Using Fuzzy Logic for Product Matching

K. Amshakala and R. Nedunchezhian

Abstract Product matching is a special type of entity matching, and it is used to identify similar products and merging products based on their attributes. Product attributes are not always crisp values and may take values from a fuzzy domain. The attributes with fuzzy data values are mapped to fuzzy sets by associating appropriate membership degree to the attribute values. The crisp data values are fuzzified to fuzzy sets based on the linguistic terms associated with the attribute domain. Recently, matching dependencies (MDs) are used to define matching rules for entity matching. In this study, MDs defined with fuzzy attributes are extracted from product offers and are used as matching rules. Matching rules can aid product matching techniques in identifying the key attributes for matching. The proposed solution is applied on a specific problem of product matching, and the results show that the matching rules improve matching accuracy.

Keywords Product matching · Data integration · Fuzzy logic · Matching dependency

1 Introduction

Product matching is a particular case of entity matching that is required to recognize different descriptions and offers referring same product. Although entity matching has received an enormous amount of effort in research [5], only modest work has been dedicated to product matching [7]. Product matching for e-commerce

K. Amshakala (✉)
Department of CSE and IT, Coimbatore Institute of Technology, Coimbatore, India
e-mail: amshakalacse@yahoo.in

R. Nedunchezhian
Sri Ranganathan Institute of Engineering and Technology, Coimbatore, India

G. S. S. Krishnan et al. (eds.), *Computational Intelligence, Cyber Security and Computational Models*, Advances in Intelligent Systems and Computing 246,
DOI: 10.1007/978-81-322-1680-3_20, © Springer India 2014

Websites introduces several specific challenges that make this problem much harder. In particular, there is a huge degree of heterogeneity in specifying name and descriptions of the same product by different merchants. For example, a desktop PC can be referred in different Websites using different terms like “Data Processor,” “Processor,” “Computer,” “Personal computer.” Similarly, offers on products come from multiple online merchants like jungle.com and pricegrabber.com. When exact string matching is not sufficient, synonym-based similarity matching can be used. WordNet ontology [12] is used to extract synonyms of product names. Not all products are described using same set of attributes. Furthermore, offers often have missing or incorrect values and are typically not well structured but merge different product characteristics in text fields. These challenges highlight the need for efficient entity matching technique that tolerates different data formats, missing attribute values, and imprecise information.

Fuzzy set theory and fuzzy logic proposed by Zadeh (1965) provide mathematical framework to deal with imprecise information. In a fuzzy set, each element of the set has an associated degree of membership. For any set *X*, a membership function on *X* is any function from *X* to the real unit interval [0,1]. The membership function which represents a fuzzy set *X*’ is usually denoted by $\mu(X)$. For an element *x* of *X*, the value $\mu_X(x)$ is called the membership degree of *x* in the fuzzy set *X*’. The membership degree $\mu_X(x)$ quantifies the grade of membership of the element *x* to the fuzzy set *X*’. The value 0 means that *x* is not a member of the fuzzy set; the value 1 means that *x* is fully a member of the fuzzy set. The values between 0 and 1 characterize fuzzy members, which belong to the fuzzy set only partially (Fig. 1).

A linguistic variable is a variable that apart from representing a fuzzy number also represents linguistic concepts interpreted in a particular context.

Functional dependencies, conventionally used for schema design and integrity constraints, are in recent times revisited for improving data quality [3, 4, 8]. However, functional dependencies based on equality function, often fall short in entity matching applications, due to a variety of information representations and formats, particularly in the Web data. Several attempts are made to replace equality function of traditional dependencies with similarity metrics. Fuzzy functional dependency (FFD) [10] is a form of FD that uses similarity metrics (membership functions) instead of strict equality function of FDs. FFDs are also used to find dependencies in databases with fuzzy attributes, whose domain has fuzzy values like high, low, small, large, young, old, etc. For example, the attributes size, price, and weight are fuzzy attributes in product database. A typical membership function for the price attribute is shown in Fig. 2.

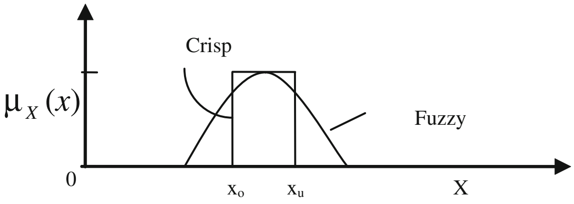


Fig. 1 Fuzzy membership function

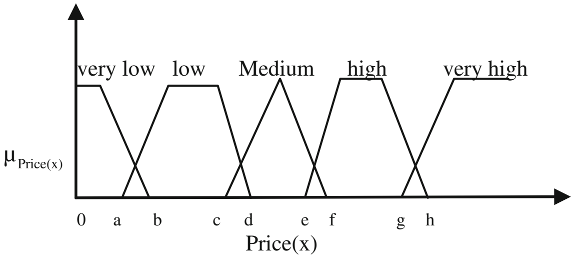


Fig. 2 Fuzzy membership function for price attribute

Recently, matching dependencies (MDs) [4, 8] are used for data quality applications, such as record matching. In order to be tolerant to different information formats, MDs target on dependencies with respect to similarity metrics, as an alternative of equality functions in conventional dependency. An MD expresses, in the form of a rule, that if the values of certain attributes in a pair of tuples are similar, then the values of other attributes in those tuples should be matched (or merged) into a common value. For example, the MD $\mathrm{R1}[X] \approx \mathrm{R2}[X] \rightarrow \mathrm{R1}[Y] = \mathrm{R2}[Y]$ says that if R1-tuple and R2-tuple have similar values for attribute X, then their values for attribute Y in R1 and R2 should be made equal. In practice, MDs are often valid in a subset of tuples and not on all the tuples of a relation. Along with FFDs, conditional matching dependencies (CMDs) proposed in [9] are used by the proposed work to infer matching rules that are appropriate for product matching. CMDs, which are variants of conditional functional dependencies (CFDs) [3], declare MDs on a subset of tuples specified by conditions. Approximate functional dependencies are also generalizations of the classical notion of a hard FD, where the value of X completely determines the value of Y not with certainty, but merely with high probability. Conditional functional dependencies (CFDs) [3] and approximate functional dependencies (AFDs) [6] differ with respect to the degree of satisfaction. While AFDs allow a small portion of tuples to violate the FD statement, conditional FDs are satisfied only by the tuples that match the condition pattern.

In this study, an information theory measure called entropy is used to define fuzzy functional dependencies and extensions of conventional FDs. Entropy of an attribute indicates the structuredness of the attribute. The reason behind using entropy as a dependency measure is that it captures probability distribution of the attribute values in a single value. The proposed approach defines MDs as fuzzy conditional matching dependencies (FCMDs). Two types of approximation are used in this approach. One is to compensate for uncertainty in matching similar values using FFDs [10] and the other approximation is to compensate for the fraction of tuples violating FDs using AFDs and CFDs. Experimentally, it is shown that the MDs and matching rules improve both the matching quality and efficiency of various record matching methods.

2 Preliminaries

In this section, we formally define functional dependency and explain how entropy is used to identify the presence of functional dependency between attributes.

2.1 Functional Dependency

A functional dependency (FD) $X \rightarrow Y$ is said to hold over sets of attributes X and Y on R if $\forall i, j$ if $ri[X] = rj[X]$ then $ri[Y] = rj[Y]$, where $r[X]$ denotes the tuple r projected on the attributes X.

2.2 Information Theory Measures

Entropy Let $P(X)$ be the probability distribution of an attribute X, the attribute entropy $H(X)$ is defined as

$$H(X) = -\sum_{x \in X} P(x) \log_2 P(x) \quad (1)$$

The entropy is a non-negative value, $H(X) \geq 0$ always. It may be interpreted as a measure of the information content of, or the uncertainty about, the attribute X. It is also called as marginal entropy of an attribute [2].

The joint entropy $H(X, Y)$ between any two attributes X, Y can be computed using joint probability distribution of the two attributes as follows

$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} P(x, y). \log_2 P(x, y) \quad (2)$$

where $P(x, y)$ is the joint probability distribution of the attributes X and Y. Also, $H(X, X) = H(X)$ and $H(X, Y) = H(Y, X)$.

Theorem 1 *Functional dependency $X \rightarrow Y$, holds if and only if $H(X,Y) = H(X)$.*

Theorem 1 indicates that the FD $X \rightarrow Y$ holds, if the joint entropy of X and Y is the same as that of X alone. By computing the joint entropy between attribute pairs and attribute entropies for all the attributes in the given table, all those functional dependencies that are true can be determined. When Theorem 1 is put in other words, for the functional dependency $X \rightarrow Y$ to hold true, the difference between $H(X, Y)$ and $H(X)$ must be equal to zero [11].

$$H(X, Y) - H(X) = 0 \quad (3)$$

2.3 Functional Dependency Extensions

Traditional functional dependencies are used to determine inconsistencies at schema level, which is not sufficient to detect inconsistencies at data level. Fuzzy attributes with crisp domain in a relation have to be fuzzified before applying FD discovery methods on the data. Let us assume that an attribute A with crisp domain when fuzzfied using different fuzzy sets results in fuzzy columns fA_1, $fA_2\ldots$ fA_n. The table is partitioned into equivalence classes that include tuple ids of those tuples that qualify as equal values along different linguistic variables associated with the attribute. The relational table is also partitioned based on crisp data values over the crisp attribute and with each linguistic dimension separately. Compute the marginal entropy of all the crisp attributes (not fuzzified) in the given relation using Eq. 1. From the projected fuzzified table, partition the tuples along the fuzzy columns fA_1, $fA_2\ldots\ldots$ $fA_\mathrm{n.}$ Entropy for the fuzzy columns is computed by considering data values with membership degree greater than the membership threshold θ as equal. Joint entropy of fuzzified attribute A and a crisp attribute B is computed as the cumulative sum of joint entropy of the fuzzy columns and crisp attribute B

$$H(AB) = \sum_{i=1}^{n} H(fA_iB) \tag{4}$$

Theorem 1 is applied to check whether any functional dependency exists between crisp and fuzzified attributes. If $H\ (AB) = H(B)$, then $B \rightarrow \theta\ A$ is true. Further, dependencies that apply conditionally appear to be particularly needed when integrating data, since dependencies that hold only in a subset of sources will hold only conditionally in the integrated data. A CFD extends an FD by incorporating a pattern tableau that enforces binding of semantically related values. Unlike its traditional counterpart, the CFD is required to hold only on tuples that satisfy a pattern in the pattern tableau, rather than on the entire relation. The minimum number of tuples that are required to satisfy the pattern is termed as the support of CFD.

When a relation U with m tuples is considered, the support entropy is calculated as follows

$$H_s = -\left(\frac{m_k}{m}\log_2\left(\frac{m_k}{m}\right) + m_r\left(\frac{1}{m}\right)\log_2\left(\frac{1}{m}\right)\right) \tag{5}$$

H_s is nothing but the entropy of candidate that has at least one partition with $\mathrm{m_k}$ tuples, where $\mathrm{m_k}$ is the minimum number of tuples that should have the same constant value for the CFD to get satisfied. m_r is the remaining number of tuples in the relation, $m_r = m - m_k$. Under certain circumstances where the relational table has uncertain data, the functional dependency $X \rightarrow Y$ may not be strictly satisfied by all the tuples in a relation. When few number of tuples violate the functional dependency, then the difference between the joint entropy $H(X, Y)$ and the marginal entropy $H(X)$ may be close to 0, but not strictly equal to zero

$$H(X, Y) - H(X) \approx 0 \tag{6}$$

Such dependencies are called as approximate FDs [6].

3 Proposed Approach

3.1 FCMD Discovery Algorithm

Matching entities gathered from multiple heterogeneous data sources place lot of challenges because the attributes describing the entities may have missing values or the values may be represented using different encodings. MDs are a recent proposal for declarative duplicate resolution tolerating uncertainties in data representations. Similar to the level-wise algorithm for discovering FDs [6], the FCMD discovery algorithm also considers the left-hand-side attributes incrementally, i.e., traverse the attributes from a smaller attribute set to larger ones to test the possible left-hand-side attributes of FDs. The following pruning rules are used to reduce the number of attribute sets to be tested for the presence of FCMDs.

3.2 Pruning Rules

3.2.1 Pruning Rule 1: Support Entropy-Based Pruning

Candidates that do not even have one partition with size greater than or equal to k need not be verified for the presence of FCMDs. Candidates with support less than k need not be checked. Supersets of such candidates are also not k-frequent. So, candidates with entropy higher than H_s can be removed from the search space.

3.2.2 Pruning Rule 2: Augmentation Property-Based Pruning

The supersets of the LHS candidate of the FCMDs need not be verified further, once when a FCMD is satisfied by the candidate. This pruning rule prevents discovering redundant FCMDs.

The time complexity of the FCMD discovery algorithm varies exponentially with respect to the number of attributes and varies linearly with respect to the number of tuples (Table 1).

Table 1 FCMD discovery algorithm

Input: Sample training relation

Output: Set of Matching Dependencies (MD)

1. Compute the marginal entropy of all the crisp attributes (not fuzzified) in the given relation using Eq. 1
2. The fuzzy attributes are associated with appropriate membership functions and they are projected as fuzzy columns in the given relation
3. From the projected fuzzified table, partition the tuples along the fuzzy columns $fA1, fA_2\ldots\ldots fA_\mathrm{n}$
4. Entropy for the fuzzy columns are computed by considering data values with membership degree greater than the threshold as equal
5. Joint Entropy of fuzzy attribute A and a crisp attribute B is computed as the cumulative sum of joint entropy of the fuzzy columns and crisp attribute B
$$H(AB) = \sum_{i=1}^{n} H(fA_iB)$$
6. Using the user specified support threshold k, the support entropy $\mathrm{H_s}$ is computed using Eq. 6
7. If the RHS and LHS value of Eq. 5 is greater than or equal to H_s then the attribute B is the possible candidate to act as LHS of FCMD (Rule 1)
8. Equation 4 is used to check if any functional dependency exists between crisp and fuzzified attributes
$$\text{If } H(AB) - H(B) \approx 0 \text{ then } B \rightarrow_\theta A \text{ is true}$$
9. Add the discovered Matching dependency to the set MD
10. Repeat through step 5 for all candidates level by level (Pruning rule 2)

As the probability distribution is represented using a single entropy measure, the set closure and set comparison operations are not required to test the presence of FCMDs.

4 Experimental Results

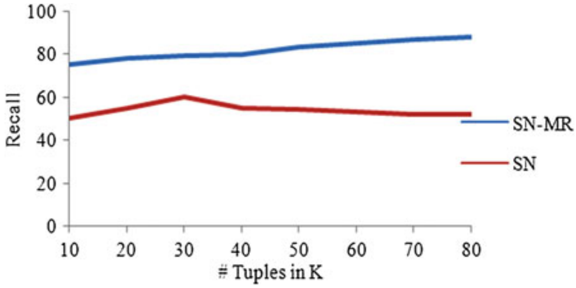
4.1 Experimental Setup

Experiments were carried out on a 2.16-GHz processors with a 2 GB RAM with Windows XP operating system. The implementation of this project is done using Java.

4.2 Dataset

The product catalog for electronic goods was collected from 20 Websites and consolidated as a dataset with 150 entities. On average, the maximum number of duplicates for an entity is 10. The data records are randomly duplicated, and dataset with 10–80 K records were created. The product catalog has four fields including product_ID, Product_Name, Manufacturer_Name, and Price.

Fig. 3 Recall versus no. of tuples

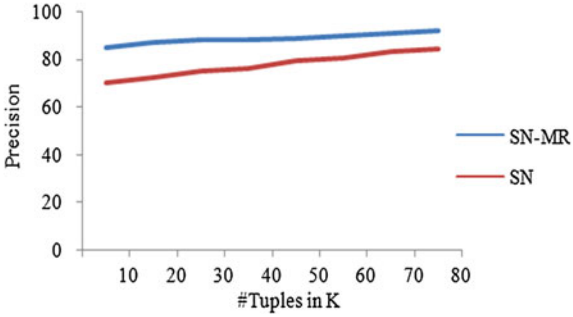


A record linkage tool called Fine-grained Records Integration and Linkage tool (FRIL) provides a rich set of tools for comparing records [13]. Furthermore, FRIL provides a graphical user interface for configuring the comparison of records. FRIL can be configured to choose different field similarity metrics and different record comparison techniques.

For testing the product catalog integration, FRIL is configured to use Q-grams for comparing Product_Name field, exact string matching for Manufacturer_Name, and numeric approximation for Price field. Sorted neighborhood method is used for record comparisons [1]. The proposed FCMD method first extracts the matching rules and then uses as rules for sorted neighborhood method. Precision and recall are the two measures used to measure the accuracy of the results returned by the record matching algorithms. Precision is used to check whether the results returned are accurate, and recall is used to check whether the results returned are complete. Recall and precision measured for the proposed approach and FRIL are shown in Figs. 3 and 4, respectively.

The matching rules discovered by FCMD method has higher recall than that used by the SN method, because of using dynamic discovery of rules rather than static rule. The precision of the results produced by FCMD method is shown in Fig. 4. The results produced by SN with matching rules discovered by FCMD algorithm includes lesser number of incorrect results, compared to that of SN, because of having matching rules with higher support.

Fig. 4 Precision versus no. of tuples



5 Conclusion

The problem of identifying similar products described used fuzzy attributes is of major concern when integrating product catalogs and matching product offers to products. Most previous work is based on predefined matching rules and supervised way of detecting duplicates using trained datasets. MDs represented using fuzzy dependencies and other FD extensions defined using entropy are proposed in this study. The proposed work also uses synonym-based string field matching, which helps in detecting duplicate records that are missed by exact string matching methods. Fuzzy attributes are modeled using fuzzy functional dependencies. This entity matching technique can be used to identify duplicates in datasets generated on the fly and do not require hand-coded rules to detect duplicate entities, which makes it more suitable for product matching.

References

1. Christen, P. (2012), A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication, IEEE Transactions on, Knowledge and Data Engineering, 24(9), 1537–1555.
2. Divesh S and Suresh V(2010), Information Theory for Data Management, Tutorial in Proceedings of the ACM SIGMOD Conference on Management of Data, 1255–1256.
3. Fan W, Geerts F, (2011) Foundations of Data Quality Management, Synthesis Lectures on Data Management, Morgan & Claypool Publishers.
4. Fan, W., Gao, H., Jia, X., Li, J, and Ma, S. (2011). Dynamic Constraints for Record Matching. The VLDB Journal, 20(4), 495–520.
5. Köpcke, H., and Rahm, E. (2010). Frameworks for Entity Matching: A Comparison. Data & Knowledge Engineering, 69(2),197-210.
6. Liu, J., Li, J., Liu, C., & Chen, Y. (2012). Discover Dependencies from Data-A Review. IEEE Transactions on Knowledge and Data Engineering, 24(2), 251–264.
7. Papadimitriou, P., P. Tsaparas, A. Fuxman and L. Getoor, (2013). TACI: Taxonomy Aware Catalog Integration, IEEE Transactions on Knowledge and Data Engineering, 25: 1643–1655.
8. Song, S. and L. Chen, (2009). Discovering Matching Dependencies, In Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp: 1421–1424.
9. Song, S., L. Chen and J.X. Yu, (2010). Extending Matching Rules with Conditions, Proceedings of the 8th International Workshop on Quality in Databases, 13–17 September.
10. Wang, S. L., Shen, J. W., & Hong, T. P. (2010). Dynamic Discovery of Fuzzy Functional Dependencies Using Partitions. Intelligent Soft Computation and Evolving Data Mining: Integrating Advanced Technologies, 44.
11. Yao, Y.Y., (2003). Information-Theoretic Measures For Knowledge Discovery and Data Mining. Entropy Measures, Maximum Entropy Principle Emerging Applications, 119: 115-136.
12. Zadeh, L.A., (1965). Fuzzy sets. Information and control, 8(3), 338–353. http://wordnet.princeton.edu, 2009
13. http://fril.sourceforge.net.