A Logical System for Domain Information Retrieval using One-Class Clustering Technique

Raju. S
Dept. of Information Technology
Mahendra Engineering College
Namakkal, Tamilnadu.

Dr. Devarajan Gopal
Dept of Computer Science & Engg.
Mahendra Engineering College
Namakkal, Tamilnadu.

***Abstract*— Many-to-many knowledge linkage is an essential task in lots of domains. Furthermore, while historically knowledge linkage is performed among entities of the same type, it is very necessary to create linkage techniques that link between matching entities of different types as well. This work proposes a new one-to-many knowledge linkage method that links between entities. This model is based on a one-class clustering tree OCCT which characterizes the entities that ought to be linked together. The tree is built such that it is simple to understand & transform in to association rules, i.e., inner nodes describing the entities, while the tree represent their matching entities from the second dataset. It proposes splitting criteria and different pruning methods which can be used OCCT method was evaluated using datasets from various domains. The results prove the effectiveness of the proposed method and show that the OCCT yields better performance in terms of precision that recall (in most cases it is statistically significant) when compared to a C4.5 decision tree linkage process.**

Keywords*— Decision tree, many-many Data linkage, Pruning, one-class Clustering component.

1. INTRODUCTION

This work proposes a new one-to-many knowledge linkage method that links between entities. They propose splitting criteria and different pruning methods which can be used for OCCT method was evaluated using datasets from different domains. The effectiveness of the method and show that the OCCT yields better performance in terms of precision and recall (in most cases it is statistically significant) when compared to decision tree based linkage method.

In this paper a brand new knowledge linkage method aimed at performing one-to-many (and can be extended to many-to-many) linkage. Knowledge linkage refers to the task of matching entities from different knowledge sources that do not share a common identifier (i.e., a foreign key). Knowledge linkage is usually performed among entities of the same type. It is common to divide knowledge linkage in to types, namely, one-to-one and one-to-many.

In one-to-one knowledge linkage, the aim is to associate record in table TA with a single matching record in table TB. In the case of one-to-many knowledge linkage, the aim is to associate record in TA with or more matching records in TB. In addition, while knowledge linkage is usually performed among entities of the same type, the proposed knowledge linkage method can match entities of different types.

This links between the entities using a One-Class Clustering Tree (OCCT). In this algorithm they follow the pre-pruning approach. In pre-pruning, a branch is pruned in the work of the induction method if not of the feasible splits are found to be more beneficial than the current node.

2. ARCHITECTURE AND DESIGN

This method is based on a one-class clustering tree (OCCT) which characterizes the entities that ought to be linked together. The tree is built such that it is simple to understand and transform in to association rules, i.e., the inner nodes describing the first set of entities, features of their matching entities from the second dataset. The Architecture diagram of the proposed method is shown below.

2.1 Data Ingestion model

In this model there is the phase of the information retrieval from the user in the form of getting those inputs from which probabilistic calculations are made and further linkage of the data entities are given way. The user is asked a set of questions from for the probability entities, using this main result is made.

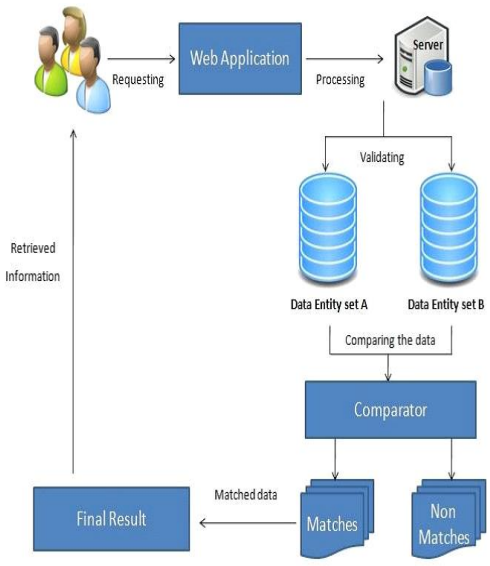

Figure 2.1 Architecture Diagram

2.2 Inducing a linkage model (Tree Construction)

The linkage model encapsulates the knowledge of which records are expected to match each other. The induction process includes deriving the structure of the tree. Building the tree requires deciding which attribute should be selected at

each level of the tree. The inner nodes of the tree consist of attributes from table *TA* only.

2.2.1 Splitting criteria:

The goal is to achieve a tree which contains a small amount of nodes is better generalize the data, avoid fitting, and it's simpler for the human eye to understand. Therefore, it is crucial to use an effective splitting criterion in order to build the tree. To choose to perform (i.e., splitting according to attribute a) if gain the most information out of this split. In this section, propose four criteria that can be used for evaluating the splitting of according to an attribute a. Each splitting criterion is used for measuring the similarity between two record sets T1 and T2, and is denoted by Sim (T1,T2). In the context of research, the similarity function that is defined by the selected splitting criteria is used in order to determine the at- tribute that creates the best split of a table; i.e., splits table T into two tables, T1 and T2, which differ from each other as much as possible. The attribute that achieves the best score (highest/lowest- depending on the splitting criterion) will be returned and used as the next split of the tree.

2.2.2 Pruning process:

Pruning is an important task in the tree induction process. A good pruning process will produce a tree which is accurate on one hand, and avoids over fitting on the other. There are two common approaches for pruning a decision tree: pre-pruning and post-pruning. In pre-pruning, a branch is pruned during the induction process if none of the possible splits are found to be more beneficial than the current node. In post-pruning, the tree is grown completely, followed by a bottom up process to determine which branches are not beneficial. In this algorithm follow the pre-pruning approach. This approach was chosen in order to reduce the time complexity of the algorithm.

2.3Applying OCCT for data linkage

During the linkage (i.e., test) phase, each possible pair of test records is tested against the linkage model in order to determine if the pair is a match. This process gives a value representing the probability of the record pair being a match. An initial score is max likelihood estimation. The algorithm is a Boolean value determining whether the given instances should be matched or not. First, the appropriate set of models is retrieved by following the values of record to the correct path of the tree (line 1). The likelihood for a match between the records is calculated by deriving the probability of each value in b, given all other values and the appropriate model link Instances(r(a),r(b))

2.3.1 Input:

TAB - set of matching instances, r(a) - an instance from table TA, r(b) – an instance from table TB, the threshold for match Output: q - a Boolean value determining if the records match or not The pseudo code of the linkage process

1: M ←the set of models matching the values of a and l ← 0
2: for each bi in b
3: li ← max(li,l)
4: l ← l + li
5: end for
6: l ← l*cardinality (a,b)
7: if l >= li then
8: q ← true
9: else
10: q ← false
11: end if
12: return q

2.4 Representing the leaves using probabilistic models

Once the construction of the tree is completed, each leaf contains a cluster (or set) of records. A set of probabilistic models is induced for each of the leaves. Each model *Mi* is used for deriving the probability of a value of attribute *bi B* from table *TB*, given the values of all other attributes from table *TB*. There are two motivations for performing this step. First, the sets of probabilistic models result in a more compact representation of the OCCT model. Second, by representing the matching records as a set of probabilistic models, the model is better generalized and avoids over fitting.

2.5 Linkage

During the linkage phase, each pair of the testing set is cross validated against the linkage method. The output is probability of the record pair match score is calculated using maximum likelihood estimation. The tested pair is classified as a match if the score is greater than a given, predefined, threshold or, if not, as a non-essential match

2.5.1 Estimation-Max Algorithm

Given a statistical model consisting of a set of observed data, unobserved latent data or missing values , and a vector of unknown parameters along with a similar function

L(:X,Z)= p**(X,Z** |)

The maximum likelihood estimate (MLE) of the unknown parameters is determined by the marginal likelihood of the observed data

$$L(\boldsymbol{\theta}; \mathbf{X}) = p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$$

However, this quantity is often intractable (e.g. if is a sequence of events, so that the number of values grows exponentially with the sequence length, find the exact calculation of the sum extremely difficult).

The Estimation Max algorithm seeks to identify the MLE of the marginal likelihood by iteratively applying the following two steps:

Expectation step (Estep):

Calculate the expected value of the log likelihood function, with respect to the conditional distribution of given under the current estimate of the Parameters

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathrm{E}_{\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)}}[\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})]$$

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$$

Maximization step (M step):

Find the parameter which maximizes the quantity:

The motivation is as follows. If the value of the parameters the usually find the value of the latent variables are maximizing the log-likelihood over all possible values of , either simply by iterating over , Conversely, if know the value of the latent variables can find an test inmate

of the parameter , typically by simply grouping the observed data points according to the value of the associated latent variable and averaging the values, of the points in each group. This suggests an iterative steps, in the case both are unknown:

1. First, initialize the parameters to some random values.
2. Compute the best value for given these parameter values.

Z just-computed value to compute a better estimate for the parameters. The Parameters associated with a particular value of will use only those data points whose associated latent variable has that value.

3Iterate steps 2 and 3 until Convergence.

3. NLP APPLICATIONS

EM is regularly used for data clustering in system learning and computer vision. In NLP, two prominent instances of the algorithm are the Baum-Welch algorithm (also known as forward-backward) and the algorithm for unsupervised stimulation of probabilistic context-free grammars. In psychometrics, EM is indispensable for estimating list of parameters and latent abilities of item response theory models. The ability to deal with missing data and observe unidentified variables, EM is becoming a useful tool and manages risk of a portfolio. It is based on a one class decision tree model that encapsulates the knowledge of which records should be linked to each other and also four possible splitting criteria and two possible pruning methods that can be used for inducing the data models. When applied the evaluation results are effective in different domains.

4. CONCLUSION

Link a record from a table *TA* with records from another table *TB*. The generated model is in the form of a tree in which the inner nodes represent attributes from *TA* and the leafs hold a compact representation of a subset of records from *TB* which are more likely to be linked with a record from *TA*, whose values are according to the path from the root of the tree to the leaf. The decision to use only the attributes of *TA* as the splitting attributes of the tree was taken in order to keep the generated model simple and easy to understand.

Thus, new OCCT tree can be easily used to obtain all records in r (a) in *TA*. Need only to traverse the OCCT tree using r (a). The leaf that is ultimately being reached holds all predicted matching records in *TB*. On the other hand, a regular classification tree which mixes *TA* and *TB* attributes is less convenient for this task. In particular, in order to get all records in *TB* that match a given r (a), to classify all possible links (i.e., r (a) with any record in *TB*) one-by-one as either match or not-match using the classification tree. This result is a tiresome process.

The future work of this paper plans to compare the focus is on the one-to-many case, the OCCT model can be used for many-to-many linkage, for example, by simply changing the roles of the two tables *TA* and *TB* and using table *TB* as the source table instead. Note that in the Movie lens dataset (movie recommender domain) to solve a many-to-many problem in which groups of users are matched with common matching movies. Although, the OCCT with other data linkage methods. It extends the OCCT model to the many-to-many cases and to handle continuous attributes.

REFERENCES

[1] Baxter.R , and, Gu.L, (2006)"Decision Models for Record Linkage," Data Mining, vol. 3755, Springer, pp. 146-160.
[2] Benjelloun.O, Garcia.H, Menestrina.D, Su.Q, Whang.S, and Widom.J, "Swoosh: a generic approach to entity resolu-tion," The VLDB Journal, vol. 18.
[3] Blockeel.H, Raedt.L.D, Ramon.J, (1998) "Top-Down Induction of Clustering Trees," ArXiv Computer Science e-prints. pp. 55-63.
[4] Christen.P, and Goiser.K, (2007) "Quality and Complexity Measures for Data Linkage and Reduplication," Quality Measures in Data Mining, Springer, pp. 127-151.
[5] Denis.F, Letouzey.F, and Gilleron.R, (2009) "Learning From Posi-tive and Unlabeled Examples," Algorithmic Learning Theory, Springer, pp. 71-85.
[6] De Comit'e.F, Denis.F, Gilleron.R, and Letouzey.F, (1999) "Positive and Unlabeled Examples Help Learning," Algorithmic Learning Theory, Springer, pp. 219-230.
[7] Domingo-Ferrer.J, and Torra.V, (2003) "Disclosure Risk Assessment in Statistical Micro data Protection via Advanced Record Linkage," Statistics andComputing, vol. 13, no. 4, pp.343-354.
[8] Elmagarmid.A.K, Elmeleegy.H, Quzzani.M, Yakout.M, and Qi.A, (2010) "Behavior Based Record Linkage," in Proc. of the VLDB Endowment, vol. 3, no. 1-2, pp. 439-448.
[9] Fellegi.I.P, and Sunter.A.B, (1969) "A Theory For Record Linkage," Journal of American statistical Society, vol. 64, no. 328, pp. 1183-1210.
[10] Frank.E, Hall.M.A, Holmes.G, Kirkby.R, and Pfahringer.B, (2005) "WEKA - A Machine Learning Workbench for Data Mining," The Data Mining and
[11] Knowledge Discovery Handbook, pp. 1305-1314.
[12] Henry.G, Ivie.S, Gatrell.H, and Giraud-Carrier.C, (2007) "A Metric-Based Machine Learning Approach to Genealogical Record Linkage," in Proc. of the 7th Annual Workshop on Technology for Family History and Genealogical Research.
[13] Larsen.M.D, and Rubin.D.B, (2001) "Iterative Automated Record Linkage Using Mixture Models," Journal of the American Statistical Association, vol. 96, no. 453. pp. 32-41.
[14] Li.C, Zhang.Y, and Li.X, (2009) "OcVFDT: One-Class Very Fast Decision Tree for One-Class Classification of Data Streams," in Proc. the 3rd Int. Workshop on Knowledge Discovery from Sensor Data, pp. 79-86, Paris, France.
[15] Rohde.D.J, Gallagher.M.R, Drinkwater.M.J, and Pimbblet.K.A, (2006) "Matching of Catalogues by Probabilistic Pattern Classification," Monthly Notices of the Royal Astronomical Soci-ety, vol. 369, no. 1, pp. 2-14.