Improving Record Linkage Accuracy with Hierarchical Feature Level Information and Parsed Data

Yun Zhou*, Minlue Wang, Valeriia Haberland, John Howroyd, Sebastian Danicic, and J. Mark Bishop

Tungsten Centre for Intelligent Data Analytics (TCIDA),
Goldsmiths, University of London, United Kingdom
{y.zhou;m.wang;v.haberland;j.howroyd;s.danicic;m.bishop}@gold.ac.uk

Abstract. Probabilistic record linkage is a well established topic in the literature. Fellegi-Sunter probabilistic record linkage and its enhanced versions are commonly used methods, which calculate match and non-match weights for each pair of records. Bayesian network classifiers – naive Bayes classifier and TAN have also been successfully used here. Recently, an extended version of TAN (called ETAN) has been developed and proved superior in classification accuracy to conventional TAN. However, no previous work has applied ETAN to record linkage and investigated the benefits of using naturally existing hierarchical feature level information and parsed fields of the datasets. In this work, we extend the naive Bayes classifier with such hierarchical feature level information. Finally we illustrate the benefits of our method over previously proposed methods on 4 datasets in terms of the linkage performance (F_1 score). We also show the results can be further improved by evaluating the benefit provided by additionally parsing the fields of these datasets.

Keywords: Probabilistic record linkage; Naive Bayes classifier; TAN and ETAN; Hierarchical feature level information; Parsed fields

1 Introduction

Record linkage (RL) [1] proposed by Halbert L. Dunn (1946) refers to the task of finding records that refer to the same entity across different data sources. These records contain identifying fields (e.g. name, address, time, postcode etc.). The simplest kind of record linkage, called deterministic or rules-based record linkage, requires all or some identifiers are identical giving a deterministic record linkage procedure. This method works well when there exists a common key identifier within the datasets. However, in *real world* applications, deterministic record linkage is problematic because of the incompleteness and privacy protection [2] of a key identifier field.

* The authors would like to thank the Tungsten Network for their financial support. (Submission date: Wednesday 9th March, 2016)

To mitigate this problem, probabilistic record linkage (also called fuzzy matching) was developed, which takes a different approach to record linkage by taking into account a wider range of potential identifiers. This method computes weights for each identifier based on its estimated ability to correctly identify a match or a non-match, and uses these weights to calculate a score (usually *log-likelihood* ratio) that two given records refer to the same entity.

Record-pairs with scores above a certain threshold are considered to be matches, while pairs with scores below another (lower) threshold are considered to be non-matches; pairs that fall between these two thresholds are considered to be "possible matches" and can be dealt with accordingly (e.g., human reviewed, linked, or not linked, depending on the requirements). Whereas deterministic record linkage requires a series of potentially complex rules to be programmed ahead of time, probabilistic record linkage can be *trained* to perform well with much less human intervention.

Good results from probabilistic record linkage may be best achieved where field structure is well defined and more specific. For example, patient addresses in medical records could be better compared where addresses are represented with a fine grained structure (i.e., premises, street number, street name, town name, city name, and postcode). This field structure could be achieved by splitting unstructured/semi-structured addresses with address parsing. Moreover, there are hierarchical restrictions between these fields, which are useful to avoid unnecessary computation of field comparison [3, 4]. These hierarchical restrictions can be mined from the semantic relationships between fields, which widely exist in *real world* record matching problems. An example of this occurs especially in address matching. For example, two restaurants with the same name located in the two cities should be more likely identified as two different restaurants, because they are probably two different branches in two cities. In this case, the city locations have higher importance than the restaurant names.

In this paper we investigate how to use these hierarchical restrictions and standardized record-pairs to improve record linkage accuracy. Also we propose an extended naive Bayes classifier to model the record linkage problem. The paper is organized as follows. In Section 2 we discuss related work in record linkage. In Section 3 we discuss the framework of a general record linkage process. In Section 4 we discuss the data cleaning method and address parser used in this paper. In Section 5 we discuss the standard probabilistic record linkage model. In Section 6 we propose our improved record linkage model with elicited hierarchical restrictions. In Section 7 we report on the experiments of 4 different real-world datasets. Our conclusions are in Section 8.

2 Related Work

Fellegi-Sunter probabilistic record linkage (PRL-FS) [5] is one of the most commonly used methods. It assigns a match/non-match weight for each corresponding field of record-pairs based on *log-likelihood* ratios. For each record-pair, a composite weight is computed by summing each field's match or non-match

weight (as summarised in Section 5). The resulting composite weight is then compared to the aforementioned thresholds to determine whether the record-pair is classified as a match, possible match (hold for human review) or non-match. Determining where to set the match/non-match thresholds is a balancing act between obtaining an acceptable sensitivity (or recall, the proportion of truly matching records that are classified match by the algorithm) and positive predictive value (or precision, the proportion of records classified match by the algorithm that truly do match).

In PRL-FS method, a match weight will only be used when two strings exactly agree in the field. However, in many *real world* problems, two strings describing the same field may not exactly (character-by-character) agree with each other because of multiple representations and typographical error (mis-spelling). For example, *Andy* and *Andrew* could be two representations of a person's first name. Moreover, *Andy* could be misspelled as *Andi*. However, the field (first name) comparisons (*Andy*, *Andrew*) and (*Andy*, *Andi*) are both treated as non-match in PRL-FS.

The US Census Bureau reports [6] that, because of multiple representations and mis-spellings, 25% of first names did not agree character-by-character among medical record-pairs that were from the same person. To obtain better performance in *real world* usage, Winkler proposed an enhanced PRL-FS method (PRL-W) [7] that takes into account field similarity (of two strings for a field within a record-pair) in the calculation of field weights, and showed better performance of PRL-W compared to PRL-FS [8]. In this paper, we also use Jaro-Winkler similarity to measure the differences between fields of two records. These field difference values and known record linkage labels are used to train the record linkage model.

Probabilistic graphical models for classification such as naive Bayes (NBC) and tree augmented naive Bayes (TAN) are also used for record linkage [9], where the single class variable contains two states: match and non-match. These models can be easily improved with domain knowledge. For example, monotonicity constraints (i.e. a higher field similarity value indicating a higher degree of 'match') can be incorporated to help reduce overfitting in classification [10]. Recently, a state-of-the-art Bayesian network classifier called ETAN [11, 12] has been proposed and shown to outperform NBC and TAN in many cases. ETAN relaxes the assumption about independence of features, and does not require features to be connected to the class.

As discussed in our previous work [13], we have applied ETAN to probabilistic record linkage, and extended naive Bayes classifier (referred to as HR-NBC) by introducing hierarchical restrictions between features. The results have shown the benefits of using hierarchical restrictions under some settings. In this paper, we introduce a standard framework for the general record linkage problem. Then, we discuss the address parsing method. Finally, we investigate if the record linkage performance could be further improved by using the address parser on 2 datasets.

3 Framework

Köpcke and Rahm [14] reviewed numerous studies of record linkage which were mainly concerned with structured and often relational data, while semi-structured and unstructured data received much less attention. It has to be noted that the difference between fully structured and semi-structured data is not strictly defined and can vary across different domains and data representations. In this paper, we focus on relational structured and semi-structured data which are defined below.

Structured data: Fully structured data is considered to be relational data where each field has a designated value if applicable. For example, if a field is designated for the first part of address such as a house number and street name, then the corresponding field in each record should contains this part of the address.

Semi-structured data: Semi-structured data implies imperfect field alignment where the data items might appear in any field which is not necessarily designated to these data items. For example, the whole address may be stored textually in a single field or may be assigned to multiple fields without any particular designation of purpose; so that, the postal town may appear in any one of them. Hence, this imperfection in data structure poses a challenge to link the records according to those fields. Non-relational data such as XML documents, may also be considered as semi-structured [15], but this becomes arguable when there is a well defined consistent schema.

Data of any structure might have noise consisting of misspellings, invalid data (e.g. $(000)000 - 000$ for telephone number), missing data, abbreviations and so on [16]. This 'noise' introduces more uncertainty into the matching of records. These challenges may be solved by data cleaning [16] such as filling in missing values, parsing fields with unstructured and ambiguous data and so on. In particular, field parsing may resolve ambiguous field alignment or split fields into constituent parts such as house number and street name.

Figure 1 shows a process of record linkage which is modelled in this paper. The input data from two sources can be either structured or semi-structured and it requires pre-processing in the case of ambiguous address fields as discussed above. In our work, a pre-processing step only resolves address fields in order to identify specific address components such as house number and street name, which might appear together in a single field. The next step is to check if there exists hierarchical restrictions (as discussed in Section 1) within the dataset, which determines a choice of model. Finally, we match two records by applying one of the discussed probabilistic models or Bayes classifiers; that is, PRL-W, TAN, ETAN, NBC or HR-NBC. Record-pairs are classified into either match or non-match classes as discussed in the remainder.

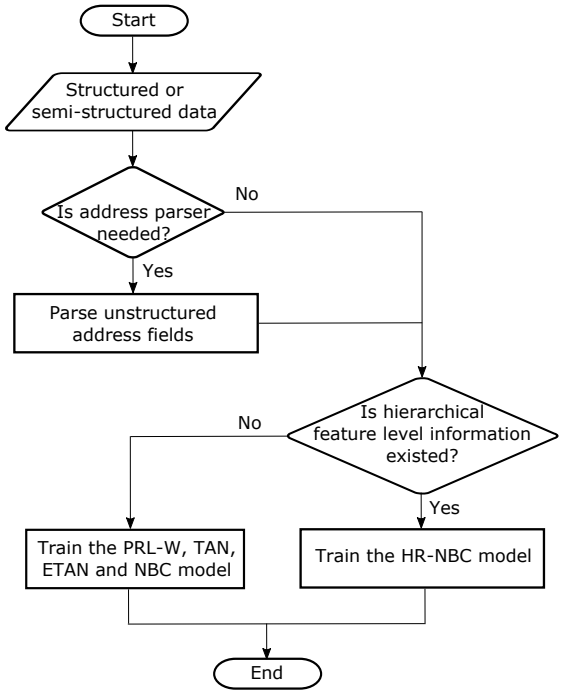


Fig. 1. The framework of linking record-pairs in this paper.

4 HMM-based Address Parser

Field comparison is a fundamental process for probabilistic record linkage methods and Bayes classifiers. However, raw data from the web or real-world databases are noisy and sometimes do not have a well-defined structure for carrying out these comparisons. Therefore, data cleaning and standardisation are usually applied before record linkage. For instance, address is a commonly used field for records containing information about people and organisations, but often exhibits variations ("roman street" vs. "roman st."). Proper segmentation of raw addresses into a set of meaningful fields (street name, street type) would be an important step for the subsequent comparison task.

In this work, we use a Hidden Markov Model (HMM) for parsing addresses as described in [17, 18]. Each address input string is firstly tokenised into a set of words and then each word is assigned with an observation label by using a number of look-up tables. The reference tables contain information about postal codes, city names or county names from postal authorities or governments. The assignments follow a greedy matching algorithm, which prefers assigning labels over a sequence of words than individual word. For example, even though "stoke" and "trent" are in a "sub-locality" table, "stoke on trent " is observed as "city" because the whole sequence of words can be found in a "city" table. Automati-

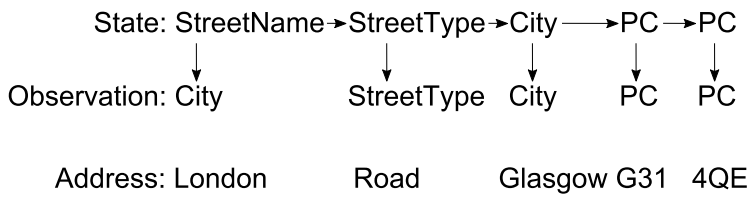


Fig. 2. An example of using HMM for parsing addresses.

cally generated observation labels are not good enough for parsing an arbitrary address because the greedy assignment algorithm is deterministic so that each word is always given a particular label. An illustrated example is presented in Figure 2. "London" is observed as "City" while in this case it is more likely to be a street name in Glasgow. A hidden Markov model is able to recover from this incorrect observation by considering an underlying state sequence.

The discrete-time HMM consists of observation sequences $\{x_{it}\}_{t=1}^{T_i}$ and corresponding hidden state sequence $\{z_{it}\}_{t=1}^{T_i}$ for each data item i. T_i is the length of i^{th} address. The transition probabilities between states are given by π_{jk}, where $\pi_{jk} = P(z_{t+1} = k \mid z_t = j)$ is the probability of transitioning to state k given the current state j. The probability of generating observations given states is governed by an observation matrix O, where $O_{jm} = P(x = m \mid z = j)$. Training is done by filling the matrices from labelled addresses and the best sequence of states given a test address can be generated by the standard Viterbi algorithm. Back to the example in Figure 2, if we see more examples containing transition from street name to street type than city to street type in the training data, the most likely state for "London" might be correctly marked as street name.

5 Probabilistic Record Linkage

5.1 PRL-FS and PRL-W

Let us assume that there are two datasets A and B of n-tuples of elements from some set F. (In practice F will normally be a set of a strings.) Given an n-tuple a we write a_i for the i-th component (or field) of a.

Matching If an element of $a \in A$ is the representation of the the same object as represented by an element of $b \in B$ we say a *matches* b and write $a \sim b$. Some elements of A and B match and others do not. If a and b do not match we write $a \nsim b$. We write $M = \{(a, b) \in A \times B | a \sim b\}$ and $U = \{(a, b) \in A \times B | a \nsim b\}$. The problem is then, given an element x in $A \times B$ to define an algorithm for deciding whether $x \in M$ or $x \in U$.

Comparison Functions on Fields We assume the existence of a function:

$$cf : F \times F \to [0, 1].$$

With the property that $\forall h \in F$, $cf(h, h) = 1$. We think of cf as a measure of *how similar* two elements of F are. Many such functions exist on strings including the normalised Levenshtein distance or Jaro-Winkler. In conventional PRL-FS method, its output is either 0 (non-match) or 1 (match). In PRL-W method, a field similarity score (Jaro-Winkler distance [7, 19]) is calculated, and normalized between from 0 and 1 to show the degree of match.

Discretisation of Comparison Function As in previous work [8], rather than concern ourselves with the *exact* value of $cf(a_i, b_i)$ we consider a set of $I_1, \cdots I_s$ of disjoint intervals exactly partitioning the closed interval $[0, 1]$. These intervals are called *states*. We say $cf(a_i, b_i)$ is in state k to mean $cf(a_i, b_i) \in I_k$.

Given an interval I_k and a record-pair (a, b) we define two values[1]:

- $m_{k,i}$ is the probability that $cf(a_i, b_i) \in I_k$ given that $a \sim b$.
- $u_{k,i}$ is the probability that $cf(a_i, b_i) \in I_k$ given that $a \nsim b$.

Given a pair (a, b), the *weight* $w_i(a, b)$ of their i-th field is defined as:

$$w_i(a, b) = \sum_{k=1}^{s} w_{k,i}(a, b)$$

where

$$w_{k,i}(a, b) = \begin{cases} \ln(\frac{m_{k,i}}{u_{k,i}}) & \text{if } cf(a_i, b_i) \in I_k \\ \ln(\frac{1-m_{k,i}}{1-u_{k,i}}) & \text{otherwise.} \end{cases}$$

The *composite weight* $w(a, b)$ for a given pair (a, b) is then defined as

$$w(a, b) = \sum_{i=1}^{n} w_i(a, b).$$

5.2 The EM Estimation of Parameters

In practice, the set M, the set of matched pairs, is unknown. Therefore, the values $m_{k,i}$, and $u_{k,i}$, defined above, are also unknown. To accurately estimate these parameters, we apply the expectation maximization (EM) algorithm with randomly sampled initial values for all these parameters.

[1] Note in conventional PRL-FS method [5], two fields are either matched or unmatched. Thus the k of $m_{k,i}$ can be omitted in this case.

The Algorithm

1. Choose a value for p, the probability that an arbitrary pair in $A \times B$ is a match.
2. Choose values for each of the $m_{k,i}$ and $u_{k,i}$, defined above.
3. *E-step*: For each pair (a, b) in $A \times B$ compute

$$g(a,b) = \frac{p \prod_{(a,b)\in A\times B} \prod_{k=1}^{s} m'_{k,i}(a,b)}{p \prod_{(a,b)\in A\times B} \prod_{k=1}^{s} m'_{k,i}(a,b) \quad + \quad (1-p) \prod_{(a,b)\in A\times B} \prod_{k=1}^{s} u'_{k,i}(a,b)} \tag{1}$$

where

$$m'_{k,i}(a,b) = \begin{cases} m_{k,i} & \text{if } cf(a_i, b_i) \in I_k \\ 1 & \text{otherwise.} \end{cases}$$

and

$$u'_{k,i}(a,b) = \begin{cases} u_{k,i} & \text{if } cf(a_i, b_i) \in I_k \\ 1 & \text{otherwise.} \end{cases}$$

4. *M-step*: Then recompute $m_{k,i}$, $u_{k,i}$, and p as follows:

$$m_{k,i} = \frac{\sum_{(a,b)\in A\times B} g'_{k,i}(a,b)}{\sum_{(a,b)\in A\times B} g(a,b)}, \quad u_{k,i} = \frac{\sum_{(a,b)\in A\times B} \tilde{g'}_{k,i}(a,b)}{\sum_{(a,b)\in A\times B} 1 - g(a,b)}, \quad p = \frac{\sum_{(a,b)\in A\times B} g(a,b)}{|A \times B|} \tag{2}$$

where

$$g'_{k,i}(a,b) = \begin{cases} g(a,b) & \text{if } cf(a_i, b_i) \in I_k \\ 0 & \text{otherwise.} \end{cases}$$

and

$$\tilde{g'}_{k,i}(a,b) = \begin{cases} 1 - g(a,b) & \text{if } cf(a_i, b_i) \in I_k \\ 0 & \text{otherwise.} \end{cases}$$

In usage, we iteratively run the E-step and M-step until a convergence criterion is satisfied: say $\sum(|\Delta m_{k,i}|) \leq 1 \times 10^{-8}$, $\sum(|\Delta u_{k,i}|) \leq 1 \times 10^{-8}$, and $|\Delta p| \leq 1 \times 10^{-8}$. Having obtained values for $m_{k,i}$ and $u_{k,i}$, we can then compute the composite weight (the natural logarithm of $g(a, b)$) for each pair defined earlier.

In our implementation, we set the decision threshold as 0.5, and do not consider possible matches. Because using a domain expert to manually examine these possible matches is expensive. Thus, the record-pair (a, b) is recognized as a match when $g(a, b) > 0.5$; otherwise it is a non-match.

6 Bayesian Network Classifiers for Record Linkage

In this section we discuss different Bayesian network classifiers (NBC, TAN and ETAN) for record linkage. After that, we discuss the hierarchical structure between features, and the proposed hierarchical restricted naive Bayes classifier (HR-NBC).

6.1 The Naive Bayes Classifier

For each pair of records, (a, b), we let $\boldsymbol{f}$ denote the feature vector $(f_i)_{k=1}^{n}$ and C be a binary class variable. Moreover, $f_i = k$ where $cf(a_i, b_i) \in I_k$, and $C = u, m$ denoting non-match, match respectively.

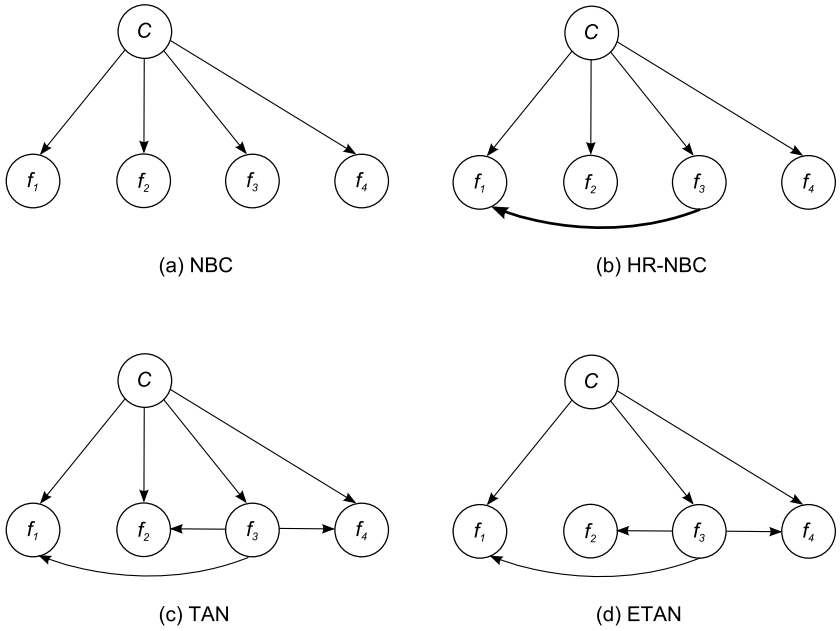


Fig. 3. The graphical representation of NBC, HR-NBC, TAN, ETAN. The bold arrow represents the dependency introduced by hierarchical feature level information.

The model calculates the probabilities $P(C = u)$ of $P(C = m)$, given the feature values (discretised distance for each field-value pair). This can be formulated as:

$$P(C|\boldsymbol{f}) = P(C) \times \frac{P(\boldsymbol{f}|C)}{P(\boldsymbol{f})} \tag{3}$$

In the naive Bayes classifier (Figure 3(a)), we assume conditional independence of features, where $P(\boldsymbol{f}|C)$ can be decomposed as $P(\boldsymbol{f}|C) = \prod_{i=1}^{n} P(f_i|C)$. Thus, equation (3) becomes:

$$P(C|\boldsymbol{f}) = P(C) \times \frac{\prod_{i=1}^{n} P(f_i|C)}{P(\boldsymbol{f})} \tag{4}$$

With this equation, we can calculate $P(C|\boldsymbol{f})$ to classify $\boldsymbol{f}$ into the class (match/non-match) with the highest $P(C|\boldsymbol{f})$. This approach is one of the baseline methods we compare our model to.

Like the probabilistic record linkage, one of the often-admitted weaknesses of this approach is that it depends upon the assumption that each of its fields is independent from the others. The tree augmented naive Bayes classier (TAN) and its improved version ETAN relax this assumption by allowing interactions between feature fields.

6.2 The Tree Augmented Naive Bayes Classifier

TAN [20] can be seen as an extension of the naive Bayes classifier by allowing a feature as a parent (Figure 3(c)). In NBC, the network structure is naive, where each feature has the class as the only parent. In TAN, the dependencies between features are learnt from the data. Given a complete data set $D = \{D_1, ..., D_L\}$ with L labelled instances, where each instance is an instantiation of all the variables. Conventional score-based algorithms for structure learning make use of certain heuristics to find the optimal DAG that best describes the observed data D over the entire space. We define:

$$\hat{G} = \arg\max_{G \in \Omega} \ell(G, D) \tag{5}$$

where $\ell(G, D)$ is the *log-likelihood* score, which is the logarithm of the likelihood function of the data that measures the fitness of a DAG G to the data D. Ω is the set of all DAGs scoring candidate structures based on the data.

Assume that the score (i.e. BDeu score [21]) is decomposable and respects likelihood equivalence, we can devise an efficient structure learning algorithm for TAN. Because every feature f_i has C as a parent, the structure (f_i has f_j and C as parents, $i \neq j$) has the same score with the structure, where f_j has f_i and C as parents:

$$\ell(f_i, \{f_j, C\}, D) + \ell(f_j, C, D) = \ell(f_j, \{f_i, C\}, D) + \ell(f_i, C, D) \tag{6}$$

In addition to the naive Bayes structure, in TAN, features are only allowed to have at most one other feature as a parent. Thus, we have a tree structure between the features. Based on the symmetry property (equation (6)), there is an efficient algorithm to find the optimal TAN structure by converting the original

problem (equation (5)) into a minimum spanning tree construction. More details can be found in [11].

6.3 The Extended TAN Classifier

As discussed in the previous section, TAN encodes a tree structure over all the features. And it has been shown to outperform naive Bayes classifier in a range of experiments [20]. However, when the training data are scarce or a feature and the class are conditionally independent given another feature, a TAN structure may not be best. Therefore, people have proposed the Extended TAN (ETAN) classifier [11, 12] to allow more structure flexibility.

ETAN is a generalization of TAN and NBC. It does not force a tree to cover all the attributes, and a feature to connect with the class. As shown in Figure 3(d), ETAN could disconnect a feature if such a feature is not important to predict C. Thus, ETAN's search space of structures includes that of TAN and NBC, and we have:

$$\ell(\hat{G}_{ETAN}, D) \geq \ell(\hat{G}_{TAN}, D) \quad and \quad \ell(\hat{G}_{TAN}, D) \geq \ell(\hat{G}_{NBC}, D) \tag{7}$$

which means the score of the optimal ETAN structure is superior or equal to that of the optimal TAN and NBC (*Lemma* 2 in [11]).

In ETAN, the symmetry property (equation (6)) does not hold, because a feature (e.g. f_2 in Figure 3(d)) is allowed to be disconnected from the class. Thus, the undirected version of the minimum spanning tree algorithm cannot be directly applied here. Based on Edmonds' algorithm for finding minimum spanning trees in directed graphs, the structure learning algorithm of ETAN was developed, which has a computational complexity that is quadratic in the number of features (as is TAN). For detailed discussions we direct the reader to the papers [11, 12].

6.4 Hierarchical Restrictions Between Features

To utilize the benefits of existing domain knowledge, we extend the NBC method by allowing hierarchical restrictions between features (HR-NBC). These restrictions are modelled as dependencies between features in HR-NBC.

Hierarchy restrictions between features commonly occur in *real world* problems. For example, Table 1 shows four address records, which refer to two restaurants (there are two duplicates). The correct linkage for these four records is: record 1 and 2 refer to one restaurant in Southwark, and record 3 and 4 refer to another restaurant in Blackheath. As we can see, even record 1 and 3 exactly match with each other in the field of restaurant name, they cannot be linked with each other because they are located in a different localities.

Based on the description of the example given in Table 1, we can see there is a hierarchical restriction between the *name* and *locality* fields, where the *locality* field has a higher feature level than the *name* field. Thus, intuitively, it is recommended to compare the *locality* field first to filter record linkage pairs. To let

Table 1. Four restaurant records with name, address, locality and type information.

Index	Name (f_1)	Address (f_2)	Locality (f_3)	Type (f_4)
1	Strada	Unit 6, RFH Belvedere Rd	Southwark	Roman
2	Strada at Belvedere	Royal Festival Hall	Southwark	Italian
3	Strada	5 Lee Rd	Blackheath	Italian
4	Strada at BH	5 Lee Road	BLACKHEATH	Italian

our classifier capture such hierarchical restriction, we introduce a dependency between these two fields ($f_3 \rightarrow f_1$) to form our HR-NBC model (Figure 3(b)). Thus, equation (4) now becomes:

$$P(C|\boldsymbol{f}) = P(C) \times \frac{P(f_1|f_3, C)\prod_{i=2}^{n} P(f_i|C)}{P(\boldsymbol{f})} \tag{8}$$

Parameter estimation Let θ denote the parameters that need to be learned in the classifier and let r be a set of fully observable record-pairs. The classical Maximum Likelihood Estimation (MLE) finds the set of parameters that maximize the data *log-likelihood* $\ell(\theta|r) = \log P(r|\theta)$.

However, for several cases in the unified model, a certain parent-child state combination would seldom appear, and the MLE learning fails in this situation. Hence, Maximum a Posteriori (MAP) algorithm is used to mediate this problem via the *Dirichlet* prior: $\hat{\theta} = \arg\max_\theta \log P(r|\theta)P(\theta)$. Because there is no informative prior, in this work we use the BDeu prior [21] with equivalent sample size (ESS) equal to 1.

7 Experiments

This section compares PRL-W to different Bayesian network classifiers. The goal of the experiments is to do an empirical comparison of the different methods, and show the advantages/disadvantages of using them in different settings. Also, it is of interest to investigate how such hierarchical feature level information and parsed addresses could improve the classifier's performance.

7.1 Settings

Our experiments are performed on four different datasets[2], two synthetic datasets [4] (*Country* and *Company*) with sampled spelling errors and two real datasets (*Restaurant* and *Tungsten*). The *Country* and *Company* datasets contain 9 and 11 fields respectively. All the field similarities are calculated by the Jaro-Winkler similarity function.

[2] These datasets can be found at `http://yzhou.github.io/`.

Restaurant is a standard dataset for record linkage study [10]. It was created by merging the information of some restaurants from two websites. In this dataset, each record contains 5 fields: name, address, city, phone and restaurant-type [3].

Tungsten is a commercial dataset from an e-invoicing company named Tungsten Corporation. In this dataset, there are 2744 duplicates introduced by user entry errors. Each record contains 5 fields: company name, country code, address line 1, address line 4 and address line 6.

The details of these 4 datasets and statistical results are summarized in Table 2.

Table 2. The details of the experimental datasets.

Dataset	Number of fields	Number of instances	Null value percentages
Country	9	520	31.8%
Company	11	4000	16.7%
Restaurant	4	2176	0.0%
Tungsten	5	1238	27.1%

The experimental platform is based on the Weka system [22]. Since TAN and ETAN can not deal with continuous field similarity values, these values are discretised with the same routine as described in PRL-W. To simulate *real world* situations, we use an affordable number (10, 50 and 100) of labelled records as our training data. The reason is clear that it would be very expensive to manually label hundreds of records. The experiments are repeated 100 times in each setting, and the results are reported with the mean.

To evaluate the performance of different methods, we compare their ability to reduce the number of *false decisions*. False decisions include **false matches** (the record-pair classified as a match for two different records) and **false non-matches** (the record-pair classified as a non-match for two records that are originally same). Thus these methods are expected to get high *precision* and *recall*, where *precision* is the number of correct matches divided by the number of all classified matches, and *recall* is the number of correct matches divided by the number of all original matches.

To consider both the *precision* and *recall* of the test, in this experiment, we use F_1 score as our evaluation criteria. This score reaches its best value at 1 and worst at 0, and is computed as follows:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{9}$$

[3] Because the phone number is unique for each restaurant, it, on its own, can be used to identify duplicates without the need to resort to probabilistic record linkage techniques. Thus, this field is not used in our experiments.

7.2 Results

The F_1 score of all five methods in different scenarios are shown in Table 3, where the highest average score in each setting is marked in bold. Results of competitors to the best score are marked with an asterisk * where there is a statistically significant difference ($p = 0.05$).

Table 3. The F_1 score of five record linkage methods in different datasets.

Dataset	L	PRL-W	TAN	ETAN	NBC	HR-NBC
Country	10	**0.974**	0.920*	0.899*	0.938*	0.941*
	50	0.971*	0.970*	0.967*	**0.976**	**0.976**
	100	0.967*	0.977*	0.978	0.980	**0.981**
Company	10	**0.999**	0.969*	0.965*	0.987*	0.988*
	50	**0.999**	0.995*	0.992*	0.997*	0.997*
	100	**0.999**	0.997*	0.996*	0.998	**0.999**
Restaurant	10	**0.996**	0.874*	0.863*	0.884*	0.897*
	50	**0.996**	0.950*	0.952*	0.957*	0.958*
	100	**0.995**	0.957*	0.958*	0.959*	0.960*
Tungsten	10	**0.990**	0.919*	0.908*	0.916*	0.916*
	50	**0.990**	0.970*	0.967*	0.972*	0.972*
	100	**0.990**	0.970*	0.969*	0.972*	0.972*
Average	N/A	**0.989**	0.956*	0.951*	0.961*	0.963*

As we can see, the PRL-W gets the best result in *Company*, *Restaurant* and *Tungsten* datasets. And its performance does not depends on the number of labelled training record-pairs. The reason is that the record linkage weights were computed with an EM-algorithm as described in equation (1) and (2) over the whole dataset (labelled and unlabelled data). As we can see from Table 2, all these three datasets have more than 1000 record-pairs. When two classes are easy to distinguish, it is not surprising that the PRL-W can attain good performance with limited labelled data.

Because of the scarcity of labelled data and the large number of features, TAN and the state-of-the-art ETAN methods have a relatively bad performance in all these four datasets. The average F_1 score of TAN and ETAN are 0.956 and 0.951, which are both smaller than the scores of NBC (0.961) and HR-NBC (0.963). In addition, although it is proven that ETAN provides a better fit to the data (equation (7)) than TAN, it receives lower classification accuracies in these settings, presumably, due to overfitting.

According to the results, both NBC and HR-NBC have high F_1 score in all settings. This demonstrates the benefits of using these two methods when

labelled data is scarce. Moreover, the performance of our HR-NBC[4] is equal to or superior to that of NBC in all these cases.

Introducing address parsing of the data

As discussed in the framework (Figure 1), unstructured address fields could be further parsed to improve training data quality. In our experiments, both *Restaurant* and *Tungsten* datasets contain such address field. Specifically, by using the HMM parser discussed in Section 4, original fields "address" of *Restaurant* and "address line 1" of *Tungsten* are further parsed into 3 fields: house number, street name and street type.

Because original address fields are further parsed, hierarchical restrictions are not introduced in the experiment. Therefore, we only discuss the performance of PRL-W, TAN, ETAN and NBC. The results of different methods on the parsed datasets are shown in Table 4. Compared with the results in Table 3, symbols – and ↑ in Table 4 represent unchanged and improved performance respectively. Moreover, values after ↑ indicate the specific increase in F_1 score of the various methods on these parsed datasets.

Table 4. The F_1 score of PRL-W, TAN, ETAN and NBC with parsed addresses.

Dataset	L	PRL-W	TAN	ETAN	NBC
Restaurant	10	**0.996** (–)	0.950* (↑0.076)	0.956* (↑0.093)	0.975* (↑0.091)
	50	**0.996** (–)	0.982* (↑0.032)	0.987* (↑0.035)	0.992* (↑0.035)
	100	**0.996** (↑0.001)	0.989* (↑0.032)	0.990* (↑0.032)	0.993* (↑0.034)
Tungsten	10	**1.000** (↑0.010)	0.982* (↑0.063)	0.977* (↑0.069)	0.987* (↑0.071)
	50	**1.000** (↑0.010)	0.995* (↑0.025)	0.992* (↑0.025)	0.996* (↑0.024)
	100	**1.000** (↑0.010)	0.996* (↑0.026)	0.994* (↑0.025)	0.997* (↑0.025)
Average	N/A	**0.998** (↑0.005)	0.982* (↑0.042)	0.983* (↑0.047)	0.990* (↑0.047)

As can be seen from the results of Table 4, the performance of all 4 methods is improved by introducing parsed addresses. Specifically, comparing to the results in Table 3, the average increases in F_1 score in Table 4 are 0.005, 0.042, 0.047 and 0.047 for PRL-W, TAN, ETAN and NBC respectively.

8 Conclusions

In this paper, we discussed hierarchical restrictions between features, and exploited the classification performance of different methods for record linkage on both synthetic and real datasets. Moreover, we showed an improved performance of the methods considered on further parsed datasets (Table 4).

[4] In each dataset, we only introduce one hierarchical restriction between the *name* and *address* field.

The results demonstrate that, in settings of limited training data, PRL-W works well and its performance is independent of the number of labelled record-pairs, TAN, NBC and HR-NBC have better performance than ETAN even though the latter method provides a theoretically better fit to the data. Compared with NBC, HR-NBC achieves equal or superior performance in experiments of Table 3 with an aptly chosen hierarchical restriction, which show the benefits of this in these datasets.

We note, however, that our method might not be preferable in all cases. For example, in a medical dataset, a patient could move her or his address and have multiple records. In this case, two records with different addresses refer to the same person. Thus, the hierarchical restrictions used in this paper will introduce extra false non-matches.

In future work we will investigate other sources of domain knowledge to enhance the performance of the resultant classifier, such as improving accuracy by using specific parameter constraints [23] and transferred knowledge [24].

Bibliography

[1] Dunn, H.L.: Record linkage*. American Journal of Public Health and the Nations Health **36**(12) (1946) 1412–1416

[2] Tromp, M., Ravelli, A.C., Bonsel, G.J., Hasman, A., Reitsma, J.B.: Results from simulated data sets: probabilistic record linkage outperforms deterministic record linkage. Journal of Clinical Epidemiology **64**(5) (2011) 565–572

[3] Ananthakrishna, R., Chaudhuri, S., Ganti, V.: Eliminating fuzzy duplicates in data warehouses. In: Proceedings of the 28th international conference on Very Large Data Bases, VLDB Endowment (2002) 586–597

[4] Leitão, L., Calado, P., Herschel, M.: Efficient and effective duplicate detection in hierarchical data. IEEE Transactions on Knowledge and Data Engineering **25**(5) (2013) 1028–1041

[5] Fellegi, I.P., Sunter, A.B.: A theory for record linkage. Journal of the American Statistical Association **64**(328) (1969) 1183–1210

[6] Winkler, W.E.: The state of record linkage and current research problems. In: Statistical Research Division, US Census Bureau, Citeseer (1999)

[7] Winkler, W.E.: String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage. In: Proceedings of the Section on Survey Research. (1990) 354–359

[8] Li, X., Guttmann, A., Cipiere, S., Maigne, L., Demongeot, J., Boire, J.Y., Ouchchane, L.: Implementation of an extended Fellegi-Sunter probabilistic record linkage method using the Jaro-Winkler string comparator. In: 2014 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), IEEE (2014) 375–379

[9] Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate record detection: A survey. IEEE Transactions on Knowledge and Data Engineering **19**(1) (2007) 1–16

[10] Ravikumar, P., Cohen, W.W.: A hierarchical graphical model for record linkage. In: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, AUAI Press (2004) 454–461

[11] de Campos, C.P., Cuccu, M., Corani, G., Zaffalon, M.: Extended tree augmented naive classifier. In: Probabilistic Graphical Models. Springer (2014) 176–189

[12] de Campos, C.P., Corani, G., Scanagatta, M., Cuccu, M., Zaffalon, M.: Learning extended tree augmented naive structures. International Journal of Approximate Reasoning (2015)

[13] Zhou, Y., Howroyd, J., Danicic, S., Bishop, J.: Extending naive bayes classifier with hierarchy feature level information for record linkage. In Suzuki, J., Ueno, M., eds.: Advanced Methodologies for Bayesian Networks. Volume 9505 of Lecture Notes in Computer Science. Springer International Publishing (2015) 93–104

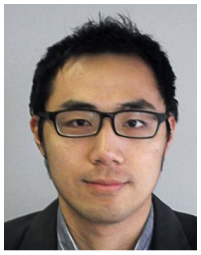
[14] Köpcke, H., Rahm, E.: Frameworks for entity matching: A comparison. Data & Knowledge Engineering **69**(2) (2010) 197 – 210

[15] Leitão, L., Calado, P., Weis, M.: Structure-based inference of XML similarity for fuzzy duplicate detection. In: Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management. CIKM '07, New York, NY, USA, ACM (2007) 293–302

[16] Rahm, E., Do, H.H.: Data cleaning: Problems and current approaches. IEEE Data Engineering Bulletin **23**(4) (2000)

[17] Churches, T., Christen, P., Lim, K., Zhu, J.X.: Preparation of name and address data for record linkage using hidden Markov models. BMC Medical Informatics and Decision Making **2**(1) (2002) 1
[18] Christen, P., Belacic, D.: Automated probabilistic address standardisation and verification. In: Australasian Data Mining Conference (AusDM05). (2005) 53–67
[19] Jaro, M.A.: Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. Journal of the American Statistical Association **84**(406) (1989) 414–420
[20] Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine learning **29**(2-3) (1997) 131–163
[21] Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning **20**(3) (1995) 197–243
[22] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD explorations newsletter **11**(1) (2009) 10–18
[23] Zhou, Y., Fenton, N., Neil, M.: Bayesian network approach to multinomial parameter learning using data and expert judgments. International Journal of Approximate Reasoning **55**(5) (2014) 1252–1268
[24] Zhou, Y., Fenton, N., Hospedales, T., Neil, M.: Probabilistic graphical models parameter learning with transferred prior and constraints. In: Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence, AUAI Press (2015) 972–981

Yun Zhou is a researcher at Tungsten Centre for Intelligent Data Analytics, Goldsmiths, University of London. He received his B.Sc. (Information Systems Engineering) and M.Sc. degrees (Management Science and Engineering) from National University of Defense Technology and his Ph.D. degree in Computer Science from Queen Mary University of London. His research interests include record linkage, machine learning and Bayesian network.



Minlue Wang is a researcher at Tungsten Centre for Intelligent Data Analytics, Goldsmiths, University of London. He received his B.Sc. and Ph.D. in Computer Science from the University of Birmingham. His research interests include planning under uncertainty, robotics, structural classification, HMM, and computational linguistic.

Valeriia Haberland is a researcher at the Tungsten Centre for Intelligent Data Analytics, Goldsmiths, University of London. She received her B.Sc. and M.Sc. degrees in Computer Systems and Networks (with distinction), Zaporizhzhya National Technical University, Ukraine; M.Sc. degree in Information Technology (with distinction), Saint Petersburg State University, Russian Federation; Ph.D. degree in Computer Science, King's College London, United Kingdom. Her current research interests include data analytics, enrichment and provenance.

John Howroyd studied Mathematics at Oxford University and University College London. As well a being an established mathematician John has published widely in computer science; in particular, in program analysis. He has also worked as Head of Research in a major project developing a Spend-Analytics system for NHS trusts. He has in-depth knowledge of Bayesian Networks, classication and clustering methods and is also an experienced database engineer specialising in efficiency and data representation.

Sebastian Danicic is the director of research at the Tungsten Centre for Intelligent Data Analytics. His research encompasses a range of different areas including program slicing, dependence analysis and transformation, program schema theory, evolutionary mutation testing, and, more recently, intelligent web spidering, Java decompilation, and Intelligent Data Analytics software watermarking and Community Detection in Software.

Mark Bishop studied Cybernetics and Computer Science at the University of Reading. He is Professor of Cognitive Computing at Gold-smiths, University of London and between 2010-2014 was Chair of the society for the study of Artificial Intelligence and the Simulation of Behaviour (AISB), the oldest Artificial Intelligence Society in the world. He has published widely in areas of Artificial Intelligence, Machine Learning and Neural Computing.