Classification Rule Learning for Data Linking

Nathalie Pernelle
LRI (Paris Sud University, CNRS UMR 8623 & INRIA Saclay Ile-de-France)
Bat 650 Universite Paris Sud
91405 Orsay Cedex France
Nathalie.Pernelle@lri.fr

Fatiha Saïs
LRI (Paris Sud University, CNRS UMR 8623 & INRIA Saclay Ile-de-France)
Bat 650 Universite Paris Sud
91405 Orsay Cedex France
Fatiha.Sais@lri.fr

ABSTRACT

Many approaches have been defined to link data items automatically. Nevertheless, when data are numerous and when the schema is unknown, most of these approaches are too time-consuming. We propose an approach where classification rules are learnt thanks to a training set made of linked data. These classification rules can then be applied in order to classify data items and reduce the linking space i.e the space made of data item pairs that have to be compared. First experiments have been conducted on RDF data sets describing electronic products.

Categories and Subject Descriptors

H.2.5 [**DATABASE MANAGEMENT**]: Heterogeneous Databases; I.2 [**Computing Methodologies**]: ARTIFICIAL INTELLIGENCE

General Terms

Algorithms, Experimentation

Keywords

RDF, Linked data, Classification rules

1. INTRODUCTION

Data linking approaches aim to detect and declare links between RDF data items from different data sources. These links allow applications to navigate from a data item to related data items, or to proceed to a data fusion step where one data item is built using all the data items that represent the same real world object.

In order to detect whether two different data items refer to the same entity, one has to compare their descriptions and computes a similarity between them. Numerous approaches have been developed to infer links in databases and artificial intelligence fields [2, 12, 10]. In the Web of data context, some approaches have been developed to infer (semi)-automatically links between data items [7, 11, 8, 9, 4]. However, when data are numerous, this similarity computation can be very time consuming. Some existing approaches exploit schema knowledge like disjunctions between data classes (e.g. electronic products and food products) to reduce the number of data item pairs to be compared [10]. Other approaches use key constraints to split data into smaller partitions [1, 13]. However, without such a priori knowledge, one cannot apply these approaches to decrease the linking space, i.e., the set of data item pairs that have to be compared using a linking method. Furthermore, when the schema of one of the data sources is unknown the set of data item pairs corresponds to the cartesian product of the data items of the two sources to be integrated.

In order to reduce the size of the linking space, when one of the two schemas is unknown, we propose an approach which learns classification rules using existing linked data. This training set is given by an expert or computed by an automatic tool and validated. One classification rule expresses that a data description which contains a particular subsegment a in the values Y of a given property p may belong to a class c: $p(X,Y) \wedge subsegment(Y,a) \Rightarrow c(X)$. When new data has to be integrated in an existing RDF data source, these rules are used to identify the classes which have to be compared to these new data. In order to measure the quality of these classification rules, we compute different measures: confidence, support and lift. This approach has been evaluated on real RDF data sources containing electronic products[1].

The paper is organized as follows. In section 2, we present some related work. The problem is defined in section 3. Then, the approach used to learn classification rules is presented in section 4. In section 5, we present first experimental results. Finally, we conclude and give some future work.

2. RELATED WORK

Data linking is a computationally expensive task. Indeed, when each data item has to be compared to all data items described in a second data set, the number of comparisons grows quadratically with the number of data items. So, a lot of approaches are interested in reducing the number of comparisons that have to be done.

Blocking methods exploit an identified (subset of) attribute(s)

LWDM 2012, March 26–30, 2012, Berlin, Germany.

[1]this work has been done in the settings of a collaboration project with Thales Corporate Service company.

to split the data items into blocks. For example, persons that share the same first five characters of their last name belong to the same block[5]. Then, comparisons are made for data items that belong to the same block. Sorted Neighbourhood (SN) method sorts the data items using a sorting key [13]. A window of a given size is moved on the list of sorted data items and those belonging to the window are compared.

In Bi-gram methods, attribute values are converted into substrings of two characters (bi-gram) and sub-lists of all possible permutations are built using a threshold (between 0.0 and 1.0). The resulting bigram lists are sorted and inserted into an inverted index, which will be used to retrieve the corresponding record numbers in a block.

When the data are in conformity with an ontology, filtering method can be defined using ontology semantic. In [10], class disjunctions are used to reduce the reconciliation space but such approaches cannot be used when the data that will be integrated are not described using the ontology vocabulary.

The aim of association rule mining is to find rules satisfying defined requirements such as minimum support or minimum confidence. This kind of knowledge can be exploited in various applications (e.g market basket analysis, medical application, ...). It has also been used to solve classification problems where classification rules are defined as association rules where the conclusion is a class attribute [6].

3. PROBLEM STATEMENT

Let us consider a local RDF data source S_L and an external RDF data source S_E . We assume that the local data source S_L is described according to an OWL ontology O_L.

The main objective is to integrate external data of S_E in the local data source S_L by guarantying the Unique Name Assumption (UNA). Hence, we have to detect and eliminate redundant new data. Without a priori knowledge, a naive way to perform this task is to compare each new data item of S_E with all the existing ones contained in S_L. Hence, the size of the linking space is of $\mid S_E \mid \times \mid S_L \mid$.

One way to decrease the size of the linking space is to compare the new data with only a part of the whole space. This is possible, if for a given external data item d, one can detect which part of local data it should be compared to. We consider that these data should belong to the same class. Since these classes are unknown for the external data, we are faced to the problem of their automatic discovery.

Let TS be the set of same-as links between external and local data items that are validated by a domain expert. We consider that the linked pairs of data items are stored with their provenance information (external or local). The problem addressed in this paper consists in learning, given TS, classification rules which are able to predict the class of a given external data item.

4. CLASSIFICATION RULE LEARNING APPROACH

In this section, we present our learning approach of classification rules. It is used to discover the class of data when the schema is unknown. Indeed, some data property values may contain pieces of information that can be used to detect the class of the data items. For example, toponyms found in *rdfs:label* often contain types of geographical places ("Dresden Elbe Valley", "Place de la Concorde", "Copacabana Beach"), measure units can be used to determine the category of the products ("ohm", "Kg", "meter") and more generally strings that are included in object names (" Louvre Museum", "Iphone 4S") can help to classify them. These properties can be easily specified by an expert.

4.1 Value-based classification rule

Let p be a data type property that is used in the RDF triples of S_E. Let c be a class that belongs to the set of classes $\mathcal{C}$ of O_L. A value-based classification rule is in the form:

$$p(X,Y) \wedge subsegment(Y,a) \Rightarrow c(X)$$

where $subsegment(Y,a)$ expresses that the segment a occurs at least one time in the value Y. The way a value is split into segments is specified by a domain expert. One can use separation characters (e.g., ':', '-', ';', ' ') or n-grams.

4.2 Quality measures of Classification rules

In order to measure the quality of a classification rule, numerous measures are defined in the literature [3] like, $support$, $confidence$, $lift$, $specificity$, $coverage$, and so on. In our work, we have chosen to use three well-known quality measures that are:

- $support$, which allows to measure the proportion of data that satisfies the rule premise and that are instances of the class appearing in the rule conclusion. The support of a classification rule $R : p(X,Y) \wedge subsegment(Y,a) \Rightarrow c(X)$ can be formalized as follows:

$$support(R) = \frac{\mid \{X \mid p(X,Y) \wedge subsegment(Y,a) \wedge c(X)\} \mid}{\mid TS \mid}$$

  Hence, thanks to the support we can qualify a rule representativeness.

- $confidence$, which allows to measure the proportion of data that are instances of the class appearing in the conclusion among the data that satisfies the premise. The confidence of a classification rule $R : p(X,Y) \wedge subsegment(Y,a) \Rightarrow c(X)$ can be formalized as follows:

$$confidence(R) = \frac{\mid \{X \mid c(X)\} \mid}{\mid X \mid p(X,Y) \wedge subsegment(Y,a) \mid}$$

  Hence, the confidence degree expresses the rule precision without considering the possible proximity between the classes in the ontology.

- $lift$, which allows to measure the confidence of the classification rule divided by the proportion of instances of the class c in TS. The lift of a classification rule

$R : p(X,Y) \wedge subsegment(Y,a) \Rightarrow c(X)$ can be formalized as follows:

$$lift(R) = \frac{confidence(R)}{\frac{|\{X|c(X)\}|}{|TS|}}$$

Hence, the lift measures the deviation of the rule from the model of statistic independency of the rule premise and rule conclusion. The lift is a value between 0 and $infinity$: when the lift value is greater than 1, it indicates that the rule premise and the rule conclusion appear more often together than expected. This means that the occurrence of the subsegment in the property value has a positive effect on a data item assignment to the class of the conclusion. Furthermore, considering our objective, the higher the lift value is, the more the linking space size is reduced.

4.3 Learning Algorithm

The rule learning algorithm is based on the idea of finding frequent subsegments in frequent property instances of the data source S_E appearing in TS, i.e. the set of same-as links given by the domain expert. Let P be a set of properties that are selected by an expert. Let th be the support threshold. The learning algorithm is performed as follows:

- for each property instance $p(i,v)$, we split the value v into a set of segments and we create the fact $subsegment(v,a)$ for each segment a of v.
- for each property $p \in P$ and for each segment a, we compute the frequency of $p(X,Y) \wedge subsegment(Y,a)$. We select only $(p(X,Y), subsegment(Y,a))$ having a frequency greater than th.
- for each class $c \in O_L$ we compute its frequency in the set of class instances of S_L appearing in the linked data TS. We keep only the classes having a frequency greater than th. This frequency is computed only for the most specific classes of the ontology O_L.
- we compute the frequency of the conjunctions in the form $p(X,Y) \wedge subsegment(Y,a) \wedge c(X)$ having a frequency greater than th.
- we build the classification rules in the form: $p(X,Y) \wedge subsegment(Y,a) \Rightarrow c(X)$ and we compute their confidence and their lift degrees.

4.4 Classification Rule ordering

The application of a classification rule determines a data linking subspace for each instance of S_E. For a given new data item i, and a rule $R_k : p(i,v) \wedge subsegment(v,' seg') \Rightarrow c(i)$, the application of R_k leads to a data linking subspace d_{ik} composed of the set of pairs: (i,j) such that $i \in S_E$, $j \in S_L$ and $c(j)$.

The whole data linking space for the data item i is then composed of the union of all the data linking subspaces obtained thanks to the application of all the classification rules involving i.

Algorithm 1 Rule Learning Algorithm

Input: – TS: set of linked data.
– TS_E: set of property facts of S_E that belong to TS .
– P: a selected set of properties of S_E (all if no selection).
– th: support threshold.
– O_L: the local ontology.
Output: – $\mathcal{R}$: set of classification rules.

```
F ← ∅ ; R ← ∅
For Each p ∈ P Do
  For Each p(i,v) ∈ TS_E Do
    SEG ← split(v)
    For Each a ∈ SEG Do
      F ← F ∪ {subsegment(v,a)}
    End Each
  End Each
  For Each a ∈ getsubsegment(F,p) Do
    If support(p(X,Y) ∧ subsegment(Y,a)) > th Then
      F ← F ∪ {p(X,Y) ∧ subsegment(Y,a)}
    EndIf
  End Each
End Each
For Each c ∈ C Do
  If support(c(X) > th Then
    F ← F ∪ {c(X)}
  EndIf
End Each
For Each c ∈ getClasses(F) Do
  For Each p ∈ getProperties(F) Do
    For Each a ∈ getsubsegment(F,p) Do
      If support(p(X,Y) ∧ subsegment(Y,a) ∧ c(X)) > th
      Then
        R ← p(X,Y) ∧ subsegment(Y,a) → c(X)
        R.sup ← support(R)
        R.conf ← confidence(R)
        R.lift ← lift(R)
        R ← R ∪ R
      EndIf
    End Each
  End Each
End Each
return R
```

The above quality measures are used to rank the obtained subspaces for each data item of S_E. More precisely, the confidence degree is used first. In case of the same confidence degree, the lift measure is used in order to consider first the smaller subspaces.

We note that, the application of two different rules may lead to the same linking subspace. In this case, we ignore the one that is obtained by the rule having the worst confidence degree.

5. EXPERIMENT RESULTS

We have evaluated our approach on a real RDF data set provided by the french company Thales Corporate Service. The data set describes electronic products that are instances of classes described in a domain ontology (566 classes containing 226 classes in the leaves of the ontology). Each time a provider gives the company RDF files where products are described using the provider vocabulary, the company has to link these RDF data to find same-as links between the mentioned electronic products with the products described in its catalog and this task is time-consuming. The catalog is too huge (millions of instances) to use reconciliation approaches to compare an instance to the whole set of instances that are described in the catalog. We have evaluated the approach

on a set of 10265 reconciliations made by company experts.

In a provider document, one electronic product is described by a provider identifier (a part-number), and the name of the manufacturer. The expert has chosen the property *part-number* to predict the class. Indeed, this part-number is alphanumeric and contains pieces of information that can be useful to the linking process. Furthermore, almost all manufacturers provide products that belong to distinct classes, this is why this information cannot be used to determine product classes.

Partnumbers have been split into 7842 distinct segments (26077 occurrences) using non-alphabetical and non-numerical characters (e.g. space, '-','.', ...). When the support threshold th is fixed at 0.002, 7058 occurrences of segments are selected and 68 selected classes have more than 20 instances in the linked data set TS. With this support, 144 classification rules have been selected. Once they are selected, the rules are grouped using their confidence. The table 1 shows the number of rules that can be selected when the confidence varies from 1 to 0.4. For each confidence threshold, we have used TS to compute the number of decisions that can be made, the precision, and the recall. There are 2107 products that can be classified correctly using only 44 rules having the maximum confidence ($confidence = 1$). For example, segments such as "ohm", "63V", "CRCW0805" are used to detect instances of the class $Fixed - film\ resistance$, the segment "T83" is used to detect $Tantalum\ capacitors$. The instances that are described in an RDF external resource will only be compared to the instances that belong to their inferred classes. To show how the use of these classification rules can reduce the reconciliation space w.r.t their confidence, we have also computed the average lift of the set of learnt rules. For all thresholds, the lift is greater than 20. It means that using a rule that has a confidence of 1, even for a big class that represents 20% of the catalog, the linkage space can be divided by 5 for one instance. We have found interesting segments for 16 classes (appearing in the leaves of the ontology) among 67 frequent classes that are described in TS.

Table 1: Classification rule results

conf.	#rules	#dec.	prec.	recall	lift
1	44	2107	100%	29%	27
0.8	22	1224	96.9%	45.7%	24
0.6	13	712	92%	49.9%	24
0.4	17	1025	83.8%	60.1%	21

6. CONCLUSION

In this paper we have presented a new method to learn automatically classification rules from a given data set. In the context of data linking where the data are numerous and described by schemas that are different or (partially) unknown, these rules can be very useful to reduce the size of the linking space. Indeed, when the class of new data can be inferred, a linking method has to compare this data with the instances of the inferred class. The learnt classification rules are concise and easy to understand by an expert. The experiment that we have conducted on real data of the electronic domain have shown that the proposed approach is suitable for this domain. To show the generality of our approach we plan to test it on data from other domains. As future work, we plan to study how the learnt classification rules can be used to infer more general rules by exploiting the semantics of the subsumption between classes of the ontology.

7. REFERENCES

[1] Baxter, R., Christen, P., and Churches, T. A comparison of fast blocking methods for record linkage. In *KDD 2003 Workshops* (2003), pp. 25–27.

[2] Dong, X., Halevy, A., and Madhavan, J. Reference reconciliation in complex information spaces. In *Special Interest Group on Management of Data(ACM SIGMOD)* (NY, USA, 2005), pp. 85–96.

[3] Guillet, F., and Hamilton, H. J., Eds. *Quality Measures in Data Mining*, vol. 43 of *Studies in Computational Intelligence*. Springer, 2007.

[4] Hassanzadeh, O., Lim, L., Kementsietsidis, A., and Wang, M. A declarative framework for semantic link discovery over relational data. In *WWW* (2009), pp. 1101–1102.

[5] Jaro, M. A. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. 414–420.

[6] Liu, B., Hsu, W., and Ma, Y. Integrating classification and association rule mining. In *KDD* (1998), pp. 80–86.

[7] Nikolov, A., and Motta, E. Data linking: Capturing and utilising implicit schema-level relations. In *Proceedings of Linked Data on the Web workshop at 19th International World Wide Web Conference(WWW)'2010* (2010).

[8] Niu, X., Rong, S., Zhang, Y., and Wang, H. Zhishi . links results for oaei 2011. *Proceedings of the 10th International Semantic Web Conference* (2011).

[9] Raimond, Y., Sutton, C., and Sandler, M. B. Interlinking music-related data on the web. *IEEE MultiMedia 16*, 2 (2009), 52–63.

[10] Saïs, F., Pernelle, N., and Rousset, M.-C. Combining a logical and a numerical method for data reconciliation. *Journal on Data Semantics 12* (2009), 66–94.

[11] Volz, J., Bizer, C., Gaedke, M., and Kobilarov, G. Discovering and maintaining links on the web of data. In *Proceedings of the 8th International Semantic Web Conference* (Berlin, Heidelberg, 2009), ISWC '09, Springer-Verlag, pp. 650–665.

[12] Winkler, W. E. Overview of record linkage and current research directions. Tech. rep., Bureau of the Census, 2006.

[13] Yan, S., Lee, D., yen Kan, M., and Giles, C. L. Adaptive sorted neighborhood methods for efficient record linkage. In *International Conference On Digital Libraries* (2007), ACM, pp. 185–194.