Towards Automated Record Linkage

Karl Goiser **Peter Christen**

Department of Computer Science,
The Australian National University,
Canberra ACT 0200, Australia
Email: {Karl.Goiser, Peter.Christen}@anu.edu.au

Abstract

The field of Record Linkage is concerned with identifying records from one or more datasets which refer to the same underlying entities. Where entity-unique identifiers are not available and errors occur, the process is non-trivial. Many techniques developed in this field require human intervention to set parameters, manually classify possibly matched records, or provide examples of matched and non-matched records. Whilst of great use and providing high quality results, the requirement of human input, besides being costly, means that if the parameters or examples are not produced or maintained properly, linkage quality will be compromised. The contributions of this paper are a critical discussion on the record linkage process, arguing for a more restrictive use of blocking in research, and evaluating and modifying the farthest-first clustering technique to produce results close to a supervised technique.

1 Introduction

Record Linkage is concerned with the process of identifying records from one or more datasets which refer to the same entities (e.g. people, organisations or objects) (Winkler 2006). Where applied to a single dataset, the process is known as *de-duplication*. The utility of Record Linkage lies in its ability to provide information that would otherwise be impossible or too costly to obtain. For example, a linkage of hospital records with motor vehicle accident data could provide information about the required procedures and outcomes for different types and severities of accidents (Christen & Goiser 2005, Winkler 2006). Record linkage is often used in the initial, preprocessing phase of data mining projects in order to enrich data or remove duplicate records. This paper is divided into three parts: this introduction, a critical discussion of the nature and some of the major issues of Record Linkage, and an experimental part which leverages the discussion and examines the possibility of conducting record linkage without human intervention.

1.1 The Record Linkage Process

Historically, the process predates computers (Gill 2001), but it wasn't until their advent and common use, together with increasing storage of information, that significant advances were made. Newcombe developed the basic idea of probabilistic linkage in the 1950's, and the mathematical foundation was set down by Fellegi and Sunter in 1969 (Fellegi & Sunter 1969).

Consider a population of *entities* (people, businesses, products, etc.) from which are drawn one or more *datasets*, some of whose *records* may refer to the same entities. The drawing down process may be the entering of information relating to a patient's hospital visit, the result of a credit card transaction, adding a new customer into a database, or recording a birth. Each entity may appear more than once in a single dataset (e.g. multiple credit card transactions, mothers giving birth, etc.) and in more than one dataset. The process of entering information about the entities into a dataset may be subject to errors such as typing mistakes, miss-spellings, optical character recognition errors, etc. There may also be differences due to the use of abbreviation or varying amounts of detail recorded. Thus finding the records which relate to the same entities can be seen to be non-trivial in the sense that no simple exact search, database join or sort could find them (Christen & Goiser 2005).

This paper assumes the linkage of two datasets, $\mathbf{A}$ and $\mathbf{B}$, with neither dataset containing duplicate records (Christen & Goiser 2005). There may be some records in $\mathbf{A}$ which refer to the same entities as records in $\mathbf{B}$, and it is the task of Record Linkage to find them. The linkage process involves two basic steps, *comparison* and *classification* (Winkler 2006). The comparison step takes pairs of records from the cross-product of the datasets, $\mathbf{A} \times \mathbf{B}$, and, for each pair, produces a vector of one or more values indicating the level of similarity or difference between attributes of the records which were compared. The values can be categorical, ordinal or numeric, but are generally real values in the range [0,1] with increasing values representing increasing similarity (Winkler 2006). The vectors place the comparison of a record pair into a space whose dimension is equal to the number of attributes compared. From these vectors, the classification step determines the class of each pair as either a *match*, or a *non-match* (Christen & Goiser 2005) (the *possible-match* classification used in the Fellegi and Sunter approach will be discussed below). Classification techniques are generally either *supervised* or *unsupervised* (Mitchell 1997). Supervised techniques use pre-classified data to generate a *classifier* which is then used to determine the class. Variations on these steps are possible, however, there must be some form of comparison between the records as well as a determination of the class.

Considering each attribute comparison, there are two distributions, one each for the matches and non-matches. Figure 1 shows this for a dataset used in the experiments in Section 3. The normal process is to select a threshold value which places as many matches as possible above, and as many non-matches below it.

Table 1: Confusion matrix of record pair classification

Actual	Classifications	
	Match	Non-match
Match	True match True positive (TP)	False non-match False negative (FN)
Non-match	False match False positive (FP)	True non-match True negative (TN)

Errors occur where comparisons of matches fall below the threshold, or non-matches above it. The confusion matrix in Table 1 describes the types of errors than can be produced (Christen & Goiser 2005).

It can be noted that, if the cost of making a false match is different from the cost of making a false non-match, it could be possible to find threshold values which, while increasing the number of false classifications, could decrease the overall costs. For example, in a cancer screening test, lowering the threshold value could decrease the number of undetected cancers (false negatives) at the expense of a higher number of erroneous detections (false positives), thus increasing the cancer detection rate.

In the traditional Fellegi and Sunter approach a third class is allowed: *possible-matches* (Fellegi & Sunter 1969). Compared record pairs are assigned to this class where the criteria are not strong enough to make a definitive match or non-match decision. The resulting pairs are referred to manual *clerical review* – where human judgement, with possible reference to more information, can allow a determination to be made. However, this can be problematic, as comparing even medium-sized datasets with only a few thousand records results in millions of comparisons, and a possible-match rate of only 1% will require thousands of manual classifications. These will take time – delaying the project – and adding to costs. Also, human decision-making has an inherent bias and error rate. As the motivation of this paper is to work towards an automated record linkage process, this class will not be considered further.

1.2 Blocking

Comparing the cross-product of $\mathbf{A} \times \mathbf{B}$ results in quadratic complexity, and is thus difficult or impossible for large datasets. Different *blocking* techniques such as standard blocking, sorted-neighbourhood, bigram indexing and canopy clustering have been developed to ameliorate this problem (Baxter, Christen & Churches 2003, Elfeky, Verykios & Elmagarmid 2002). The idea is to cheaply filter out obvious non-matches before executing the more detailed and time-consuming comparisons.

As an example of the process, standard blocking criteria uses record attributes such as postal/zip code to create blocks of similar records, with the comparisons then only being between records in the same block. A problem with this approach is that inaccuracy in the criteria can lead to records being placed in the wrong blocks, thus removing them from being compared with any potential matches. This can be solved by conducting several passes using different criteria each time (Winkler 2006). In (Winkler 2005), it is suggested that a 10-pass blocking strategy will reduce the number of record pair comparisons required for the linkage of two datasets with one million records each from 10^{12} to around $10^7 - 10^8$.

1.3 Modern Approaches

In recent years, researchers have started to incorporate techniques from Machine Learning, Data Mining, Information Retrieval, and Artificial Intelligence research to improve the linkage process. A popular approach (Bilenko & Mooney 2003, Chaudhuri, Ganjam, Ganti & Motwani 2003, Cohen, Ravikumar & Fienberg 2003, Yancey 2004, Zhu & Ungar 2000) has been to learn distance measures (like edit-distance) that are used for approximate string comparisons (Christen 2006). As shown in (Cohen et al. 2003), combining different learned string comparison methods can result in improved linkage classification. An Information Retrieval–based approach (Cohen 1998) is to represent records as document vectors and to compute the cosine distance between such vectors, while (Nahm, Bilenko & Mooney 2002) explore the use of support vector machines to classify record pairs.

Active learning is used in (Sarawagi & Bhamidipaty 2002) and (Tejada, Knoblock & Minton 2002) to address the problem of lack of training data. The basic idea is to iteratively select for human determination, a comparison which is the hardest for the technique to classify, then learn from that and build a better classifier. This has the effect of significantly reducing the number of manual determinations required. A hybrid system is described in (Elfeky et al. 2002) which utilises both unsupervised (clustering) and supervised (instance-based learning and decision trees) machine learning techniques. Active learning is also used in (Michalowski, Thakkar & Knoblock 2004), however, secondary data sources are used in place of human input, thus making their approach unsupervised. This work is part of an increasing interest in the application of record linkage to web-based data and technologies.

High-dimensional overlapping clustering is applied in (McCallum, Nigam & Ungar 2000) as an alternative to traditional blocking (in order to reduce the number of record pair comparisons to be made), while in (Gu & Baxter 2004*a*), the use of simple K-means clustering together with a user-tunable fuzzy region for the class of possible-matches is investigated. Methods based on nearest neighbours are explored in (Chaudhuri, Ganti & Motwani 2005), with the idea being to capture local structural properties instead of a single global distance approach. Graphical models (Ravikumar & Cohen 2004) are another unsupervised technique that aims at using the structural information available in the data to build hierarchical probabilistic models for record pair classification.

Many of these new approaches are based on supervised learning techniques and require training data which is often not available in real world situations, or only obtainable via manual preparation (a costly process similar to manual clerical review). Additionally, many of the recent publications in this area present experimental studies that are based on small datasets of up to a couple of thousand records (Christen 2005).

2 Critical Discussion

This section discusses some problematic issues affecting the record linkage process: task complexity, parameter freedom, the availability of training data, and blocking - all affecting the ability to automate the linkage process.

2.1 Task Complexity but Match Rarity

With at most one true match between each record in **A** and **B** (assuming no duplicates), the largest possible number of matches is the smaller of the size of **A** and **B**, $n = min(|\mathbf{A}|, |\mathbf{B}|)$, with $|\cdot|$ denoting the number of records in a dataset. Where $n = |\mathbf{A}| = |\mathbf{B}|$, there can be no more than n matches: every record in **A** is linked to a different record in **B** (Christen & Goiser 2005). However, as every record in **A** potentially needs to be compared against every record in **B**, the number of comparisons required is $|\mathbf{A}| \times |\mathbf{B}|$ which is n^2 for $n = |\mathbf{A}| = |\mathbf{B}|$. Thus, while the number of matches increases linearly with the size of the data, the number of comparisons required increases quadratically. When de-duplicating a dataset, all records potentially need to be compared with all others, thus requiring $n(n-1)/2$ comparisons.

This result has significant ramifications for Record Linkage. For example, in many areas such as social security, health, tax, corporate customer information, a dataset of one million records is considered to be small, yet conducting a million times a million, 10^{12}, complicated comparisons would not be considered viable due to the time it would take. For example, if a comparison requires 0.1 milliseconds per attribute and there are 10 attributes to compare, the linkage of two datasets with one-million records each (assuming no blocking) would require 10^9 seconds which is nearly 32 years!

The other aspect of the linear increase in matches versus the quadratic increase in the problem size is that the matches become rare. In the above example, while 10^{12} comparisons are potentially required, the maximum number of matches is 10^6. The rate of matches to comparisons – the 'hit' rate – for these datasets is one-in-a-million.

Figure 1 shows density plots of a real-world sampled dataset (n = 996,166, comprising 353 matches and 995,813 non-matches, a match rate of 1/2,822) (Centre for Epidemiology and Research, NSW Department of Health 2001). It can be seen that, while matches form a distinct grouping with their own modal peak, they hardly register in comparison to the large number of non-matches. This must be considered when selecting and using classification methods. Even when blocking is applied, the number of matches to non-matches is often very different. Thus, while the complexity of the task becomes difficult for medium- to large-sized datasets, the matches themselves become increasingly rare.

2.2 Parameter Freedom

Many comparison and classification techniques allow tuning in order to increase their accuracy, or to allow trading off one benefit in order to increase another. For example, in the Fellegi and Sunter model, the threshold values between non-matches, possible-matches and matches are user-adjustable – changing the values will alter the linkage quality as well as the number of record pairs set aside for clerical review (Fellegi & Sunter 1969).

Accurate setting of parameters requires a high degree of knowledge of the techniques used as well as of the characteristics of the data in question, and can take a significant amount of time and effort to determine properly. Once set, the parameters can only be re-used in further linkages with confidence if the characteristics of the new data were such that the same values would be generated. This paper addresses this problem by choosing to research methods which do not require parameters to be set.

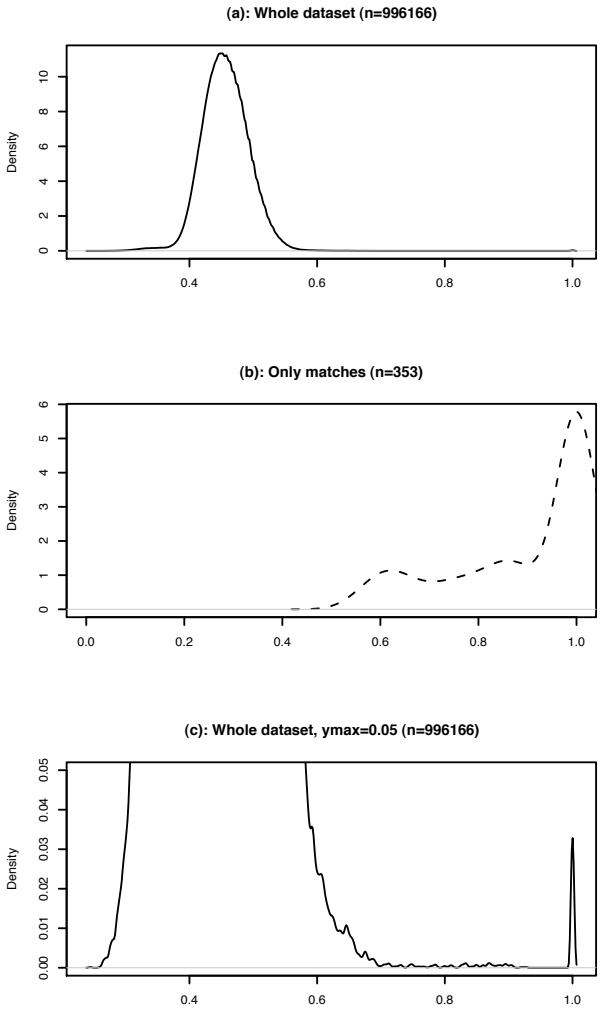


Figure 1: Density plots of one of the sampled datasets for: (a) the whole dataset, (b) just the matches, (c) a 20× y-axis magnification of (a) showing the peak for the matches.

2.3 Availability of Example Data

In order to set parameters, some knowledge of the data must be obtained, whether through data analysis, or supervised Machine Learning techniques (Mitchell 1997). This requires the availability of *example* or *training* data - that is, data for which the match/non-match state is known in advance.

One way of generating such example data is to randomly sample pairs of records and manually classify them. However, considering the size of datasets and the rarity of matches within them, many thousands of record pairs may need to be examined in order to obtain a few examples of matched records, and it may not be known if they form a representative sample of all the matches. A solution to this is to use a technique like active learning to bias the sampling of examples in order to increase the representation of matches (Sarawagi & Bhamidipaty 2002).

Sampling bias and representation must be considered when making use of the example data.

Example data are thus subject to the same sorts of problems as associated with parameters (as discussed in Section 2.2). Further, given that example cases are provided as input to generate parameters, they can be seen as parameters in themselves - different examples will generate different parameter values.

2.4 Blocking

To the extent that blocking removes comparisons in a consistent fashion, it becomes a source of bias - a confounding factor - which, in fact, is its intent: to consistently remove obvious non-matches. However, it must be recognised that biased data will have an effect on the results of classification methods which adapt to or learn from that data. Where blocking removes true matches, it can be seen as a failing, and the extent to which this occurs with consistency is, again, a source of bias. As an example, (Gu & Baxter 2004*a*) block four small datasets and show that between 8% and 30% of the true matches are removed by blocking.

Other issues with blocking include the question of the amount of time saved: from the above, does the reduction from 10^{12} to 10^{8} (or similar amounts) merely change the problem from impossible to unfeasible? Also, errors in the data or the blocking criteria may mean that no matter how many passes are conducted, some true matches won't be passed through (e.g., where they aren't assigned to the same block in standard blocking). Thus, setting up the blocking criteria requires knowledge of the data and the blocking technique used. As an example, standard blocking works optimally when the data is evenly distributed into a moderate number of blocks. However, if only a single block is too large (e.g. a block with surname value of "Smith"), the quadratic issue returns. See (Gu & Baxter 2004*b*) for further discussion, and a potential solution. Note that blocking criteria, again, are parameters - see the discussion on parameter freedom above.

To the extent that different blocking methods and blocking criteria (including not blocking) result in different data, they produce different biases. Thus comparisons between classification methods become invalid or difficult if different blocking methods or criteria are used. The important question to ask about the results of research which uses blocking then becomes: if different blocking methods or criteria were used, can it be shown that the results would be the same? If not, blocking can be regarded as integral to the process, and cannot be divorced from it.

Given these problems, it is strongly recommended that researchers only block their data if it is too large to feasibly conduct a linkage on, or if the research is into blocking techniques. With the ready availability of relatively fast personal computers with large main memory, it cannot be seen how it could be justified to use blocking in research which would require fewer than one million comparisons.

It is noted, however, that without blocking, the linkage or de-duplication of large datasets could not be accomplished. It is thus necessary but problematic, and careful attention must be paid to its use. In using blocking, it must be understood that potential matches will be removed, and that the data will be biased which may affect the results of further procedures.

3 Experiments

Given the above discussion, it was decided to examine the feasibility of parameter-free techniques for record linkage - that is, investigate if techniques which don't require parameters can produce results comparable to those which do. All the presented experiments were conducted without using blocking.

3.1 Experimental Setup

Three comparison and three classification methods were chosen. The *Febrl* (Freely Extensible Biomedical Record Linkage) (Christen, Churches & Hegland 2004) open source record linkage system was used for the comparison step, while the *Weka* (Witten & Frank 2005) open source data mining package was used for the classification step.

3.1.1 String Comparison Methods

Of the three methods, the first two were chosen because they are commonly used in record linkage, and the third because it is novel in this field and could prove to be potentially useful. None of these methods require parameters.

In the **Jaro-Winkler** (**JW**) comparison method (Winkler 1990, Winkler 2006, Yancey 2006) a similarity score based on the number of common characters, character transpositions, and string lengths, as well as giving a higher score for having a common prefix of length up to 4 characters, is calculated.

In the **edit-distance** (**ED**) method, a similarity score is calculated using the normalisation of the minimal number of single-letter insertions, deletions and substitutions required to transform one string into the other (Winkler 2006, Yancey 2006).

Compression comparison (**CC**) (Cilibrasi & Vitanyi 2005, Keogh, Lonardi & Ratanamahatana 2004) uses the fact that the compression of the concatenation of two similar strings is shorter than that of dissimilar ones. The normalised compression distance was used:

$$NCD(x,y) = \frac{C(xy) - min(C(x), C(y))}{max(C(x), C(y))}$$

where $C()$ is a compression algorithm such as `zlib` or `GZip`, and xy is the concatenation of the strings, x and y.

3.1.2 Classification Methods

One supervised and two unsupervised classification methods were chosen. The supervised method requires training data, and, being partitioning clustering techniques, the unsupervised methods require the specification of the number of clusters. As the aim is to have a cluster of matches and a cluster of non-matches, this value is fixed at two. Being fixed, the value doesn't change meaning it is not supplied as a parameter.

Decision trees (**DT**) are one of the major Machine Learning techniques (Mitchell 1997). The normal procedure is to use training data to build a *classifier*, which is then used to classify further data. For this work, they are used as a base line – to compare the results of the unsupervised methods against. As such, to give the best possible results, *all* the data in the dataset under investigation is used for both training *and* testing.

***K*-means** (**KM**) is a commonly used simple unsupervised clustering technique (Han & Kamber 2001). Previous papers which have looked at K-means in the context of record linkage include (Elfeky et al. 2002, Gu & Baxter 2004*a*).

The **farthest-first** (**FF**) clustering technique was first presented in (Gonzalez 1985). It is a very simple algorithm and very fast: assign the centroid for the first cluster to a random point. For the second centroid, choose the point which is farthest from it. For all following centroids, choose the point which is farthest from all the centroids chosen so far.

It is interesting in that, unlike **KM** which can halt at local minima, it is guaranteed to provide a solution within two times the optimal solution value of the objective function used to choose the clusters - the *2-approximation problem* (Gonzalez 1985).

3.2 Data Sources

Three data sources were used in the experiments, two synthetically generated, and one real-world dataset.

3.2.1 Synthetic Data

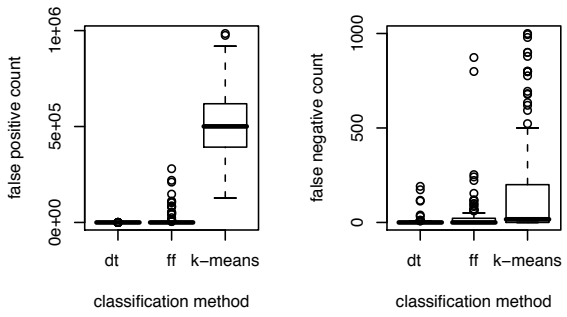
Synthetic data was generated using the dataset generator from *Febrl* (Christen et al. 2004), which allows probabilistic creation of records, as well as simulation of common types of errors at specifiable rates (Christen 2005). Two groups of datasets were generated, one with a maximum of one error per record, and a second with up to 3 errors in any attribute. The generated attributes were: given-name, surname, street-number, street-type, street-value, suburb, postcode, state, date-of-birth, age, phone-number, and social security identifier. For each group, seven pairs of datasets were generated with 0%, 10%, 20%, 50%, 80%, 90% and 100% overlap – the amount of duplication between the datasets, with 0% being no common record, and 100% being duplicate datasets. Each of the pairs of datasets contained one thousand records, thus requiring one million comparisons. From these datasets, variations were generated using different concatenations of the original attributes: **one**: the attributes were concatenated into a single attribute; **three**: the attributes were concatenated into 'name', 'address', and 'other' attributes; **all**: all attributes were kept. That is, generating the variation involved taking the attributes to use and joining them together, delimited by a space, into a new attribute.

The total number of linkages on synthetic datasets were thus: 2 different numbers of errors per record, 7 types of overlap, 3 combinations of attributes, 3 comparison methods, 3 classification methods, giving a total of **378** discreet record linkages of a million comparisons each.

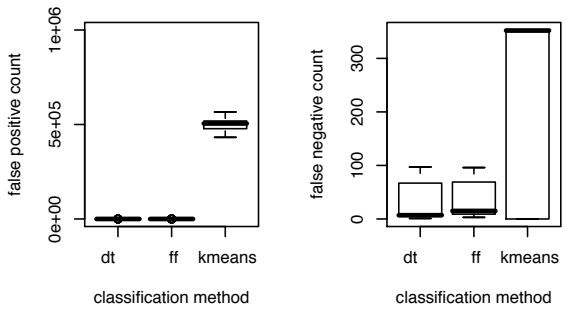
3.2.2 Real-World Data

Access was available to a confidential dataset in which extensive effort had previously been made to correctly classify record duplicates. This dataset, the *New South Wales Midwives Data Collection* (MDC) (Centre for Epidemiology and Research, NSW Department of Health 2001), was provided by the New South Wales Department of Health. It contains 175,211 records relating to births in the years 1999 and 2000 and had been de-duplicated using the commercial probabilistic software, *AutoMatch* (MatchWare Technologies 1998), including post-linkage manual clerical review. Of the 175,211 records, it had been determined that 158,081 mothers appeared once, 8,295 appeared twice, 176 appeared three times, and 3 appeared four times in the dataset.

As an exhaustive de-duplication would require 15,349,359,655 comparisons, it was decided to sample the data, and have about the same number of comparisons as the synthetic data. A sample size



(a) combined synthetic data (all three comparison methods)



(b) MDC (**CC** only)

Figure 2: Errors by classification method for (a) the combined synthetic data, and (b) the MDC data.

of $n = 1,412$, resulting in $n(n-1)/2 = 996,166$ comparisons was decided on. Each sample comprised 353 randomly selected matched pairs and 706 randomly selected non-matched records. Fifty samples were drawn, the three attribute combinations were generated, and the **CC** comparison method was used with all three classification methods. For the MDC data, there were thus a total of **450** discrete record linkages of 996,166 comparisons each.

3.3 Results

Note that unless described as density plots (where the area under the curve is 1), the graphs all use boxplots. A boxplot is a concise graphical device showing the upper and lower quartile (the box), the median (the line crossing the box), and the range (the ends of the lines extending from the box, often called whiskers). Where a data point is suspected to be an outlier, it is plotted as a point, and the whisker is then set at 1.5 times the inter-quartile range. Where the median is offset within the box, it is an indication of skewness in the data.

How does the choice of classification method affect the results?

The boxplots in Figure 2 describe the errors produced by the classification techniques used for the synthetic datasets (combined), and the MDC data. It can be seen that **KM** performs worse than the other two. In fact, for **KM**, the mean number of false positives for all the linkages in the synthetic data is 509,064. Since there were one million comparisons, this means that the method does slightly worse than chance. (For brevity, **KM** will not be further discussed when comparing classifiers.) Otherwise, it is of interest that **FF** has comparable results with **DT**.

Why does K-means do so badly?

Han and Kamber point out that, "the K-means method is not suitable for discovering clusters with non-convex shapes or clusters of very different size" (Han & Kamber 2001) (p. 350). For a linkage without blocking as in these experiments, there is an overwhelming number of non-matches (for the synthetic data: $10^6 - (10^3 \times overlap\%/100)$, and for the sampled MDC data: $996,166 - 353$). These different cluster sizes can be seen to be the cause of why **KM** does so badly.

The reason that **KM** has been used successfully in record linkage, e.g. (Gu & Baxter 2004*a*), is that it has been preceded by blocking which has had the effect of evening-up the class sizes. Thus, for **KM** to be successful in record linkage, blocking must have been previously used – and had the effect of evening up the class sizes.

How does the choice of comparison method affect the results?

Figure 3 shows boxplots of the error counts for the combined synthetic experiments. For the false negatives, it can be seen that **FF** performs comparably with **DT** except for two cases of **JW** comparisons (which may be due to **FF** randomly selecting centroids which are not be near the mode of the true negatives). With the false positives, the errors appear larger, increasingly for **ED** and **JW**. As most of these errors are also associated with the **all** concatenation group, it can be seen that **JW** does not provide good discrimination for **FF** when provided with a larger number of comparisons. Note that the median number of false positive errors is 1.5 and 1.0 for **CC** and **ED** comparisons respectively, while the median false negative values are 0, so the results are actually very good for most of the datasets.

Density plots of the results of the different comparison methods are shown in Figure 4. The modes of the distributions of the two classes are closest together for **JW**. As **FF** uses a threshold midway between the centroids to determine the class, it can be seen that the more the overlap between the modes, the more errors would be generated. Other aspects that can be seen to affect accuracy are the skewness of the distributions, and the combination of spread and distance between centroids. For example, the larger the spread and the closer the centroids the more the overlap, so the greater the number of errors. Thus, it can be seen that **FF** performs better in conjunction with **CC** or **ED**.

Can a reasonable result be obtained with parameter-less techniques?

Linking synthetic datasets with some overlap, using **ED** for comparisons of the attributes concatenated together and **FF** for classification, resulted in 0 false positives and a mean of 6.833 (median 0.5) false negative errors. For the MDC dataset, use of **CC** on the three attribute datasets with **FF** classification resulted in a mean of 4.68 (median 3.00) false positives and a mean of 9.76 (median 9.00) false negatives. Regularly having less than ten errors in around one million comparisons is a very small error rate. This is a strong indication that a linkage of reasonable quality can be obtained without requiring parameters.

Why does farthest-first do so well?

Examining the **FF** algorithm and considering Figure 1, it can be seen that a random choice of item will almost always make a selection near the mode

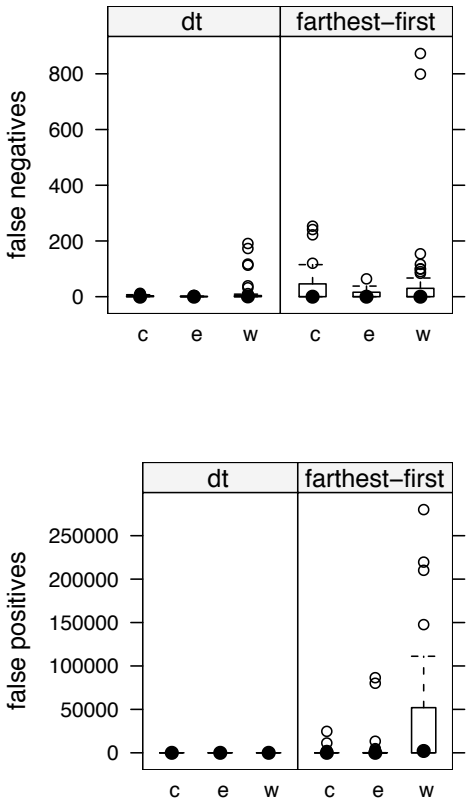


Figure 3: Errors by comparison methods (c=**CC**, e=**ED**, w=**JW**) for **DT** and **FF** classification methods using the combined synthetic datasets.

of the true-negatives for the first centroid, and for the second, it follows that the farthest from it will be the item with the maximum value - the pair with the highest similarity results. As the threshold value is midway between these two centroids, and since both **ED** and **CC** fulfil triangle inequality (Cilibrasi & Vitanyi 2005, Marzal & Vidal 1993), it can be seen that all results on one side of the threshold will be closer to its centroid than any on the other side. Thus, by not conducting blocking, an important characteristic of the data has been retained – one which allows **FF** to return very positive results.

3.4 Attempt to Improve Farthest-First

An improvement would be to remove the random choice for the first centroid, thus removing the chance of choosing an item away from the mode of the true negatives. To examine this, the original Weka **FF** algorithm was translated into the Python programming language[1], and four alternatives for picking the centroids were implemented:

- **Default**: the normal selection process for **FF**.
- **Mode**: for each attribute, bin the values, then choose the largest bin as the first centroid. This is equivalent to producing a histogram plot and selecting the longest bar – a crude density-based selection process. One thousand bins were used across the range of the data values.
- **0-1**: choose the values, 0 and 1 as the centroids. The idea here is that the comparison routines

[1] `www.python.org`

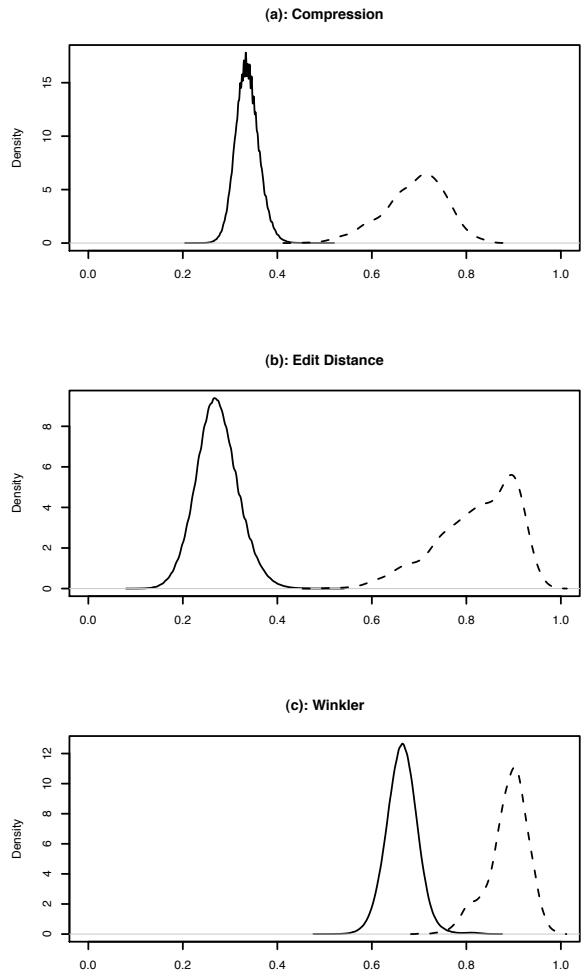


Figure 4: Density plots of the results for the different comparison methods when applied to the dataset with 100% overlap, and up to 3 errors in any attribute. The left peak is that of the non-matches, and the right is that of the matches. Note that each graph contains two separate density plots superimposed (as denoted by the solid and dashed lines).

return values in the range, 0..1. Thus, no processing is required to discover these centroids.

- **Range**: use the minimum and maximum values of the data as the centroids. This can be seen to be the same as normalising the comparison results and choosing the **0-1** modification.

Figure 5 shows the results of applying these modifications to the combined synthetic datasets. Note that **Default** differs slightly from the Weka version in that the latter normalises the attributes, but the Python implementation doesn't since the input data is already in the range 0..1. Also, as both Weka and **Default** use random selection, there is inherent variation in the results.

It can be seen that false positive rates for modifications **0-1** and **Range** appear higher than the others, while their false negative rates are very low. This indicates that the threshold values are set too low. However, the **ED** results for modification **0-1** show little or no errors. This modification uses 0 and 1 for the centroids, and earlier discussion has shown that **ED** separates the modal peaks more than the other methods. **Mode** does appear to show advantages over **Default** in that the three outliers in the false negative graph have been removed. However, differences are very minor as would be expected - except that possible selection of centroids away from the mode had been eliminated. These results were similar for the MDC dataset.

4 Conclusion and Future Work

Record Linkage was introduced and some characteristics and challenges to the field were presented. Blocking was discussed and it was noted that it can bias results unless controlled-for in experiments. This can also compromise the comparison of record linkage techniques. It was therefore recommended that blocking not be used in research unless necessary.

Experiments using unsupervised techniques in the linkage process were conducted and it was found that the K-means clustering technique was not suitable for the linkage of data which had not previously been blocked. The use of the farthest-first classification technique on non-blocked data was found to produce very promising results.

In terms of further research, the complexity of the edit-distance comparison method is the product of the lengths of the strings being compared, which is costly where the strings are long. An improvement would be to use other versions of the technique, such as those used in genetics (Christen 2006). Given the positive results for edit-distance with the modification to farthest-first classification on the synthetic datasets, further examination using real-world datasets will be conducted. The Expectation Maximisation (EM) algorithm (Winkler 1988) provides a method whereby parameters such as the thresholds in the Fellegi and Sunter model (Fellegi & Sunter 1969) can be estimated. Further work will include comparing EM against the farthest-first classification method.

Copies of this paper, the Febrl record linkage system, and other publications can be obtained from: http://datamining.anu.edu.au/linkage.html.

Acknowledgments

This work is supported by an Australian Research Council (ARC) Linkage Grant LP0453463 and partially funded by the NSW Department of Health.

References

Baxter, R. A., Christen, P. & Churches, T. (2003), A comparison of fast blocking methods for record linkage, *in* 'ACM SIGKDD'03 Workshop on Data Cleaning, Record Linkage, and Object Consolidation', Washington, DC, USA, pp. 25–27.

Bilenko, M. & Mooney, R. J. (2003), Adaptive duplicate detection using learnable string similarity measures, *in* 'Proceedings of ACM SIGKDD', ACM Press, Washington DC, pp. 39–48.

Centre for Epidemiology and Research, NSW Department of Health (2001), 'New South Wales mothers and babies 2001', *NSW Public Health Bull* **13:S-4**.

Chaudhuri, S., Ganjam, K., Ganti, V. & Motwani, R. (2003), Robust and efficient fuzzy match for online data cleaning, *in* 'Proceedings of ACM SIGMOD', San Diego, pp. 313–324.

Chaudhuri, S., Ganti, V. & Motwani, R. (2005), Robust identification of fuzzy duplicates, *in* 'Proceedings of the 21st international conference on

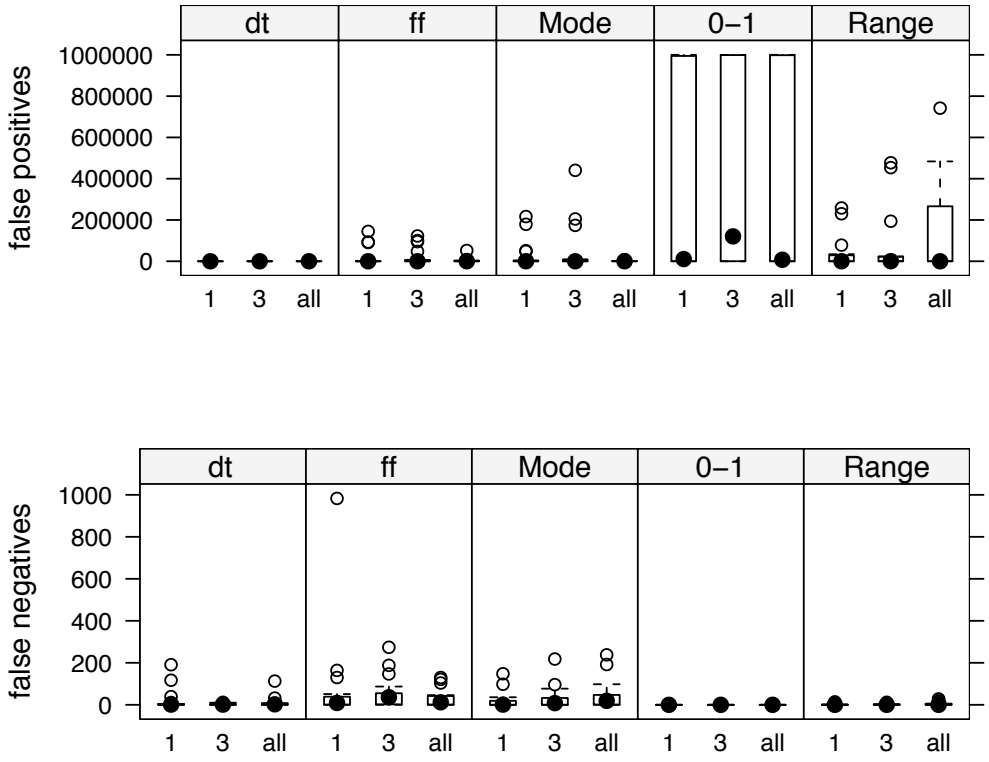


Figure 5: Errors by concatenation method for the classification methods (dt = **DT**, ff = unmodified Python **FF**) for data from the synthetic experiments combined.

data engineering (ICDE'05)', Tokyo, pp. 865–876.

Christen, P. (2005), Probabilistic data generation for deduplication and data linkage, *in* 'IDEAL'05', Springer LNCS 3578, Brisbane, pp. 109–116.

Christen, P. (2006), A comparison of personal name matching: Techniques and practical issues, *in* 'The Second International Workshop on Mining Complex Data (MCD'06)'.

Christen, P., Churches, T. & Hegland, M. (2004), Febrl – A parallel open source data linkage system, *in* 'Proceedings of the 8th PAKDD', pp. 638–647.

Christen, P. & Goiser, K. (2005), Assessing deduplication and data linkage quality: What to measure?, *in* 'Proceedings of the fourth Australasian Data Mining Conference (AusDM 2005)', Sydney.

Cilibrasi, R. & Vitanyi, P. (2005), Clustering by compression, *in* 'IEEE Trans. Information Theory', Vol. 51, pp. 1523– 1545.

Cohen, W. W. (1998), Integration of heterogeneous databases without common domains using queries based on textual similarity, *in* 'Proceedings of ACM SIGMOD', Seattle, pp. 201–212.

Cohen, W. W., Ravikumar, P. & Fienberg, S. (2003), A comparison of string distance metrics for name-matching tasks, *in* 'Proceedings of IJCAI-03 workshop on information integration on the Web (IIWeb-03)', Acapulco, pp. 73–78.

Elfeky, M. G., Verykios, V. S. & Elmagarmid, A. K. (2002), TAILOR: A record linkage toolbox, *in* 'Proceedings of ICDE', San Jose, pp. 17–28.

Fellegi, I. P. & Sunter, A. B. (1969), A theory for record linkage, *in* 'Journal of the American Statistical Association', Vol. 64, pp. 1183–1210.

Gill, L. (2001), Methods for automatic record matching and linking and their use in national statistics, *in* 'National Statistics Methodology Series', number 25.

Gonzalez, T. F. (1985), Clustering to minimize the maximum intercluster distance, *in* 'Theoretical Computer Science', Vol. 38, pp. 293–306.

Gu, L. & Baxter, R. (2004*a*), Decision models for record linkage, *in* 'AusDM 2004, Springer LNAI 3755', Cairns, Australia, pp. 146–160.

Gu, L. & Baxter, R. A. (2004*b*), Adaptive filtering for efficient record linkage, *in* 'Proceedings of the Fourth SIAM International Conference on Data Mining (SDM-04)', Orlando, Florida.

Han, J. & Kamber, M. (2001), *Data Mining: Concepts and Techniques*, Morgan Kaufmann, San Fransisco, CA.

Keogh, E., Lonardi, S. & Ratanamahatana, C. (2004), Towards parameter-free data mining, *in* '2004 ACM SIGKDD international conference on knowledge discovery and data mining', pp. 206–215.

Marzal, A. & Vidal, E. (1993), 'Computation of normalized edit distance and applications.', *IEEE Trans. Pattern Anal. Mach. Intell.* **15**(9), 926–932.

MatchWare Technologies (1998), *AutoStan and AutoMatch, User's Manuals*, Kennebunk, Maine.

McCallum, A., Nigam, K. & Ungar, L. (2000), Efficient clustering of high-dimensional data sets with application to reference matching, *in* 'Proceedings of ACM SIGKDD', Boston, pp. 169–178.

Michalowski, M., Thakkar, S. & Knoblock, C. A. (2004), Exploiting secondary sources for automatic object consolidation, *in* 'Proceedings of the 2004 VLDB Workshop on Information Integration on the Web'.

Mitchell, T. M. (1997), *Machine Learning*, McGraw-Hill, Boston.

Nahm, U., Bilenko, M. & Mooney, R. (2002), Two approaches to handling noisy variation in text mining, *in* 'Proceedings of the ICML-2002 workshop on text learning (TextML'2002)', Sydney, pp. 18–27.

Ravikumar, P. & Cohen, W. W. (2004), A hierarchical graphical model for record linkage, *in* 'roc. of the 20th Conference on Uncertainty in Artificial Intelligence', Banff, Canada, pp. 454–461.

Sarawagi, S. & Bhamidipaty, A. (2002), Interactive deduplication using active learning, *in* 'Proceedings of ACM SIGKDD', ACM Press, Edmonton, pp. 269–278.

Tejada, S., Knoblock, C. & Minton, S. (2002), Learning domain-independent string transformation weights for high accuracy object identification, *in* 'Proceedings of ACM SIGKDD', Edmonton, pp. 350–359.

Winkler, W. E. (1988), Using the EM algorithm for weight computation in the Fellegi-Sunter model of record linkage, *in* 'Proceedings of the Survey Research Methods Section, American Statistical Association'.

Winkler, W. E. (1990), String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage, *in* 'Section on Survey Research Methods', American Statistical Association, pp. 354–359.

Winkler, W. E. (2005), Approximate string comparator search strategies for very large administrative lists, Technical Report RRS2005/02, US Bureau of the Census.

Winkler, W. E. (2006), Overview of record linkage and current research directions, Technical Report RRS2006/02, US Bureau of the Census.

Witten, I. H. & Frank, E. (2005), *Data Mining: Practical machine learning tools and techniques*, 2nd edn, Morgan Kaufmann, San Francisco.

Yancey, W. E. (2004), An adaptive string comparator for record linkage, Technical Report RR2004/02, US Bureau of the Census.

Yancey, W. E. (2006), Evaluating string comparator performance for record linkage, Technical Report RRS2005/05, US Bureau of the Census.

Zhu, J. & Ungar, L. (2000), String edit analysis for merging databases, *in* 'KDD workshop on text mining, held at ACM SIGKDD', Boston.