled. For the other cases, the procedure achieves three tasks. First it finds the attribute having the highest info gain in the main database, and then it builds the set of links over the other databases and the main database having a usefulness value greater than a defined threshold in order to have a reasonable set of the most useful links across the databases. The threshold usefulness is a constant defined previously after a set of experiments and discussions with many experts and researchers. The third task is finding A_{max} the attribute with the highest info gain among all the found attributes. Finally, the procedure *TreeGrowth* will achieve the remaining task related to building the tree, such as splitting and branching.

```
Input : Heterogeneous Databases, $R_t$ target relation.
Output : Tree decision of label classes.
Procedure DecisionTreeScatter
If Stopping_cond()
Then
  leaf_node:= Create_node();
  leaf_node.label:= Classify();
  Return leaf_node;
Else
  For each active relation $R_a$
     $A_{max}$ := info-gain($R_a$, $R_t$);
  For each inactive relation $R_i$;
     UsefulLinks:= Set of joinable links having  usefulness
     greater than a defined threshold;
     $A_{max}$ := info-gain($R_a$, $R_i$);
  $T$:= Tree_growth($T$, $A_{max}$);
  Return node($T$, $A_{max}$);
End
```

Figure 3. Decision tree Algorithm pseudo code

7. Experiments and Discussion

Comprehensive experiments on real datasets are performed to show accuracy and efficiency of the proposed system. The experiments are run on i5 PC, with 4GB RAM running Windows XP. The language used is C# under Visual Studio.Net 2018.

To evaluate efficiency and accuracy of the proposed system, we performed experiments on five real datasets DbLife, Inventory, DbCsd, LoanBank and MoviePeople. Table 1 shows these datasets where some details are presented for each database such as the number of relations, the number of attributes, the number of keys and the number of records of the main relation.

A. Experiments on Link Discovery Techniques

We first examine the foreign-key links identified by the proposed system. Initially, all foreign-keys are identified and labeled manually as correct or incorrect, and then for each dataset, we choose one database (called the main database) and apply link discovery procedure for the remaining databases in order to find automatically all the foreign-keys. Finally, we compute the overall accuracy and recall as follows.

$$Precision = \frac{Identified\ Correct\ foreign\ keys}{All\ foreign\ keys} \quad (4)$$

$$Recall = \frac{Correct\ identified\ foreign\ keys}{All\ foreign\ keys} \quad (5)$$

The results in Fig. 4 show that discovery links techniques achieve high accuracy (76 to 89%) and important recall (68 to 88%).

TABLE I. Datasets used in the experiments

Dataset	Databases with number of relations, number of attributes, number of keys, and number of records of the main relation.	Classification task
DbLife	Publication (7,36,15,1885) Support (4,22,12,975) Researchers (4,17,8,825) Co-authorship (4,13,7,772) Department (5,11,6,54) DBWorldEvents (5,9,4,382)	Predict topics of the researchers
Inventory	Products (7,79,34,1372) Stores (3,9,5,245) Availability (5,19,11,462) Associated inventory (3,8,6,533) Sites (2,9,4,76)	Predict the availability of products
DbCsd	Students (14,52,23,2500) Researchers (4,19,7,610) Publication (7,35,14,1500)	Predict research fields of students
LoanBank	Loan (8,43,12, 842) Bank (4,19,10,6520)	Predict loan results
MoviePeople	Movie (4,33,12,5625) People (5,13,8,3200)	Determine whether a director is old or new (career before or after 1970)

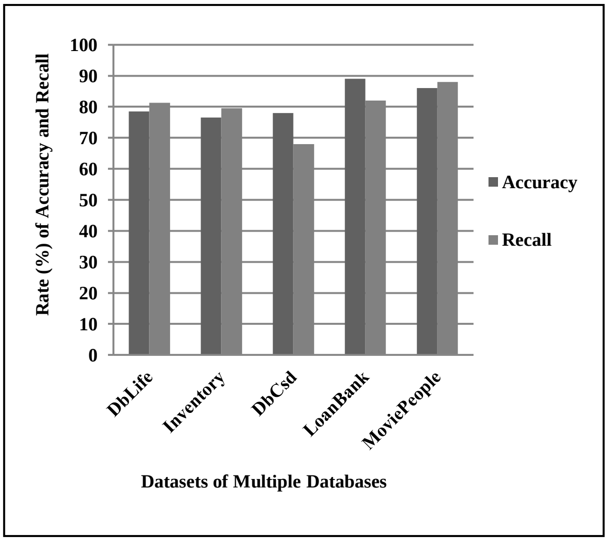


Figure 4. Join discovery accuracy of the system SCATTER

B. Experiments on Predicting Usefulness of Links

Experiments on both datasets were conducted using three-fold cross validation, where the training set is divided into three blocks, each block is hold out once as a testing set, while the remaining blocks are used as a training set in the regression model.

Three-fold cross validation strategy is chosen after many experiments taking various folds of both datasets. The procedure is executed three times on different training sets. In this experiment, a link is considered useful as its usefulness is greater than 0.52. This threshold is fixed in order to keep a reasonable number of links between the databases. In order to have a good threshold, we execute the program and compute the number of useful links obtained with the accuracy rate. After many executions with different values of usefulness, we keep the best value having a number of useful links with a high accuracy. Finally, this value is used as a threshold of link usefulness. For our case, and after about 15 executions of SCATTER, we obtain the results shown in Table 2.

TABLE II. FINDING THE THRESHOLD OF LINK USEFULNESS USING A REPEATED EXECUTION OF SCATTER WITH DIFFERENT VALUES.

Execution of SCATTER	Threshold of link usefulness	Average Number of obtained useful links	Average Accuracy (%)
1	0.35	178	62
2	0.40	170	65
3	0.41	152	65
4	0.43	135	66
5	0.45	120	70
6	0.49	117	75
7	0.50	116	76
8	**0.52**	**110**	**76**
9	0.54	108	76
10	0.55	108	75
11	0.58	102	70
12	0.60	100	62
13	0.65	85	58
14	0.70	60	46
15	0.75	60	45

We compute the precision, accuracy and recall of prediction. The results are presented in Fig. 5. One can see that the proposed regression model achieves high accuracy and good precision for predicting usefulness of links.

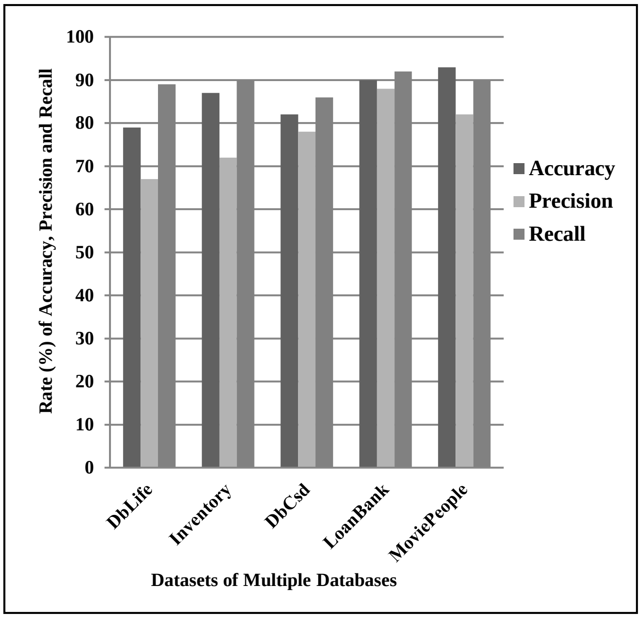


Figure 5. Predicting usefulness of links using Support Vector Regression in the system SCATTER.

C. *Experiments on Classification algorithm*

SCATTER is compared with two previous works: MDBM and HeteroClass. MDBM is a rule-based classification algorithm used in multi-relational and heterogeneous databases [4], while HeteroClass is an effective classification approach from heterogeneous databases using ensemble classifiers [3]. We compare the accuracy and running time (in seconds) for the five datasets used in the experiments. We notice that in MDBM the foreign-key links are assumed to be known, so we modify it to use our discovery link component.

Comparing results are presented in Fig. 6. It can be seen that the three algorithms have nearly the same accuracy. However, HeteroClass and SCATTER achieve both a better accuracy than MDBM. SCATTER is more accurate than the other algorithms because it combines techniques of both MDBM and HeteroClass.

In Table 3 we compare training and testing times of three algorithms. One can see that HeteroClass takes more time to perform classification than the other algorithms. However, MDBM using rule-based classification is faster, but SCATTER is faster than HeteroClass thanks to Pyro the approximate link discovery algorithm, and to the usefulness identification component. It can be seen overall, that the three algorithms are fast; the slowest in these datasets, take two minutes in training and 2.6 seconds in testing.

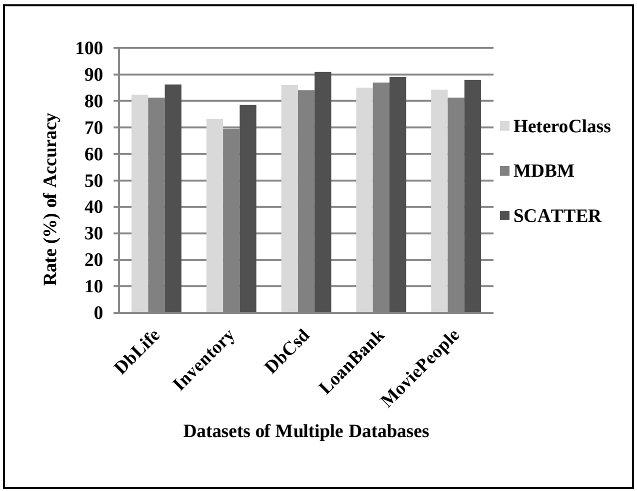


Figure 6. Comparing classification accuracy of SCATTER and two previous algorithms in five datasets of multiple databases

TABLE III. COMPARING RUNNING TIME (IN SECONDS) OF THREE ALGORITHMS IN FIVE DATASETS.

	DbLife	**Inventory**	**DbCsd**	**LoanBank**	**Movie People**
Training times					
HeteroClass	52	78	64	120	45.2
MDBM	40	65	40	108	39
SCATTER	49	76	58	105	40
Testing times					
HeteroClass	1.4	2.1	2.3	2.6	1.5
MDBM	0.9	1.4	1.2	1.7	0.6
SCATTER	1	1.4	1.8	1.7	1.3

8. CONCLUSION

Multiple heterogeneous databases are widely used in many disciplines, such as decision support and medical research. In this paper, we have studied the problem of classification from multiple heterogeneous relational databases.

We developed SCATTER, a system that addresses the data heterogeneity problem using schema matching and structure discovery techniques. A decision tree classification approach is performed using a regression based model to predict usefulness of links. We evaluated our proposed system using five real-world datasets and show the improvement over two recent applicable algorithms in this context. The average accuracy of our proposed system was 86.5%. The results were satisfactory and the proposed system tends to be a fully automated classification over multiple heterogeneous databases.

SCATTER may be used in a distributed environment, and it will be interesting to test it in order to perform classification across a set of distributed databases. Although, SCATTER gives techniques to automatically resolve the heterogeneity problem, it is interesting to test other link discovery approaches in order to have more accuracy and recall.

REFERENCES

[1] J.A.R. Castillo, A. Silvescu, D. Caragea, J. Pathak, and V.G. Honavar. “Information extraction and integration from heterogeneous, distributed, autonomous information sources -a federated ontology-driven query-centric approach”. IEEE International Conference on Information Reuse and Integration, pp.183-191, 2003

[2] E. Rahm and P.A. Bernstein. “A survey of approaches to automatic schema matching”. VLDB, vol. 10(04), pp.334-350, 2001.

[3] M. Sayyadian. “HeteroClass: a framework for effective classification from heterogeneous databases”. Project Report CS512, University of Wisconsin, Madison, 2006.

[4] X. Yin and J. Han. “Efficient classification from multiple heterogeneous databases”. Knowledge Discovery. Proceedings of PKDD 2005, vol. 3721, pp.404-416, 2005.

[5] T. Mehenni and A. Moussaoui. “Data mining from multiple heterogeneous relational databases using decision tree classification”. Pattern Recgnition Letters, vol. 33, pp. 1768-1775, 2012.

[6] T. Mehenni. “Integration of useful links in distributed databases using decision tree classification”. In 6th International Conference on Information Systems and Economic Intelligence (SIIE), IEEE, pp. 5-9, 2015.

[7] T. Ozsu, and P. Valduriez. “Principles of Distributed Database Systems”. Springer. Third edition, 2011.

[8] T. Ramkumar, S. Hariharan, S. Selvamuthukumaran. “A survey on mining multiple data sources”. WIREs Data Mining Knowledge Discovery , pp. 1–11, 2012.

[9] A. Adhikari, J. Adhikari and W. Pedrycz. “Data analysis and pattern recognition in multiple databases”. Springer International Publishing, Vol. 61, 2014.

[10] T. Mehenni. “Data mining from multiple heterogeneous Relational databases”. PhD Thesis, University of Bejaia. Unpublished, 2014.

[11] A. Adhikari, P. Ramachandrarao, W. Pedrycz. “Developing Multi-database Mining Applications”. Springer, London, 2010.

[12] A. Adhikari and J. Adhikari. “Mining Patterns in Different Related Databases”. In Advances in Knowledge Discovery in Databases, Springer, Cham, pp. 83-95, 2015

[13] A. Adhikari, J. Adhikari and W. Pedrycz. “Synthesizing Global Patterns in Multiple Large Data Sources”. In Data Analysis and Pattern Recognition in Multiple Databases, Springer, Cham, pp. 61-74, 2014.

[14] A. Adhikari, L.C. Jain and B. Prasad.”A State-of-the-Art Review of Knowledge Discovery in Multiple Databases”. Journal of Intelligent Systems, vol. 26(1), pp. 23-34, 2017.

[15] A. Adhikari and P.R. Rao. “Synthesizing global exceptional patterns in multiple databases”. Proceedings of the 3rd Indian International Conference on Artificial Intelligence, pp.512-531, 2007.

[16] S. Zhang, X. You, Z. Jin, and X. Wu. “Mining globally interesting patterns from multiple databases using kernel estimation”. Expert Systems with Applications, vol. 36(8): pp.10863-10869, 2009.

[17] H. Liu, H. Lu, and J. Yao. “Identifying Relevant databases for multi-database mining”. Research and Development. Knowledge Discovery and Data Mining, vol. 13(94), pp. 210-221, 1998.

[18] R. Sugumar, S. Hariharan and T. Ramkumar. “Classification of Databases for Multi-Database Mining”. IUP Journal of Information Technology, vol. 9(2), 2013.

[19] M. Salim, S.A.R. Hebri and K. Salim. “Multi-database Classification Approaches: A Literature Review”. COSI, pp. 158-169, 2016.

[20] H. Li, X. Hu, and Y. Zhang. "An improved database classification algorithm for multi-database mining". Frontiers in Algorithmics, vol. 55(98), pp. 346-357, 2009.

[21] R. Wang et al., "Review on mining data from multiple data sources", Pattern Recognition Letters, 2018, https://doi.org/10.1016/j.patrec. 2018.01.013

[22] Y. Huhtala, J. Karkkainen, J. Porkka, and H. Toivonen. "TANE: an efficient algorithm for discovering functional and approximate dependencies". The Computer Journal, vol. 42(2), 1999.

[23] S. Kruse and F. Naumann. "Efficient Discovery of Approximate Dependencies", Proceedings of the VLDB Endowment, Vol. 11, No. 7, pp. 759-772, 2018.

[24] S. Melnik, H. Molina-Garcia, and E. Rahm. "Similarity flooding: a versatile graph matching algorithm". ICDE, 2002.

[25] P.A. Cornillon and E. Matzner-Lober. "Regression: theory and applications", "Régression: théorie et applications" (in French) . Springer, Paris, 2007.

Dr. Tahar Mehenni received in 1992 the Engineer degree in Computer Science from University Ain El-Bey of Constantine, Algeria. The Magistere degree in Computer Science from University Mohamed Boudiaf of M'sila, Algeria, in 2006, and the PhD degree in Computer Science from University A. Mira of Bejaia, Algeria, in 2014. He has been working in the area of data mining since 2007. His research work focuses on data-driven decision making and multi-database mining. He also works for a long time in metaheuristics and optimization problems.