Improving Temporal Record Linkage Using Regression Classification

Yichen Hu(✉), Qing Wang, Dinusha Vatsalan, and Peter Christen

Research School of Computer Science, The Australian National University, Canberra, ACT 0200, Australia
{yichen.hu,qing.wang,dinusha.vatsalan,peter.christen}@anu.edu.au

Abstract. Temporal record linkage is the process of identifying groups of records that are collected over a period of time, such as in census or voter registration databases, where records in the same group represent the same real-world entity. Such databases often contain temporal information, such as the time when a record was created or when it was modified. Unlike traditional record linkage, which considers differences between records from the same entity as errors or variations, temporal record linkage aims to capture records from entities where the attribute values are known to change over time. In this paper we propose a novel approach that extends an existing temporal approach called *decay model*, to categorically calculate probabilities of change for each attribute. Our novel method uses a regression-based machine learning model to predict decays for sets of attributes. Each such set of attributes has a principle attribute and support attributes, where values of the support attributes can affect the decay of the principle attribute. Our experimental results on a real US voter database show that our proposed approach results in better linkage quality compared to the decay model approach.

Keywords: Data matching · Temporal data · Decay · Attribute weighting · Entity resolution

1 Introduction

Record linkage (also known as data matching, entity resolution, and duplicate detection) identifies records that refer to the same real-world entity [5]. Record linkage is being used in many application domains, such as linking patient data for disease outbreak detection or clinical trails in the health industry [5], credit checking and fraud detection in the finance industry [6], and constructing population databases for social science research [11]. Challenges in record linkage are caused by the lack of unique identifiers (such as national identifier numbers), dirty data (such as misspellings and missing values), legitimate updates over time (such as changes in last name or address), and the lack of informative attributes (i.e. a dataset might not contain gender and/or date of birth).

This work was partially funded by the Australian Research Council under Discovery Project DP160101934.

J. Kim et al. (Eds.): PAKDD 2017, Part I, LNAI 10234, pp. 561–573, 2017.
DOI: 10.1007/978-3-319-57454-7_44

Record linkage generally involves the following steps [5]: data preprocessing, blocking, comparison and classification, and evaluation. This paper focuses on record pair comparison and classification, especially the task of calculating a similarity value with greater effectiveness at distinguishing between matches and non-matches for temporal data compared to previous temporal and non-temporal linkage techniques. While record linkage has been studied for several decades, until recently most works in this field did not use any temporal information available in datasets [13]. However, records of the same entity can be collected over a long period of time (years or even decades), such as census data that in many countries are collected every five or ten years. During such periods, certain attribute values of an entity are likely to change, such as a person's job position, living address, and potentially their last name (if somebody gets married).

Traditional record linkage methods assume that highly similar records are most likely to belong to the same entity [5]. These techniques do not perform well on temporal data, because entities might change some of their attribute values over time. For example, when a person changes his or her last name or address, their new record is not linked to earlier records because the attribute values do not match, or their earlier records are linked by mistake to records of a different person who has the same last name and/or address [6].

Temporal record linkage aims to address the above issues by using temporal information, such as the time-stamp when a record was created or modified. These time-stamps can be used to sort records by time and calculate temporal distances between records. They therefore provide opportunities for new record linkage approaches (examples will be discussed in Sect. 2). A dataset needs to contain temporal information for each record to be used in temporal record linkage, such as the date when being entered (for medical records), the date when being published (for publication records), or the date when being collected (for datasets collected by taking snapshots of databases at different points in time).

Table 1. An example of a temporal datasets.

RecID	EntID	FName	MName	LName	Address	Sex	Age	EntryDate
r1	e2	Elsa		Clark	161 Castlereagh St, Sydney	F	24	2011-09-11
r2	e1	Ella	Rose	Taylor	456 Kent Street, Sydney	F	23	2011-10-12
r3	e1	Ella	Louise	Taylor	456 Kent Street, Sydney	F	23	2012-02-20
r4	e1	Ella	Louise	Clark	299 Elizabeth St, Sydney	F	24	2012-06-30
r5	e2	Elsa		Taylor	201 Kent Street, Sydney	F	26	2013-08-05

Example: Given five records of two entities as in Table 1, if we do not consider the temporal information *EntryDate*, records r2 and r5 will have a high similarity and will therefore be matched incorrectly, whereas records r2 and r4 will have a low similarity and this true match will be missed. Temporal record linkage aims to correctly link r1 to r5, and link r2, r3 and r4 together using the temporal information in the *EntryDate* attribute.

This paper extends an existing temporal linkage approach called *decay model* [13], which learns the probability for an attribute to change over time (*disagreement decay*) and the probability for an attribute to share the same value among different entities over time (*agreement decay*). It then uses these decays to adjust the weight given to each attribute, where the sum of adjusted attribute weights is used to calculate the similarity between a pair of records and decide if they are a match or non-match based on a similarity threshold [5]. The *decay model* assumes the probability for an attribute to change its value over a certain time period is the same for every entity, and this assumption is not always true. For example, young people are more likely to change their address than seniors, and young females are more likely to change their last name than senior males.

Contributions: We integrate a linear regression model into the *decay model* [13]. Our model uses support attributes to calculate the decay of a principle attribute whose decay is affected by the values of those support attributes. The calculated decays are therefore more specific to each entity. For example, a person's gender can affect the likelihood of changes in their last name, and when we calculate a decay for last name with gender as a support attribute we can learn a gender sensitive decay model for last name. Our intuition is that the probability for an attribute to change over time can be predicted more accurately with the help of other attributes upon which it depends. We also propose a method to adjust the impact of decay models, and evaluate our approach on four subsets of a real US voter dataset. The experimental results show that our approach improves the linkage quality compared to two baseline approaches.

2 Related Work

We discuss related work in the two areas of record linkage that are non-temporal and temporal models. The common objective of both types of models is to decide if a pair of records is a match, or if a record belongs to a cluster of records where all records refer to the same entity. Temporal models consider temporal information in addition to attribute similarities as used in non-temporal models.

Fellegi and Sunter [7] proposed a statistical non-temporal linkage model. This model weights each attribute according to two types of probabilities: (1) the probability for a pair of records that agree on an attribute to be a match; and (2) the probability for a pair of records that agree on an attribute to be a non-match. The weights of attributes are summed to calculate the matching score for each pair of records. These probabilities can be learned from training data or estimated using the Estimation-Maximization algorithm [9].

Li et al. [13] were the first to propose a temporal model which considers the probability for an attribute's value to change over time, where this probability is learned from training data. The model calculates a disagreement decay (the likelihood for an attribute to change within a certain time period) and an agreement decay (the likelihood for an entity's attribute value to be the same as another entity's within a time period). The two types of decays are used to adjust the weight of each attribute as used in the similarity calculations.

More recently, Li et al. [12] proposed a temporal model which learns the probability for each attribute value to change to some commonly occurring value over time. However, this approach requires a temporal dataset to have attributes whose values change to some commonly occurring values, such as job positions (for example, the position 'technician' can change to 'manager').

Christen and Gayler [6] modified the approach proposed by Li et al. [13] to iteratively train a temporal model using a stream of time-stamped records. Every time a certain number of records are matched, the approach uses the matching results to retrain the temporal model. The difference between this approach and the original temporal model [13] is that the latter only learns the temporal model from training data once, whereas the former continuously trains the temporal model using linkage results produced by itself.

Chiang et al. [3] proposed an algorithm which learns the probability for an attribute's value to recur within different time periods. For each value of an attribute, the algorithm constructs a transition history and uses this history to calculate the probability for a value to recur. These probabilities are used to adjust the original similarity of a pair of records.

All these existing works do not address dependencies between attributes when calculating the probability for an attribute to change over time. Although Li et al. [13] introduced a decay model, their model only calculates the probability for attribute values to change independently. We believe the decay model can be improved and made more effective to improve the linkage quality by considering dependencies between attributes.

3 Problem Statement

We now define the notation as well as the problem we aim to tackle in this paper. Let $\mathbf{R}$ be a set of records and $\mathbf{E}$ be a set of entities. Each record $r \in \mathbf{R}$ has a list of attribute values $[a_1, a_2, \ldots, a_k]$ and a time-stamp $r.t$, where each value a_i $(1 \leq i \leq k)$ is associated with an attribute A, and we use $r.A$ to denote the value a_i of A in r. Every record $r \in \mathbf{R}$ must belong to exactly one entity $e \in \mathbf{E}$. The entity to which a record r belongs to is denoted as $e(r)$.

Attribute values of an entity e can change over time, where each change (update) is represented by a new record r_i with a time-stamp $r_i.t$ and attribute value(s) that is/are different from the previous record. For example, let r_1, r_2 be two records belonging to e (in another word, $e(r_1) = e(r_2)$). If $r_1.t < r_2.t$ and $\exists A \in \mathbf{A} : r_1.A \neq r_2.A$, then we say that the value of attribute A of entity e has changed between the two time-stamps $r_1.t$ and $r_2.t$.

Given a training dataset $\mathbf{C}$ in the form of a set of clusters of records. Each cluster $C \in \mathbf{C}$ contains a set of records $\{r_1, r_2, \ldots\}$. All records in a cluster C represent the same entity, and records in different clusters represent different entities.

The temporal record linkage problem is to link all $r_a, r_b \in \mathbf{R}$, where $e(r_a) = e(r_b)$, $\exists A \in \mathbf{A} : r_a.A \neq r_b.A$. Note that it is possible to have a pair of records where $e(r_a) = e(r_b)$ and $\forall A \in \mathbf{A} : r_a.A = r_b.A$, which means no temporal update has occurred between the two records. In this case we will only keep the oldest record of the pair during data preprocessing.

The goal of our work is to address the temporal record linkage problem using a weighting strategy which adjusts the importance of attributes in order to improve the quality of linkage. Our work uses a regression model to train and predict parameters for a temporal model. Our solution is based on the assumption that adjusting the weights of each attribute A according to its probability to change over time can improve the quality of record linkage.

4 Temporal Record Linkage Framework

In this section we discuss the temporal record linkage framework used in our work. *Swoosh* is a generic record linkage method which compares records according to features (sets of attributes) selected by the user [1]. A pair of records is merged into a new record when one of their features meets the matching criteria provided by the user, and then the two original records are removed. Swoosh treats the classifier, which decides whether a pair of records is a match, as a blackbox. In this paper, we use a threshold-based classifier that classifies a pair of records as a match when its similarity is greater than a user defined similarity threshold. The objective of our proposed approach is to calculate a similarity value for a pair of records using temporal information.

The *decay model* calculates the similarity of a pair of records using the similarity of each pair of attribute values that is adjusted by weights. The weight of each attribute is calculated according to its disagreement and agreement decays [13]. Disagreement decay is the probability for an attribute to change its value within a time period, and agreement decay is the probability for multiple entities to have the same attribute value within a time period [13]. A time distance Δt refers to the difference between two time-stamps, and is measured by a time unit defined by the user, such as days, years, or hours.

A *life span* l refers to the time distance of an attribute value to be used by an entity. An attribute's life span is *full* when the value has a date when it was used first and another date when it was changed to another value. The time distance between the first and second date is a *full life span*, denoted as l_f. Similarly, if the attribute's value does not change between two time-stamps, the time distance between the time-stamps is a *partial life span*, denoted as l_p.

For example, assume that an entity has three different last names over five records, with time-stamps in the form of (year-month): 'Taylor' (2011-10) $\rightarrow$ 'Taylor' (2011-12) $\rightarrow$ 'Spire' (2012-12) $\rightarrow$ 'Spire' (2013-10) $\rightarrow$ 'Wright' (2015-10). The time distance between the first and the third records is one full life span with a length of 14 months (2011-10 to 2012-12), and the distance between the third and the fifth records is another full life span with a length of 34 months (2012-12 to 2015-10). Note that, in this example, month is being used as a time unit but this is not necessary for all datasets. From this example, the time distance between the first and the second records is a partial life span with a length of 2 months (the time distance between 2011-10 and 2011-12), and the time distance between the third and the fourth records (2012-12 and 2013-10) is another partial life span with a length of 10 months.

Let $\bar{L}_f$ denote the list of all full life spans l_f of an attribute A for all entities and let $\bar{L}_p$ denote the list of all partial life spans l_p of an attribute A for all entities. Then the disagreement decay is formally defined as below.

Definition 1. *(Disagreement decay $d^{\neq}$)* [13]*: Let Δt be a time distance, $A \in \boldsymbol{A}$ be an attribute. The disagreement decay of A over Δt is the probability $d^{\neq}(A, \Delta t)$ that an entity changes its value of A within Δt:*

$$d^{\neq}(A, \Delta t) = (|\{l \in \bar{L}_f | l \leq \Delta t\}|)/(|\bar{L}_f| + |\{l \in \bar{L}_p | l \geq \Delta t\}|) \tag{1}$$

Let $\bar{L}$ denote a list of both full and partial life spans of an attribute A for all entities. For each record, if it has the same attribute value with another record which belongs to a different entity, the time distance between the two records is added to $\bar{L}$. If no entity has the same attribute value, a life span with length ∞ is added to $\bar{L}$. Then the agreement decay is formally defined as below.

Definition 2. *(Agreement decay $d^{=}$)* [13]*: Let Δt be a time distance, $A \in \boldsymbol{A}$ be an attribute. The agreement decay of A over Δt is the probability $d^{=}(A, \Delta t)$ that two different entities share the same value of A within Δt:*

$$d^{=}(A, \Delta t) = (|\{l \in \bar{L} | l \leq \Delta t\}|)/(|\bar{L}|) \tag{2}$$

The *decay model* uses the agreement and disagreement decay to calculate w_A (weight of attribute A), as shown in (3). The comparison function s_A calculates the similarity between a pair of attribute values. s_A is defined by the user and it returns a similarity value in the range $[0, 1]$. These comparison functions can be approximate string similarity functions, such as edit-distance or Jaro-Winkler [5].

$$w_A(s_A, \Delta t) = 1 - s_A \cdot d^{=}(A, \Delta t) - (1 - s_A) \cdot d^{\neq}(A, \Delta t) \tag{3}$$

Weights are used to calculate the pair-wise similarity between two records, as shown in (4). s_r denotes the decay adjusted similarity between two records r_a and r_b. s_r is the final similarity score that is used to classify a pair of records, which decides if it is a match or non-match. s_r is in the range $[0, 1]$.

$$s_r(r_a, r_b) = \frac{\sum_{A \in \mathbf{A}} w_A(s_A(r_a.A, r_b.A), |r_a.t - r_b.t|) \cdot s_A(r_a.A, r_b.A)}{\sum_{A \in \mathbf{A}} w_A(s_A(r_a.A, r_b.A), |r_a.t - r_b.t|)} \tag{4}$$

5 Improved Decay Model

In this section we introduce an improved temporal model based on the decay model [13] described above.

5.1 Predicting Probability with a Regression Model

From the previous equations we can see that the agreement and disagreement decays are calculated using only a single attribute. For example, when the disagreement decay of attribute *last name* is calculated, the temporal model calculates the overall probability for an entity to change its last name within a given time distance Δt. However, the probability for an entity to change its last name is often associated with gender and age. The disagreement decay for last name, calculated without considering the gender and age values of an entity, would be too high for older males, and too low for younger females, because it is rare for an older man to change his last name, but more common for a young woman to change her last name when she gets married.

A set of *support attributes* is selected for *principle attributes* where the value of a support attribute may affect the probability for their attribute value to change. For example, when predicting the probability for values in attribute *address* to change, attributes *gender* and *age* can be used as support attributes to make the prediction more accurate. Support attributes are selected by the user based on their domain knowledge for each principle attribute, and each principle attribute can have zero to many support attributes. They can also be selected using a feature selection strategy that is able to explore the dependency between features [2].

To create a training dataset for each attribute, our algorithm iterates through the records of each entity. The algorithm checks if an entity has changed its principle attribute value within a time distance. The time distance ranges from 1 to the maximum time distance of the whole dataset. For each time distance, a training instance is created using the time distance and values of the support attributes as features, and using the status of value change (changed or unchanged) as class value. The training dataset is then used to train a regression model.

In this paper, we use a linear regression model, as commonly used in parameter estimation and prediction [10], to predict disagreement probability.

Disagreement Probability: We introduce a concept called *disagreement probability* $d^{\neq}_{prob}$, which has a similar definition as disagreement decay (as shown in (1)), but is modified in order to be used with a regression model. From (5), we can see that the difference between $d^{\neq}_{prob}$ and $d^{\neq}$ is that the divisor of $d^{\neq}_{prob}$ is fixed for each entity. With a fixed divisor, we can create training instances for a regression model according to if $l \leq \Delta t$. When a full life span $l \in \bar{L}_f$ is encountered, we can decide if it is lower than a certain Δt and create a training record. These training records are used to train the regression model to predict disagreement probability. For each Δt, a training record is created using: (1) the value of each support attribute of A whose model is being built; (2) the current Δt; and (3) a class value which is equal to 1 if $l \leq \Delta t$, or 0 if $l > \Delta t$ or $l \in \bar{L}_p$.

When a class value of a training record equals to 1, it means the value of A of an entity has been changed within Δt and the life span is full, whereas a class value 0 means the value of A has not been changed within Δt and the life span is partial. It needs to be noted that (5) is only relevant when we create training records. The equation provides a conceptual insight about why we create training records following the steps described above. The $d^{\neq}_{prob}$ that is being used after the training stage is predicted using the trained regression model, rather than calculated using (5).

$$d^{\neq}_{prob}(A, \Delta t) = (|\{l \in \bar{L}_f | l \leq \Delta t\}|)/(|\bar{L}_f| + |\bar{L}_p|) \tag{5}$$

$d^{\neq}_{prob}$ is normalized into the range $[0, 1]$, and then it can be used as a weight to adjust attribute-wise similarities, as shown in (6). s_p denotes the similarity between a pair of records adjusted using $d^{\neq}_{prob}$.

$$s_p(r_a, r_b) = \sum_{A \in \mathbf{A}} \frac{1 - d^{\neq}_{prob}(A, |r_a.t - r_b.t|)}{\sum_{A' \in \mathbf{A}} 1 - d^{\neq}_{prob}(A', |r_a.t - r_b.t|)} \cdot s_A(r_a.A, r_b.A) \tag{6}$$

Combining Disagreement Probability with Agreement Decay: Disagreement probability can be normalized as: $d^{\neq}_{nprob}(A, \Delta t) = (d^{\neq}_{prob}(A, \Delta t))/(max(d^{\neq}_{prob}(A)))$, where $max(d^{\neq}_{prob}(A))$ is the maximum disagreement probability over all Δt. Using (3) and (4) above, with $d^{\neq}$ being replaced by $d^{\neq}_{nprob}$, a different w_A can be calculated, while (7) shows how to calculate w_A using $d^{\neq}_{nprob}$.

$$w_A(s_A, \Delta t) = 1 - s_A \cdot d^{=}(A, \Delta t) - (1 - s_A) \cdot d^{\neq}_{nprob}(A, \Delta t) \tag{7}$$

5.2 Adjusting the Impact of Decay Models

The intuition of using decays to adjust attribute weights is that attributes that have higher probability to change their values are less reliable than those that change less often. However, the normalized probabilities of changing values may not immediately represent the optimal weighting of attributes. For example, let *last name* have a probability to change as 10% over 3 years, and *first name* have a probability to change as 2% over the same time period. While the ratio between the two probabilities is 5:1, it does not immediately suggest that *first name* is five times more important than *last name*.

To control the impact of temporal models, a parameter $\alpha \in [0, \infty]$ is introduced during normalization, as shown in (8). When α is 0, the temporal model has the maximum impact in adjusting similarity output. When α is very large, the impact of the temporal model is close to none. The parameter α is chosen by the user based on domain knowledge. In Sect. 6, we will evaluate a range of α values.

$$s_r(r_a, r_b) = \frac{\sum_{A \in \mathbf{A}} (w_A(s_A(r_a.A, r_b.A), |r_a.t - r_b.t|) + \alpha) \cdot s_A(r_a.A, r_b.A)}{\sum_{A \in \mathbf{A}} (w_A(s_A(r_a.A, r_b.A), |r_a.t - r_b.t|) + \alpha)} \tag{8}$$

Algorithm 1. Record Linkage with Regression-based Temporal Model

```
Input:
    - A set of temporal record clusters for training: $\mathbf{C}$
    - A set of temporal records to be linked: $\mathbf{R}$
    - For each attribute $A \in \mathbf{A}$, a set of support attributes: $L_A$
    - A similarity threshold: $t_s$
    - An impact adjustment value: $\alpha$
Output:
    - A set of merged records, each record represents an entity: $\mathbf{R}'$
1: $T = hashtable()$   // A hashtable of training instances for each attribute
2: for $C$ in $\mathbf{C}$ do
3:     for $A$ in $\mathbf{A}$ do
4:         Create a training instance $i$ using $L_A$ and the approach described in Sect. 5.1.
5:         $T[A].append(i)$
6: $M_a = hashtable()$   // Agreement decay models
7: $M_d = hashtable()$   // Disagreement probability models
8: for $A$ in $\mathbf{A}$ do      // Train two models for each attribute
9:     $M_a[A] = decayModel(T[A])$
10:    $M_d[A] = linearModel(T[A])$
11: Get pairs of records $\mathbf{P}$ from $\mathbf{R}$ using Swoosh described in Sect. 4.
12: for $p$ in $\mathbf{P}$ do                          // For each pair of records
13:     $decaysAgree = hashtable()$      // Map $A \in \mathbf{A}$ to an agreement decay
14:     $probsDisagree = hashtable()$    // Map $A \in \mathbf{A}$ to a disagreement probability
15:     for $A$ in $\mathbf{A}$ do      // Calculate agreement decays and disagreement probabilities
16:         $decaysAgree[A] = M_a[A](p)$
17:         $probsDisagree[A] = M_d[A](p)$
18:     $s_r = \mathrm{sim}(decaysAgree, probsDisagree, \alpha)$   // Calculate similarity using (8)
19:     if $s_r >= t_s$ then
20:         $\mathbf{R}.removeRecords(p)$    // Remove the two records of the pair
21:         $r' = merge(p)$                  // Merge the pair of records into a new record
22:         $\mathbf{R}.push(r')$            // Add the new record to record set
23:         Create new records pairs using $r'$ then push the new pairs into $\mathbf{P}$
24: return $\mathbf{R}$ as $\mathbf{R}'$     // Return the merged records that cannot be merged any further
```

5.3 Algorithmic Overview of Regression-Based Temporal Linkage

Algorithm 1 describes the main steps of our approach, which integrates with our framework and produces a set of linked (merged) records from a set of temporal records **R**. From lines 1 to 5, the algorithm creates a list of training instances for each attribute. Each training instance contains a class value that indicates if the attribute value of an entity has been changed within a time period. From lines 6 to 10, the algorithm trains an agreement decay model and a disagreement probability model for each attribute, using the training instance sets created. From lines 11 to 23, the algorithm compares records in pairs, merges the record pair that is classified as a match into a new record, and compares the new record against the remaining records.

6 Experiments

In this section we first describe the datasets, baseline methods and measures used in our experiments. Then we present and discuss the experimental results.

6.1 Experimental Settings

Datasets: The real temporal datasets we used in this paper are from the North Carolina Voter Registration (NCVR) dataset collected every two months[1]. The datasets have ground truth (entity identifiers) available for all records. We selected the following attributes: *first*, *middle*, and *last name*, *name suffix*, *street address*, *city*, *gender*, and *age*. *Gender* was selected as the support attribute for *last name*, *age* was selected as the support attribute for *street address*, while the remaining attributes have no support attributes. The choice of support attributes was made according to domain knowledge. The NCVR dataset in total contains 8,336,205 entities, from which we randomly selected 5K, 10K, 50K, and 100K entities and their temporal records to create testing datasets. In each of the four testing datasets, 76% of the entities have one temporal record, 18.6% have two temporal records, 4% have three temporal records, and 1.4% entities have more than three temporal records. Each temporal record has one or more attribute value(s) that are different from the other records.

1K and 10K entities were randomly selected from the NCVR dataset and their temporal records are used for training. The 1K training dataset is used to train the models when using 5K and 10K testing datasets, and the 10K training dataset is used to train the models when using 50K and 100K testing datasets.

Measures: We used the standard quality measures of precision, recall, and F-measure to evaluate the record linkage quality [5] (noting recent work on how the F-measure can be misleading for record linkage when used to compare different classifiers at the same similarity threshold by weighting precision and recall differently [8]). Let $\mathbf{R}$ be a record linkage result in the form of clusters of records that are matching, and $\mathbf{S}$ be the ground truth that $\mathbf{R}$ corresponds to, which is also in the form of clusters of records. We calculate pair-wise precision $(P) = (|\mathbf{R} \cap \mathbf{S}|)/(|\mathbf{R}|)$, pair-wise recall $(R) = (|\mathbf{R} \cap \mathbf{S}|)/(|\mathbf{S}|)$, and $F_1 = 2 * P * R/(P+R)$. We used similarity thresholds $t_s = [0.6, 0.65, 0.7, 0.75, 0.8]$, and values for α in the range from 0 to 8 with an increment of 0.5. The highest F_1 score of each method was selected as the final result.

For string attributes, the similarity of a pair of attribute values was calculated using the Jaro-Winkler string comparison function [5]. The similarity of a pair of age values was calculated as: $s_{age} = 1/(|age_1 - age_2| + 1)$.

We implemented all algorithms in Python 2.7, and the experiments were conducted on a server with 64-bit Intel Xeon (2.4 GHz) CPUs, 128 GB of memory and running Ubuntu 14.04. We used the Sklearn package[2] for the linear regression classification. We implemented four algorithms for the experimental study. The first two are baselines, and the last two are the proposed approaches: (1) No model: A baseline approach with no temporal model. Weights of attributes were not adjusted by a temporal model. (2) Decay model (*Decay*): A baseline approach using the temporal model proposed by Li et al. [13] (Sect. 4). (3) Disagreement

[1] http://dl.ncsbe.gov/.
[2] http://scikit-learn.org.

probability regression model (*Disprob*): A temporal model which uses a regression model to predict the disagreement probability, and reduces the weights of attributes when their predicted disagreement probability is high (Sect. 5.1). (4) Disagreement probability combined with agreement decay regression model (*Mixed*): With the disagreement probability being predicted in the same way as the method above, the mixed method also calculates agreement decay from the decay model. The disagreement probability and agreement decay are combined to adjust the weight of each attribute (Sect. 5.1).

Table 2. Linkage results on the NCVR datasets with best results highlighted in bold.

Dataset	5K			10K			50K			100K		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
No model	**0.99**	0.93	0.96	**0.99**	0.90	0.94	0.94	**0.90**	0.92	0.91	**0.91**	0.90
Decay	0.96	0.95	0.96	0.95	0.93	0.92	**0.97**	0.87	0.92	**0.96**	0.88	0.91
Disprob	0.98	0.93	0.96	0.97	0.90	0.94	0.95	0.88	0.91	0.95	0.86	0.91
Mixed	0.97	**0.96**	**0.97**	0.94	**0.95**	0.94	0.96	0.90	**0.93**	0.95	0.90	**0.92**

Impact adjustment as discussed in Sect. 5.2 is implemented for *Decay* and *Disprob*, as well as the proposed algorithm *Mixed*, to allow a fair comparison.

6.2 Experimental Results

Table 2 shows the linkage results of the four algorithms on the testing datasets with impact adjustment. Results with the highest F_1 were selected. On the smaller testing datasets (5K and 10K), the *Mixed* approach achieved better recall but lower precision than the non-temporal baseline. On the larger testing datasets (50K and 100K), *Mixed* maintained similar recalls as the non-temporal baseline while performing better at precision. The result shows that our technique performs better when a dataset is large, while it does not perform worse than other techniques on smaller datasets. Because the 50K and 100K datasets used a larger training set of 10K entities the improvements would be due to this larger number of training records.

One significant difference between the smaller and the larger testing datasets is the percentage of non-matching record pairs. Even with blocking [5], the number of non-matching pairs still grows faster than linear with respect to the size of a dataset. A linkage algorithm will encounter non-match pairs more often when the testing dataset is large. As a result, we can observe that the precision of the *No model* approach decreases as the size of a dataset increases.

Figure 1 shows the effect of the impact adjustment parameter α (see Sect. 5.2). The temporal models did not perform well when the impact adjustment was not applied ($\alpha = 0$). It implies that directly applying probabilities on the weights can over-weight some attributes and decrease the linkage quality.

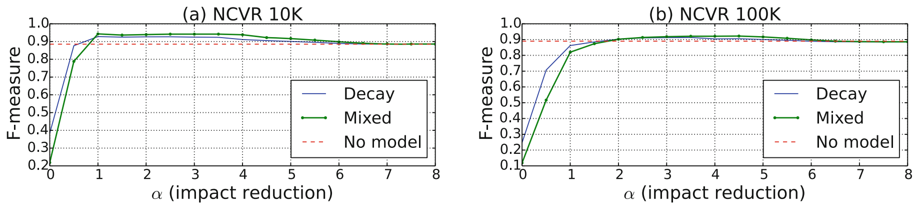

Fig. 1. The effect of different values for the impact reduction parameter (α). Without impact reduction ($\alpha = 0$), the temporal models (*Decay*, *Mixed*) performed poorly. At a certain point, the temporal models start to outperform the non-temporal baseline approach (*No model*). With the impact being reduced further (α increases), the temporal models eventually performed the same as the non-temporal baseline because the impact of the models has been reduced to the extent that is not significant anymore.

7 Conclusion and Future Work

In this paper we have developed a temporal model to improve the quality of temporal record linkage. Our model uses a linear regression model and multiple attribute values to predict the probability for an attribute value to change within a certain time period, and the model adjusts the weight of the attribute used in similarity calculations accordingly. The intuition of our approach is to use the dependency between attributes to predict their probability to change over time more accurately. We evaluated our approaches on four real-world datasets derived from the NCVR database. The experimental results show that our approach performed better than two baseline approaches.

In the future, we will investigate attribute dependencies for calculating the probability for two different entities to share the same attribute value (agreement probability), which can also be affected by other attributes. For example, two people who have the same phone number have a high probability to have the same address. We also aim to incorporate a frequency based weighting strategy into our framework to see if undesired high similarities can be adjusted properly. Another possible direction is to test the temporal models with different clustering techniques, such as those proposed by Li et al. [13] and Chiang et al. [4].

References

1. Benjelloun, O., Garcia-Molina, H., Menestrina, D., Su, Q., Whang, S.E., Widom, J.: Swoosh: a generic approach to entity resolution. VLDB J. **18**(1), 255–276 (2009)
2. Blum, A.L., Langley, P.: Selection of relevant features and examples in machine learning. Artif. Intell. **97**(1–2), 245–271 (1997)
3. Chiang, Y.H., Doan, A., Naughton, J.F.: Modeling entity evolution for temporal record matching. In: ACM SIGMOD, Snowbird, Utah (2014)
4. Chiang, Y.H., Doan, A., Naughton, J.F.: Tracking entities in the dynamic world: a fast algorithm for matching temporal records. PVLDB **7**(6), 469–480 (2014)
5. Christen, P.: Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Springer, Heidelberg (2012)

6. Christen, P., Gayler, R.W.: Adaptive temporal entity resolution on dynamic databases. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013. LNCS (LNAI), vol. 7819, pp. 558–569. Springer, Heidelberg (2013). doi:10.1007/978-3-642-37456-2_47
7. Fellegi, I.P., Sunter, A.B.: A theory for record linkage. JASA **64**(328), 1183–1210 (1969)
8. Hand, D., Christen, P.: A note on using the F-measure for evaluating data linkage algorithms. Pre-print NI16047, Isaac Newton Institute, Cambridge, UK (2016)
9. Herzog, T.N., Scheuren, F.J., Winkler, W.E.: Data Quality and Record Linkage Techniques. Springer, New York (2007)
10. Krueger, D., Montgomery, D.C., Peck, E.A., Vining, G.G.: Introduction to Linear Regression Analysis. Wiley, Hoboken (2015)
11. Kum, H.C., Krishnamurthy, A., Machanavajjhala, A., Ahalt, S.C.: Social genome: putting big data to work for population informatics. IEEE Comput. **47**(1), 56–63 (2014)
12. Li, F., Lee, M.L., Hsu, W., Tan, W.C.: Linking temporal records for profiling entities. In: ACM SIGMOD, Melbourne (2015)
13. Li, P., Dong, X.L., Maurino, A., Srivastava, D.: Linking temporal records. PVLDB **4**(11), 956–967 (2011)